Still all Greeklish to me: Greeklish to Greek Transliteration

Anastasios Toumazatos[*], **John Pavlopoulos**[*‡],
Ion Androutsopoulos[*‡], **Stavros Vassos**[†]
[*]Department of Informatics, Athens University of Economics and Business, Greece
[‡]Archimedes/Athena RC, Greece
[†]Helvia.ai
{toumazatos, annis, ion}@aueb.gr, stavros@helvia.ai

Abstract

Modern Greek is normally written in the Greek alphabet. In informal online messages, however, Greek is often written using characters available on Latin-character keyboards, a form known as Greeklish. Originally used to bypass the lack of support for the Greek alphabet in older computers, Greeklish is now also used to avoid switching languages on multilingual keyboards, hide spelling mistakes, or as a form of slang. There is no consensus mapping, hence the same Greek word can be written in numerous different ways in Greeklish. Even native Greek speakers may struggle to understand (or be annoyed by) Greeklish, which requires paying careful attention to context to decipher. Greeklish may also be a problem for NLP models trained on Greek datasets written in the Greek alphabet. Experimenting with a range of statistical and deep learning models on both artificial and real-life Greeklish data, we find that: (i) prompting large language models (e.g., GPT-4) performs impressively well with few- or even zero-shot training, outperforming several fine-tuned encoder-decoder models; however (ii) a twenty years old statistical Greeklish transliteration model is still very competitive; and (iii) the problem is still far from having been solved; (iv) nevertheless, downstream Greek NLP systems that need to cope with Greeklish, such as moderation classifiers, can benefit significantly even with the current non-perfect transliteration systems. We make all our code, models, and data available and suggest future improvements, based on an analysis of our experimental results.

Keywords: Transliteration, Greeklish, Greek

1. Introduction

Greeklish, a way to write Greek using the Latin alphabet (Koutsogiannis and Mitsikopoulou, 2017), has consistently maintained a significant presence as a means of written communication between Greek speakers over the last decades. Initially utilized in order to overcome the lack of support for Greek in older computers, this writing form continues to be used in informal settings, where it fills various needs. Such needs include being a convenient way to switch between a language written in the Latin alphabet (e.g., English) and Greek in the same passage, without switching the keyboard input language and without getting accustomed with Greek touch-typing. Other use cases for Greeklish include providing a way to write Greek without worrying about spelling mistakes, which is a concern both among native speakers and foreigners learning Greek, or as a means of writing slang.

Greeklish does not follow any official set of rules to dictate how certain characters in Greek should be mapped to Latin characters, leading to many Greeklish equivalents for any given text in Greek, and many unique styles of writing (Figure 1).[1] Furthermore, Greeklish does not resemble Greek solely on a phonetic basis. Rather, it typically consists of a combination of PHONETIC, OPTICAL, and KEY-SHARING representations (Table 1 shows examples). OPTICAL uses characters that have a similar look (e.g., "8" and "θ"), disregarding the phonetic resemblance to the source, while KEY-SHARING involves sharing the same key in English-Greek keyboards (e.g., "u" and "θ"). The choice of resemblance is largely subject to the writer's preferences, with all three categories often being intermixed in the same passage. Owing to the aforementioned arbitrariness in writing style and lack of rules, Greeklish poses a substantial challenge in

Figure 1: Greeklish lacks standardization, leading to many different writing styles across users.

[1]Greek can be transliterated to Latin characters using the ISO 843 standard, but its mappings are almost never used in Greeklish.

LREC-COLING 2024, pages 15309–15319

various NLP applications, where a model trained on regular Greek data may potentially be unable to handle Greeklish effectively (e.g., Figure 2).

In this work, we delve into the challenges and intricacies of Greeklish-to-Greek (G2G) transliteration. We investigate the performance of various statistical and deep learning models when confronted with artificial and real-life Greeklish data. Additionally, we evaluate the effectiveness of large language models (LLMs), such as GPT-4, which exhibit impressive performance with minimal or no specific training, when compared to the rest of the models. We also compare our work to previous attempts to tackle the problem of G2G transliteration, finding that a statistical model for Greeklish transliteration (Chalamandaris et al., 2004) developed in 2004 still remains highly competitive. Our experimental results also indicate that the G2G problem remains far from having been solved. Nevertheless, further experiments show that downstream Greek NLP systems that need to cope with Greeklish, such as toxicity moderation classifiers, can benefit significantly from current imperfect G2G transliteration methods. By sharing our code, models, and data, we aim to contribute to the existing body of knowledge on G2G and propose future improvements based on our experimental results.[2]

Greeklish	Greek	Category
e	αι	PHONETIC
e	ε	KEY-SHARING
ai	αι	OPTICAL
x	χ	OPTICAL
x	ξ	PHONETIC
ch	χ	PHONETIC
th	θ	PHONETIC
u	θ	KEY-SHARING
8	θ	OPTICAL

Table 1: Example Greeklish–Greek character mappings from different resemblance categories.

