Cross-Lingual Transfer from Related Languages: Treating Low-Resource Maltese as Multilingual Code-Switching

Kurt Micallef[1]
kurt.micallef@um.edu.mt

Nizar Habash[2]
nizar.habash@nyu.edu

Claudia Borg[1]
claudia.borg@um.edu.mt

Fadhl Eryani[3,2]
fadhl.eryani@nyu.edu

Houda Bouamor[4]
hbouamor@cmu.edu

[1]Department of Artificial Intelligence, University of Malta
[2]Computational Approaches to Modeling Language Lab, New York University Abu Dhabi
[3]University of Tübingen
[4]Carnegie Mellon University Qatar

Abstract

Although multilingual language models exhibit impressive cross-lingual transfer capabilities on unseen languages, the performance on downstream tasks is impacted when there is a script disparity with the languages used in the multilingual model's pre-training data. Using transliteration offers a straightforward yet effective means to align the script of a resource-rich language with a target language, thereby enhancing cross-lingual transfer capabilities. However, for mixed languages, this approach is suboptimal, since only a subset of the language benefits from the cross-lingual transfer while the remainder is impeded. In this work, we focus on Maltese, a Semitic language, with substantial influences from Arabic, Italian, and English, and notably written in Latin script. We present a novel dataset annotated with word-level etymology. We use this dataset to train a classifier that enables us to make informed decisions regarding the appropriate processing of each token in the Maltese language. We contrast indiscriminate transliteration or translation to mixing processing pipelines that only transliterate words of Arabic origin, thereby resulting in text with a mixture of scripts. We fine-tune the processed data on four downstream tasks and show that conditional transliteration based on word etymology yields the best results, surpassing fine-tuning with raw Maltese or Maltese processed with non-selective pipelines.

1 Introduction

Due to their impressive cross-lingual transfer capabilities, multilingual models have facilitated the development of NLP tools for low-resource languages (Kondratyuk and Straka, 2019; Wu and Dredze, 2019; Conneau et al., 2020). However, multilingual models may fall short in addressing lower-resourced languages (Wu and Dredze, 2020; Muller et al., 2021). In particular, Muller et al. (2021) show that the cross-lingual transfer capabilities of a model are affected if the related language seen during pre-training uses a different script. They further show that transliterating to match the script of the related language improves performance.

In this work, we focus on Maltese – a Semitic language with an Arabic base and substantial Romance influences written in Latin script. Micallef et al. (2023) transliterate Maltese into Arabic script and demonstrate improved performances in certain scenarios when fine-tuning with an Arabic large language model as opposed to a multilingual one in the original script. However, being influenced by a mixture of languages – predominantly Arabic, Italian, and English – we argue that transliterating Maltese entirely into the Arabic script ignores the non-Arabic aspect of the language. Hence, the advantages derived from transliteration are diminished by the losses incurred through moving farther from Italian and English.

Therefore, we propose mixing scripts and applying transliteration selectively. Specifically, we apply transliteration to Maltese words of Arabic origin, keeping the others in their original Latin script. We also experiment with mixing transliterations with word-level translations, which yielded the best results overall.

Our main contributions are as follows:

1. We annotate a new Maltese dataset with etymological tags (Section 3).
2. We train several etymological classifiers using the annotated data (Section 4.1).
3. Using automatic etymological classifications,

we define various processing pipelines to conditionally transliterate or translate words based on their etymology (Section 4.2).

4. We conduct a thorough evaluation, fine-tuning a variety of language models with different processing pipelines and shed new light on the cross-lingual transfer capabilities exhibited by these models (Section 5).

The code, the new etymological annotations, and classifiers are released publicly.[1]

2 Background and Related Work

Due to the mixed nature of Maltese, the language can be viewed as a highly code-switched language among Arabic, Italian, and English. An analysis of the dictionary from Aquilina (1987, 1990) reveals that 32.4%, 52.5%, and 6.1% of Maltese words are of Arabic, Italian/Sicilian, and English origin, respectively (Brincat, 2017). The remaining cases include mixed or unknown-origin words. We note that Arabic-origin words tend to have higher token frequencies and include function words, and the dictionary entries do not include all inflected cases.