2. Related Work

Greeklish to Greek: Chalamandaris et al. (2004) introduced a G2G system grounded on statistical methods and lexicons from extensive corpora to handle the inconsistency and variety in Greeklish. The system transcribes Greeklish words into all possible phonetic representations, considering various Greeklish resemblance categories and combinations. Subsequently, a trigram model operating on Greek phonetic representations prunes these alternatives. The system then searches for the most probable solutions within a lexicon derived from large Greek corpora, making the final decision based on probabilities and context-dependent rules. Additionally, a language identification algorithm attempts to avoid transliterating non-Greek (e.g., English) words that may be be mixed with Greeklish (code switching). In subsequent work, Chalamandaris et al. (2006) ran an online data-collection demo, which led to a large dataset of human-verified transliterations and improvements of their previous system. The dataset is not publicly available, but the system can be used online.[3] Although similar alternative systems exist, we disregarded undocumented models. An example is GreeklishConverter, which is undocumented and performed poorly in preliminary experiments, compared to the model of Chalamandaris et al. (2006).[4]

Commercial APIs: We initially considered commercially available translation APIs, namely Google's Cloud Translation API[5] and Microsoft's Azure Translator.[6] In their current versions, neither service officially supports Greeklish. When translating from Greeklish to Greek the sentences of our test sets (§3), Google's service offered no translation for approximately one out of four sentences, returning them unchanged.[7] Microsoft's service fared even worse, leaving random parts of the inputs not transliterated, resulting in outputs that contained a mixture of Greeklish and Greek. These findings further reinforce our belief that the transliteration from Greeklish to Greek constitutes an open problem space, and that substantial progress can be made by revisiting it with new models and techniques.

Code Switching: Greeklish can in practice be interleaved with code switching (Fragkou, 2013), a phenomenon where words or phrases from different languages are mixed, often in the same sentence (Auer, 2013; Doğruöz et al., 2021). Unlike document- or sentence-level language identification (Ren et al., 2022; Jurgens et al., 2017), code switching requires segmenting sentences per language at the word level. To simplify our study, and to bypass the fact that we did not have any annotated data to train language segmentation models for Greeklish code-switched with other languages, we do not address code switching, i.e., we assume we are given a text in Greeklish that does not contain any parts written in other languages (e.g.,

[2] Our code, models, and data are publicly available at https://github.com/nlpaueb/greeklish.

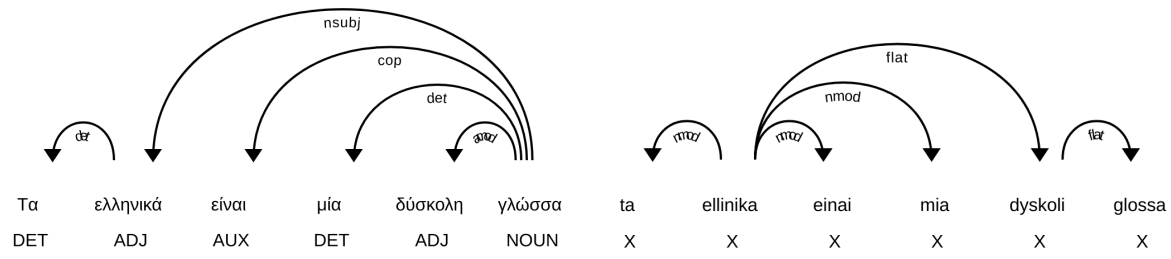
[3] http://speech.ilsp.gr/greeklish/greeklishdemo.asp

[4] https://greeklishconverter.com/

[5] https://cloud.google.com/translate

[6] https://azure.microsoft.com/en-us/products/cognitive-services/translator

[7] In preliminary experiments, it was achieving a character error rate as high as 40%.



(a) Greek (b) Greeklish

Figure 2: Dependency parsing (with spaCy) breaks down when the same sentence that is written in Greek ("Greek is a difficult language") is written in Greeklish.

English), though this assumption hurts the performance of our models on real-world data (§5.2).

Pinyin: The adaptation of languages to foreign alphabets is a common phenomenon around the world. A notable such example is Pinyin, a standardized system for writing romanized Chinese, utilizing the Latin alphabet, with the addition of four special diacritics to represent tonal variations. Pinyin plays a substantial role in aiding the learning of Mandarin Chinese and its pronunciation, rendering it a topic with continuous research activity. Earlier attempts, such as that of Chen et al. (2015), focused on developing Pinyin Input Method Engines that were based on neural or back-off n-gram language models. More recent research has focused on adapting state-of-the-art LLMs, such as the Chinese GPT-2 (Du, 2019) and a Chinese pre-trained language model (Zhang et al., 2020), to Pinyin transliteration. Tan et al. (2022) conducted experiments using both of the aforementioned systems, by including and excluding abbreviated Pinyin. Their conclusion was that, while the pre-trained model had difficulty adapting to Pinyin, GPT-2 and its subsequent modifications yielded promising results. Also, when abbreviations were included, performance significantly deteriorated. Unlike Greeklish, Pinyin is a more standardized system, being often employed in formal settings such as education (Lü, 2017).