Our work is related to previous works dealing with languages not written in their standard script and/or mixed with other languages, predominantly English. Pant and Dadu (2020) define a pipeline for Hinglish written in Latin script, which only transliterates Hindi-tagged tokens to Devanagari script. Eskander et al. (2014) define a pipeline for transliterating Arabizi (Darwish, 2014) text into Arabic script, which includes separate subprocesses for symbols, names, foreign words, and Arabic words. Shazal et al. (2020) define a neural model for transliteration of Arabizi text into Arabic script, but they skip English words similar to Pant and Dadu (2020).

While these approaches are similar to some of the pipelines presented in Section 4.2, the majority of their token distribution (80%+) is in Latinized Hindi or Arabic, compared to around 60% Arabic-origin tokens for Maltese (Table 1). This, in addition to the evolution of Maltese as a distinct language, adds to the complexity of using off-the-shelf models for language modeling (Chau et al., 2020; Muller et al., 2021; Micallef et al., 2022).

Thus, in this work, we build a robust classification model to predict word etymologies, using newly annotated data, to provide more accurate information to our processing pipelines.

3 Etymology Annotations

To build our dataset, we extracted 439 sentences (9,683 tokens) from the Maltese Universal Dependencies Treebank (Čéplö, 2018) training set. We were directly involved in the creation of the guidelines, the annotation of the tokens, including extensive discussions and resolution of disagreements. Among us, we have native language expertise in Arabic and Maltese and second language expertise in English and Italian. We relied extensively on authoritative references (mentioned below). The following are the labels we annotated with.

Arabic Maltese tokens of Arabic origin, following the etymological classification by Aquilina (1987, 1990). This includes words that are derived from Arabic dialects, such as Tunisian Arabic, but we retain the same classification for these.

Non-Arabic Maltese tokens whose origin is some language other than Arabic. During our annotation, we noticed that most of these are of Italian origin. There were a few cases that were of English origin, for example, *ċekk* ‘cheque’. An ambiguity arises for certain Maltese words which correspond to related words in both Italian and English, for example, *rapport* is closely related to both English ‘report’ and Italian ‘rapporto’.[2] Moreover, a few words are also derived from other languages, such as Sicilian and French. For these reasons, we opt to group these words under this single category.

Mixed These are Maltese tokens containing a mixture of Arabic and non-Arabic influences. The mixed influences take various forms, of which we identify the following sub-categories:

1. **Verbs**: Verbs of non-Arabic origin with Arabic morphology to convey different conjugations. For example, *nispjegaw* ‘we explain’, from Italian ‘spiegare’ with the Arabic prefix *ni-* (1st Person Present) and suffix *-w* (Plural). Careful attention was given to Maltese words that share a close surface form with Italian. For example, although the Maltese verb *spjega* ‘he explained’ has a similar form to Italian *spiega* ‘he explains’, the difference in their tense inflection lead us to consider the Maltese verb as **Mixed** and not **Non-Arabic**, since it does not follow the Italian conjugation rules.

[1] https://github.com/MLRS/malti/tree/2024.eacl

[2] Aquilina (1990) lists both Italian and English words as possible cognates for *rapport*.

2. **Plurals**: Non-Arabic-origin nouns that form the plural with Arabic morphology using regular and broken plural formations. For example, regular plural *partijiet* 'parts' composed of the stem *parti* (Italian 'parte'), and the Arabic suffix *-ijiet*; broken plural *ġranet* 'days', singular *ġurnata* (Italian 'giornata').
3. **Univerbations**: Single words composed of several Arabic and non-Arabic words. For example, *minflok* 'instead of', which is composed of *minn* 'from' (Arabic من *mn*),[3] *fi* 'in' (Arabic في *fy*), and *lok* 'location' (Sicilian 'locu', Italian 'località').

Code-switching Non-Maltese words borrowed from another language, typically English. As such, these words do not follow Maltese orthographic rules as they are written verbatim from the borrowed words.

Name Names of entities that are further categorized into **Name (Arabic)** and **Name (Non-Arabic)** for names of Arabic and non-Arabic origin, respectively. Again, we rely on the etymological classification given by Aquilina (1987, 1990), but make use of additional sources to determine the origin of certain names – for surnames, for instance, we use Maltagenealogy.[4] Note that this category does not capture entities composed of words that could be used for non-entities. For example, *Gvern ta' Malta* 'Government of Malta' would be considered as a single entity in a Named-Entity Recognition task, but we annotate the phrase as *Gvern*/**Non-Arabic** *ta'*/**Arabic** *Malta*/**Name (Arabic)**. Non-Maltese words in named entities are tagged as either **Code-Switching** if translatable, or **Name** if not. For instance, while both words in *Planning Authority* would be classified as **Code-Switching**, both words in *JF Motors* are tagged as **Name (Non-Arabic)**.