Arabizi: Another notable case is Arabizi, a romanization system for Arabic. Arabizi differs from formal Arabic transliteration systems, which use explicit transliteration rules, along with diacritics and special characters not included in the Latin alphabet.[8] By contrast, Arabizi only uses characters of the Latin alphabet, as well as numbers to represent certain letters. It is employed predominantly in informal settings, being particularly popular for written communication over the Internet. Current research on this particular writing scheme is mostly oriented towards detection and transliteration from Arabizi to Arabic, as well as sentiment analysis. Shazal et al. (2020) experimented with various sequence-to-sequence models for Arabizi to Arabic transliteration, with their best system achieving a 80.6% word accuracy score and a 58.7% BLEU score. They did not consider, however, any Transformer-based models. Nevertheless, their work is the most recent development in Arabizi-to-Arabic transliteration. Regarding Arabizi sentiment analysis, recent research is focused on providing new datasets for training and benchmarking current and future systems (Fourati et al., 2021), as well as adapting Large Language Models for Arabizi. Baert et al. (2020) proposed an adaptation of BERT for Arabizi, while also releasing a large sentiment analysis dataset of general-dialect Arabizi, on which the model achieved F1 scores between 59.9% and 74.4%. They also obtained a score of 83.8% on a Tunisian-dialect Arabizi dataset (Fourati et al., 2020). Tackling code switching in Arabizi also posses a significant challenge. While no model has so far been able to solve this issue to a satisfactory extent, recent work focusing in isolation on identifying and segmenting code-switched Arabizi with various other languages, mainly English and French, has yielded promising results (Shehadi and Wintner, 2022). Being a non-standardized romanization scheme used almost exclusively for informal writing, Arabizi shares more similarities with Greeklish than Pinyin; and the previous research discussed above indicates that Arabizi to Arabic transliteration is far from having been solved, a conclusion we also reach for G2G.

[8] A popular standard for Arabic transliteration is ISO 233, its most current version being 233-2:1993.

3. Data

There is no publicly available parallel corpus for G2G. To bypass this problem, we developed a synthetic parallel corpus, by deriving automatically different plausible Greeklish transliterations of existing Greek data. For evaluation purposes, we used both synthetic and real-world parallel datasets.

3.1. Synthetic datasets

We created an empirically-derived mapping, shown in Table 2, which can be used to turn any Greek sentence to its plausible Greeklish renderings (Table 1).[9] For Greek characters that can be rendered in many ways in Greeklish, we chose randomly (uniform distribution) a rendering each time we encountered the characters, when generating synthetic Greeklish data from texts in the Greek alphabet. Although a particular speaker may in practice tend to use particular renderings more often, it is not uncommon for the same speaker to use different renderings for the same Greek alphabet character even in the same sentence. Importantly, our synthetic dataset creation assumes purely Greeklish input, as already noted. Real-world Greeklish text may contain code-switching with other languages (e.g., English), which is a factor to be considered when applying models trained on our synthetic data. Additionally, the rule-based nature of our synthetic dataset generation may limit its ability to fully capture certain contextual character-level dependencies present in real-world Greeklish usage, as well as adequately represent the diverse individual G2G mappings across different writers.

Europarl We used the Greek part of the Europarl corpus (Koehn, 2005),[10] which comprises all the Greek proceedings from the sessions of the European Parliament between 1996 and 2011. In total, the dataset contains 1,014,850 Greek sentences, with an average length of 162.65 characters.

TV subtitles We extracted 50,000 dialog turns from TV-series subtitles,[11] to include regularly used conversational Greek (including informal context, e.g., slang). This dataset offers a contrast to the domain-specific, more formal, and to an extent templated language of the European Parliament.

3.2. Real-world datasets

We developed and present two real-world parallel datasets intended for *evaluation* purposes. By real-world, we mean that we used texts originally written in Greeklish, which were transliterated by humans. By contrast, the synthetic datasets of the previous section were obtained by starting from texts written in the Greek alphabet and rendering them in Greeklish via the mappings of Table 2.

[9] Our criteria were based on pronunciation, character-shape similarities, and key-sharing. We are not aware of cases where more than two Greeklish characters are used to represent a Greek character, or vice versa.

[10] https://www.statmt.org/europarl/

[11] We include subtitles from 'Para 5', a popular Greek series, and the first season of 'Friends'.

Greeklish	Greek	Greeklish	Greek
A	A	N	N
Ai	Aι	Nt	Nτ
B	{B, Mπ}	O	{O, Ω}
D	{Δ, Nτ}	Oi	Oι
E	{E, Aι}	Ou	Ου
Ei	Eι	P	{Π, Ψ}
F	Φ	Q	Θ
G	Γ	R	P
H	{H, X}	S	Σ
I	{H, I, Y, Eι, Oι, Yι}	T	T
K	K	Th	Θ
Ks	Ξ	U	{Θ, Ου, Y}
L	Λ	V	B
M	M	W	Ω
Mp	Mπ	X	{Ξ, X}
Y	Y	o	{ο, ω}
Yi	Yι	oi	οι
Z	Z	ou	ου
a	α	p	π
ai	αι	ps	ψ
b	{β, μπ}	r	ρ
d	{δ, ντ}	s	{σ, ς}
e	{ε, αι}	t	τ
ei	ει	th	θ
f	φ	u	{υ, θ, ου}
g	γ	ui	υι
h	{η, χ}	v	β
i	{η, ι, υ, ει, οι, υι}	w	ω
k	κ	x	{ξ, χ}
ks	ξ	y	υ
l	λ	z	ζ
m	μ	n	ν
mp	μπ	nt	ντ
8	{Θ, θ}	3	{Ξ, ξ}

Table 2: Character and bi-character mapping from Greeklish to Greek, with braces denoting OR.

Survivorbot This dataset was provided by helvia.ai. It comprises Greeklish dialog turns of users who interacted with a chatbot developed to assist viewers of a popular TV show in Greece ('Survivor') during its first season. As shown in Table 3, texts were typically short (15 characters), but lengthy instances were also present. Slang and explicit content is also included, with very few examples of code-switched text. Six postgraduate students, trained for annotation tasks during their MSc courses, were employed to perform the transliteration task, and yield the ground truth. Inter-annotator agreement, measured with mean pairwise edit distance, was 0.60 ± 0.11.

Gazzetta We used the dataset of Pavlopoulos et al. (2017), manually selecting posts written mostly (or entirely) in Greeklish, and removing posts revealing sensitive information. This dataset was sourced from the discussion fora of a popular Greek sports site ('Gazzetta'). It contains various examples of informal language, instances of slang, as well as examples of code-switching, mainly in the form of sports terminology and foreign player names. We used both posts accepted (GAZZETTA-ACC) and rejected (GAZZETTA-REJ) by the moder-

Dataset	Size	Avg. len.	Max len.
EUROPARL	300	108.7	198
SURVIVORBOT	126	13.2	88
GAZZETTA-ACC	100	56.4	199
GAZZETTA-REJ	113	75.5	388
PARA5	300	20.5	39
FRIENDS	300	17.7	37

Table 3: Text statistics (in characters) per test set.

ator. Rejected posts often contain explicit expressions, more spelling and grammar mistakes, and are phrased in a more ambiguous manner, constituting a significantly more challenging transliteration task compared to accepted posts. The sentences were transliterated from Greeklish to Greek by four annotators.[12] Inter-annotator agreement with mean pairwise edit distance was 3.48±0.41 (accepted) and 6.05±0.84 (rejected).

4. Methods

4.1. RBSLM

The first method we developed, RBSLM, is inspired by noisy channel models for context-aware spelling correction (Jurafsky and Martin, 2023). Given a sequence of Greeklish characters $c_{1:m} = c_1, \ldots, c_m$, a rule-based component aware of all the common Greek-Greeklish mappings of (bi-) characters (Table 2) generates the set $RB(c_{1:m})$ of all plausible Greek character sequences $t_{1:r}$ (of possibly different lengths r) that may correspond to $c_{1:m}$. The hypotheses of $RB(c_{1:m})$ are ranked by a (character-based) n-gram language model,[13] and the top hypothesis $\hat{t}_{1:r}$ is kept:

$$\hat{t}_{1:r} = argmax_{t_{1:r} \in RB(c_{1:m})} P_{SLM}(t_{1:r})$$

where $P_{SLM}(t_{1:r})$ is the probability of the (statistical) n-gram language model.[14] In practice, $RB(w_{1:m})$ grows intractably large even for single sentences. Hence, we use a beam search decoder, which scans the Greeklish sequence $w_{1:m}$ from left to right (Fig. 3), uses the rule-based component to obtain possible transliterations of the next Greeklish character or bi-character to extend the current hypotheses (candidate sequences of Greek characters up to that point), then invokes the language model to rank the hypotheses and keep the r-best.[15] A shortcoming of RBSLM is the limited context-awareness of n-gram language models, which is further amplified by operating at the character level.

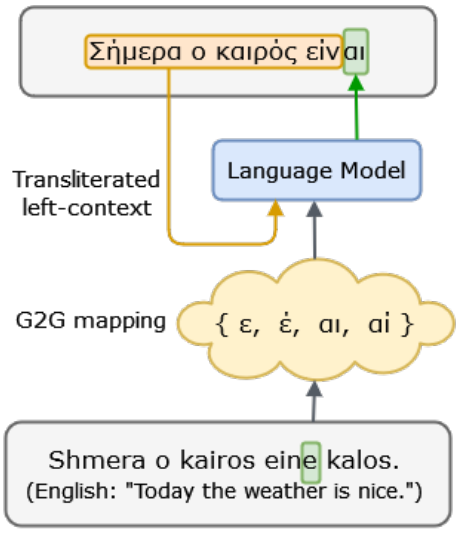


Figure 3: Illustration of beam search decoding in RBSLM and RBNLM. A single hypothesis (Greek alphabet characters so far) is shown for simplicity.

4.2. RBNLM

To address the limited context-awareness of RBSLM, we replace the n-gram language model by an LSTM-based (Hochreiter and Schmidhuber, 1997) neural language model,[16] also operating on characters. The resulting method, called RBNLM, is otherwise identical to RBSLM.

4.3. T5-based

We experimented with MT5 (Xue et al., 2021), the multilingual version of Google's T5 transformer (Raffel et al., 2020), as well as the subsequently developed BYT5 (Xue et al., 2022), a token-free model that handles text on the byte level. Ingólfsdóttir et al. (2023) concluded that for grammatical correction tasks, BYT5 outperforms subword-level models, such as MT5 and MBART (Liu et al., 2020), especially for morphologically rich languages (like Greek). Our study offers an opportunity to explore whether this trend holds in the context of a transliteration task. Both MT5 and BYT5 have undergone extensive pre-training on a vast multilingual dataset. We anticipate that by leveraging their previous exposure to Greek (possibly also Greeklish)

[12] We used the same annotators as in SURVIVORBOT.

[13] We tried $n \in \{2, \ldots, 9\}$ in preliminary experiments and 6 was the best.

[14] We experimented with bigram and trigram models, using simple Laplace smoothing, which we consider a reasonable option for character-based n-gram language models, where new n-grams do not emerge easily.

[15] The model remains fairly performant for r up to ten. All results are reported for this value of r. We observed diminishing returns for r between 5 and 7.

[16] We used a single LSTM layer, since stacking didn't provide any improvement in preliminary experiments.

during pre-training, they will be able to adapt to the G2G task with limited task-specific fine-tuning.

4.4. GPT-based

We included both GPT-3.5[17] and GPT-4[18] models in our experiments, as provided by OpenAI's API service.[19] We also experimented with the free web version of CHATGPT.[20] We experimented with prompt engineering in order to observe how various changes affect the transliterated output. Within this context, we explored providing the model with a set of parallel examples of source sequences in Greeklish along with their Greek transliterations as demonstrators (few-shot in-context learning). We further investigated the effect of asking CHATGPT to retry the transliteration, aiming to improve on its original attempt; a form of an iterative refinement process. By analysing the results of this approach, we aim to discover if there are significant differences between successive transliteration attempts, and whether this approach makes the model generate paraphrases, rather than transliterations of the input text.

4.5. All Greek to me!

We include in our experiments the system of Chalamandaris et al. (2006) called "All Greek to me!" (§2), hereafter ALLGREEK. Despite its age, ALLGREEK is the product of the most recent published research on G2G we are aware of, and was trained on real-world data (§2). It also includes a component intended to identify code-switched (e.g., English) fragments not to be transliterated, which is valuable in real-world data. Hence, ALLGREEK is a very strong baseline.

5. Experiments

We first present the evaluation measures and then the results of all the models using two families of parallel datasets, synthetic (Greeklish parts generated automatically from texts written in the Greek alphabet, using the mappings of Table 2) and real-world (real Greeklish texts transliterated by human annotators). The former comprises EUROPARL, FRIENDS, PARA5, while the latter comprises SURVIVORBOT, GAZZETTA-ACC, GAZZETTA-REJ. All synthetic test sets where derived from randomly selecting 300 texts of the corresponding original datasets, excluding data used for training or development. The real-world datasets were used only for testing (except from few-shot training in specific experiments discussed below).

5.1. Evaluation measures

We evaluate the performance of models using character error rate (CER) and word error rate (WER).[21] Since G2G is a transliteration task, measuring the error rate at the character and word level is an appropriate evaluation, unlike machine translation measures, such as BLEU (Papineni et al., 2002), BLEURT (Sellam et al., 2020), that are intended to 'forgive' paraphrases of the ground truth.

5.2. Results

We include results of the ALLGREEK model of Chalamandaris et al. (2006) on our own test datasets. We also use the configuration of RBSLM that performed best on development EUROPARL data, which uses a 6-gram character language model with a beam search width of 10. Both RBSLM and RBNLM were trained on 40k randomly selected Greek sentences from the EUROPARL dataset. MT5 and BYT5 were initially fine-tuned on a subset of 15k randomly selected samples from EUROPARL.[22] Having significantly outperformed MT5 in their respective base configurations, BYT5 was further fine-tuned on an additional 25k randomly selected texts from EUROPARL (BYT5-EU) or on 10k examples from the PARA5 and FRIENDS datasets respectively (BYT5-TV). We also examine the performance of BYT5-TV in a 90-shot training scenario[23], where for each real-world test dataset, the model is further fine-tuned on 90 test samples (Greeklish inputs and their gold transliterations) and asked to transliterate (is evaluated on) the rest of the test samples (BYT5-90S). Finally, we compare with the best-performing GPT-based model configuration of our experiments, GPT-4 with a 6-shot example initial prompt (GPT-4-6S). The remaining GPT-based models and configurations that where examined will be shared in an online Appendix, in our code repository.

Synthetic Table 4 presents the results of the models on the *synthetic* parallel datasets (Greeklish parts generated automatically from texts written in the Greek alphabet). BYT5-EU achieved the

[17] The exact version used was GPT 3.5-TURBO-0314.

[18] The exact version used was GPT -4-0314.

[19] https://openai.com/blog/openai-api

[20] We experimented with the versions available between April and May of 2023.

[21] We used Huggingface's CER, WER implementations.

[22] MT5 and BYT5, being encoder-decoder models, were trained using the artificially created Greeklish-Greek parallel version of EUROPARL, with Greeklish being the source and Greek the target language. RBSLM and RBNLM used the Greek part of the corpus only, to train their language models.

[23] We applied the 90-shot training only for the real-world datasets to simulate the case where some domain-specific training data are available.

best performance in EUROPARL, and BYT5-TV was the best on TV subtitles, as expected. GPT-4-6S performed clearly worse than the best model in all three datasets, but remarkably well given that it was provided with only six training examples. It also performed worse than BYT5 in EUROPARL and worse than the simpler ALLGREEK on TV subtitles.

MODEL	EUROPARL		FRIENDS (TV)		PARA5 (TV)	
	CER	WER	CER	WER	CER	WER
ALLGREEK	2.98	11.54	7.56	19.62	5.32	16.32
RBSLM	4.71	15.92	14.29	39.67	12.98	36.53
RBNLM	1.34	4.45	9.68	27.03	7.36	20.55
MT5	6.06	10.14	32.60	48.53	31.56	43.86
BYT5	1.30	4.11	14.36	38.20	13.93	34.28
BYT5-EU	**0.64**	**2.30**	11.36	29.74	9.80	25.04
BYT5-TV	1.37	3.92	**3.77**	**11.69**	**3.51**	**9.93**
GPT-4-6S	2.27	5.21	9.74	21.41	8.34	25.15

Table 4: Transliteration evaluation on **synthetic** parallel data (Greeklish part generated automatically from texts in Greek). Horizontal lines group systems trained on the same training subsets. The test subsets were the same for all systems.

Real-world Table 5 presents the results of the models on *real-world* parallel data (real Greeklish texts transliterated by humans). GPT-4-6S achieved the best performance across all measures and datasets, followed by ALLGREEK. BYT5 performed significantly worse, but it was improved when it was further fine-tuned on EUROPARL data (BYT5-EU) and (even more) when it was tuned on more informal TV subtitles (BYT5-TV), as one would expect. When we employed the 90-shot scheme, where BYT5-TV was further tuned on a small (90) instances from the test set (excluded from the evaluation), results improved further.

MODEL	SURVIVORBOT		GAZZETTA-ACC		GAZZETTA-REJ	
	CER	WER	CER	WER	CER	WER
ALLGREEK	14.66	30.48	9.71	24.54	11.17	28.94
RBSLM	22.70	55.16	18.55	43.01	20.70	43.07
RBNLM	19.99	50.21	14.59	33.10	17.41	35.42
MT5	40.99	59.67	30.02	43.70	39.26	50.03
BYT5	29.45	59.89	18.23	35.01	27.63	42.49
BYT5-EU	22.96	51.18	15.57	32.08	24.65	37.70
BYT5-TV	17.70	39.78	13.19	31.11	22.30	35.63
BYT5-90S	16.41	38.22	11.17	25.98	17.69	29.61
GPT-4-6S	**9.44**	**22.74**	**8.02**	**18.36**	**10.80**	**21.76**

Table 5: Transliteration evaluation on **real-world** parallel data (real Greeklish texts transliterated to Greek by humans).

5.3. Error Analysis

MT5 generated inaccurate and grammatically incorrect transliterations for a large percentage of inputs. Also, the output occasionally included words and characters not used in Greek (Table 6). The reason behind this may be MT5's massive token vocabulary (∼250,000 distinct tokens), which can make it hard for the model to learn a robust mapping between Greeklish and Greek tokens, also ensuring that irrelevant tokens (e.g., from Thai, Arabic) are not used in the transliterations.

BYT5 is pre-trained on the same amount of data as MT5, but does not seem to suffer from the same weaknesses as MT5 in G2G. The difference from MT5 is that BYT5 operates directly on UTF-8 bytes, and therefore its token space is limited to only a few hundred. We can only speculate that this limited token space allows it to learn better mappings between Greeklish and Greek characters.

ALLGREEK appears to randomly leave parts of a sentence untransliterated. We found no noticeable pattern in instances where this problem occurs, however it is likely due to sentences containing words that are not present in ALLGREEK's dictionary, and may therefore be misunderstood as code-switching (e.g., to English) by the system.

GPT-4-6S was the best out of all models in error rates. By exploring its transliteration outputs, we observe that it handled accurately rare words and names, abbreviations, and slang. However, it was at times reluctant to transliterate text containing inappropriate language. For example, it consistently failed to handle three particular samples from GAZZETTA-REJ, despite the various prompts we tried. However, we also note that the transliteration of explicit content is a relatively rare issue, and only desirable within very specific contexts.

6. Downstream Task: Moderation

We made an effort to explore the possible benefits or harms of automated G2G transliteration in a downstream NLP task. We considered moderation of online user-generated posts (Pavlopoulos et al., 2017), a task of high social impact. More specifically, we investigated to what extent providing the Greek transliterations of originally Greeklish posts alters the decisions and scores of a LLM, such as GPT, compared to using the original posts, when the task is to predict if a post should be rejected by the moderator or not.

6.1. Automated Moderation Setup

We used the Gazzetta dataset (§3.2), which contains both accepted and rejected posts from online users. That is, a moderator decided whether each such post was safe to publish in an online forum (accepted) or not (rejected). We used these moderator-assigned labels as the ground truth, to evaluate GPT-3.5 and GPT-4 in a zero-shot setting of the moderation task.[24] Posts were initially pro-

[24]We experimented with both GPT-3.5 and GPT-4 for this classification task, with the former performing better,

other-language characters	GREEKLISH	gia na to doume to allani to keno mpogiobits an kalucei
	GOLD	Για να το δούμε το αλάνι το κενό Bojovic αν καλύψει
	MT5	για να το δούμε το άλνενο το κένο μπορντ้ारअईΣ αν καλύπτει
incorrect spelling	GREEKLISH	to gipedo tis aek to eidane? :P
	GOLD	Το γήπεδο της ΑΕΚ το είδανε; :P
	BYT5	Το γυπέδο της αεκ το είδανε? :P
	MT5	Το γύρω πόλων της άμες το έθατε?
non-tranlsiterated words	GREEKLISH	ginei ke files, boris na mu peis;
	GOLD	γίνει και φίλες, μπορείς να μου πεις;
	ALLGREEK	γίνει και φίλες, boris να μου πεις;

Table 6: Transliteration errors (characters in red correspond to Thai).

Greeklish (SOURCE)	**Greek** (TRANSLITERATED)
An i united eprepe na valei k 4o gia na perasei tha to evaze xalara	Αν η United έπρεπε να βάλει **και** 4ο για να περάσει, θα το έβαζε άνετα
Ta topika einai ola xera kai giafto den paei kanena paidaki na spasi ta podia tou	Τα τοπικά είναι όλα ξερά και γι'αυτό δεν πάει κανένα παιδάκι να σπάσει τα πόδια του
Na afairethoun t vraveia apo ton berg ton anastasiou kai ton risvani na t dwsoun ston mitroglou p einai stin agglia ston mitsel pou pire prwtathlima apo ton augousto p perase kai kalutero neo paixth na to dwsoun ston vergo	Να αφαιρεθούν τα βραβεία από τον Βεργκ τον Αναστασίου και τον Ρισβάνη να τα δώσουν στον Μήτρογλου που είναι στην Αγγλία στον Μίτσελ που πήρε πρωτάθλημα από τον Αύγουστο που πέρασε και για καλύτερο νέο παίχτη να το δώσουν στον Βέργο

Table 7: Terms altered correctly by GPT-3.5, including abbreviations, short forms, and named entities.

vided written in Greeklish, i.e., in their original form, and the LLM was instructed to act as the moderator (Table 8). Then, we repeated this experiment by providing the LLM with automatically generated transliterations instead of the original Greeklish posts. We used the transliterations generated by GPT-4-6S, since it was the best performing system in the previous experiments.

You are an expert in online content moderation. The following is an online user comment in Greek or Greeklish, which can either be accepted for publication or rejected as toxic by the moderator. Classify the next candidate post as either 1 for "rejected" or 0 for "accepted"

Table 8: Moderation prompt for GPT models.

and 0.55, respectively. Looking at specific examples, it seems that providing a transliterated input helps GPT correct its mistakes by (i) turning abbreviated or short-form Greeklish to correct Greek (from orange to green in Figure 7), and (ii) writing named entities correctly, such as names of players, teams and countries, that might otherwise appear as unknown words (or noise) when written in Greeklish (purple in Figure 7). These results indicate that automatically transliterating Greeklish input has the potential to improve the performance of an existing downstream Greek NLP system that has to cope with Greeklish, even when the transliterations are not completely accurate, and without any changes in the downstream system itself.

6.2. Empirical Analysis

The results of the downstream application of G2G (Table 9) show an increased Accuracy when the posts are automatically transliterated before automatic moderation, from 0.49 to 0.53.[25] Following Pavlopoulos et al. (2017), we also evaluate using $F_{\beta=2}(P_{reject}, P_{accept})$, which places more emphasis on avoiding wrongly accepted posts in a fully-automatic moderation scenario, reporting a considerable improvement, from 0.54 to 0.69. Also, both Precision and Recall improve from 0.51 to 0.59

	GREEKLISH		TRANSLITERATED	
ACCURACY	0.49		**0.53**	
$F_2(P_{reject}, P_{accept})$	0.54		**0.69**	
	PRE	REC	PRE	REC
ACCEPTED	0.47	0.71	**0.50**	**0.89**
REJECTED	0.54	**0.30**	**0.69**	0.21
AVERAGE	0.51	0.51	**0.59**	**0.55**

Table 9: Accuracy, $F_{\beta=2}$, Precision, and Recall of GPT-3.5 in zero-shot moderation of Greeklish and (automatically transliterated) Greek posts.

hence we only show its results to save space.

[25] Similar benefits were observed when GPT-4 was the moderator (two Accuracy points higher from 0.46).

7. Conclusions

Experimenting with a range of statistical and deep learning models on both artificial and real-life Greeklish-to-Greek parallel data, we found that: (i) prompting LLMs performs impressively well with few- or even zero-shot training, outperforming several fine-tuned encoder-decoder models; however (ii) the twenty years old statistical Greeklish transliteration model of Chalamandaris et al. (2004, 2006) is still very competitive; and (iii) the G2G problem is still far from having been solved; (iv) nevertheless, downstream Greek NLP systems that need to cope with Greeklish, such as moderation toxicity classifiers, can benefit significantly even with the current non-perfect transliteration systems. Future work will need to effectively address code switching between Greeklish and other languages (e.g., English), which likely affects the performance of our G2G models in real-world applications. We make all our code, models, and data available.

8. Ethics Statement

- We are committed to making our research results and data open and accessible to the scientific community whenever possible.
- We have provided clear and transparent reporting of our methods, results, and findings, hiding information only for the purposes of anonymity during the review process.
- This work is original, and it has not been previously published elsewhere.
- We declare no potential conflicts of interest that might be perceived as influencing the results or recommendations presented in this paper.
- All data sources, including human subjects, have been treated with respect and have received appropriate consent, when necessary.
- All authors have reviewed and agreed on the content of this paper and their respective contributions.

Acknowledgements

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

Peter Auer. 2013. *Code-switching in conversation: Language, interaction and identity*. Routledge.

Gaétan Baert, Souhir Gahbiche, Guillaume Gadek, and Alexandre Pauchet. 2020. Arabizi language models for sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 592–603, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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).

Aimilios Chalamandaris, Pirros Tsiakoulis, Spyros Raptis, Georgios P Giannopoulos, and George Carayannis. 2004. Bypassing greeklish! In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Shenyuan Chen, Hai Zhao, and Rui Wang. 2015. Neural network language model for Chinese Pinyin input method engine. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 455–461, Shanghai, China.

Kevin KH Chung. 2002. Effective use of hanyu pinyin and english translations as extra stimulus prompts on learning of chinese characters. *Educational Psychology*, 22(2):149–164.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Zeyao Du. 2019. Gpt2-chinese: Tools for training gpt2 model in chinese language. https://github.com/Morizeyao/GPT2-Chinese.

Chayma Fourati, Hatem Haddad, Abir Messaoudi, Moez BenHajhmida, Aymen Ben Elhaj Mabrouk, and Malek Naski. 2021. Introducing a large Tunisian Arabizi dialectal dataset for sentiment analysis. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 226–230, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Chayma Fourati, Abir Messaoudi, and Hatem Haddad. 2020. Tunizi: a tunisian arabizi sentiment analysis dataset. In *AfricaNLP Workshop, Putting Africa on the NLP Map. ICLR 2020, Virtual Event*, volume arXiv:3091079.

Pavlina Fragkou. 2013. Text segmentation for language identification in Greek forums. In *Proceedings of the Workshop on Adaptation of Language Resources and Tools for Closely Related Languages and Language Variants*, pages 23–29, Hissar, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Svanhvít Lilja Ingólfsdóttir, Petur Ragnarsson, Haukur Jónsson, Haukur Simonarson, Vilhjalmur Thorsteinsson, and Vésteinn Snæbjarnarson. 2023. Byte-level grammatical error correction using synthetic and curated corpora. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7299–7316, Toronto, Canada. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2023. Spelling correction and the noisy channel. In *Speech and Language Processing, 3rd Edition (in preparation). Appendix B.*

David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 51–57, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

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

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Chan Lü. 2017. The roles of pinyin skill in english-chinese biliteracy learning: Evidence from chinese immersion learners. *Foreign Language Annals*, 50(2):306–322.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Xingzhang Ren, Baosong Yang, Dayiheng Liu, Haibo Zhang, Xiaoyu Lv, Liang Yao, and Jun Xie. 2022. Unsupervised preference-aware language identification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3847–3852, Dublin, Ireland. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Ali Shazal, Aiza Usman, and Nizar Habash. 2020. A unified model for Arabizi detection and transliteration using sequence-to-sequence models. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 167–177, Barcelona, Spain (Online). Association for Computational Linguistics.

Safaa Shehadi and Shuly Wintner. 2022. Identifying code-switching in Arabizi. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 194–204, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Minghuan Tan, Yong Dai, Duyu Tang, Zhangyin Feng, Guoping Huang, Jing Jiang, Jiwei Li, and Shuming Shi. 2022. Exploring and adapting Chinese GPT to Pinyin input method. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 1899–1909, Dublin, Ireland. Association for Computational Linguistics.

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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, Fanchao Qi, Xiaozhi Wang, Yanan Zheng, Guoyang Zeng, Huanqi Cao, Shengqi Chen, Daixuan Li, Zhenbo Sun, Zhiyuan Liu, Minlie Huang, Wentao Han, Jie Tang, Juanzi Li, Xiaoyan Zhu, and Maosong Sun. 2020. Cpm: A large-scale generative chinese pre-trained language model.