Symbol Tokens that can be considered language universal such as digits and punctuation symbols.

A summary of the annotation frequencies is given in Table 1. In addition to the raw token counts, we also provide the etymology distribution for the set of unique tokens (types).

Label	**Token**		**Type**	
Arabic	5,848	60%	1,122	47%
Non-Arabic	1,559	16%	660	27%
Mixed	271	3%	186	8%
Code-Switching	398	4%	169	7%
Name (Arabic)	146	2%	36	1%
Name (Non-Arabic)	423	4%	171	7%
Symbol	1,038	11%	65	3%
Total	**9,683**	**100%**	**2,409**	**100%**

Table 1: Etymology annotation frequencies of tokens and types.

4 Methodology

Our objective is to process Maltese tokens in such a way as to improve cross-lingual transfer. We design pipelines that use **transliteration** and **translation** as our main tools to process Maltese (Section 4.2).

For **transliteration**, we use the implementation from Micallef et al. (2023). Specifically, we extend the non-deterministic character mappings with Tunisian word model ranking and full closed-class token mappings, by making some modifications to the character maps. Primarily, we add mappings for digits and other common symbols to Arabic script instead of passing them as is. We also include additional mappings for some letters that were missing in Micallef et al. (2023), such as from *t* to ث *θ*.

For **translation**, word-level translations are extracted from Google Translate. Admittedly, this may give sub-optimal translations due to the lack of sentence context. However, we do not translate at the sentence level because we make token-level decisions and sometimes require partial translations of a subset of words in a sentence. In addition, most of the tasks used in the evaluation (Section 5) are token-level classification tasks. Hence, we decided against using word aligners with a sentence translation since this would amplify the noise in the processing pipeline. At the same time, word-level translations allow us to reduce the processing power needed, as they are extracted once on the unique set of tokens in the datasets used in Section 5, and saved as static token mappings.

The processing pipelines make use of an etymology classifier (Section 4.1), which also uses the transliterations and the translations as features.

4.1 Etymology Classifiers

Using the data from Section 3, we build a classifier. We experiment with the following models.

[3] HSB Arabic transliteration (Habash et al., 2007).

[4] https://maltagenealogy.com/maltese-surname-origins/

Translation A set of heuristics based on word-level translations and the edit distances between them and the original token. When the distance with both the Italian and English translations is 0, it is considered to be a Symbol if it contains digits or punctuation symbols and a Name otherwise. When the distance with either of the Italian or English translations is 0, it is considered to be Code-Switching. Otherwise, the token is considered to be Arabic or Non-Arabic, based on the minimum distance between the Arabic, Italian, and English translations. We calculate the Arabic distance using the transliteration instead of the original token. As such this is not trained on the data as it uses the features statically.

MLE A Maximum Likelihood Estimator that predicts the tag observed for the token in the training data. When multiple tags are observed for a given word, the most frequently seen tag is predicted. If a token has never been encountered before, the most commonly observed tag is predicted, which, in this context, is Arabic.

CRF A Conditional Random Field (Lafferty et al., 2001) model which makes predictions using the sentence context. In addition to the original and lower-cased token and positional markers, for each token, the following features are included:

- **Orthography**: low-level boolean features indicating the presence of uppercase characters, digits, punctuation symbols, and Maltese special characters (*ċ*, *ġ*, *ħ*, and *ż*).
- **N-Grams**: Boolean features indicating the presence of a frequent n-gram in the token and the presence of each n-gram in the token. A set of 197 frequent n-grams is extracted by taking the unique uncased words from Korpus Malti v4.0 (Micallef et al., 2022) and computing the most common trigrams and bigrams.
- **Closed-Class**: a boolean feature indicating whether the token is one of the full closed-class tokens from Micallef et al. (2023).
- **Trans^2**: the **translations** of a token into Arabic, Italian, and English, taken from the pre-computed token-level translations. We also include the **transliteration** of the token into Arabic.
- **Distances**: the Levenshtein distance (Levenshtein, 1966) between the token and each of the translations. The Arabic translation distance is computed using the transliteration.

Model	**All**	**Seen**	**Unseen**
Translation	69.72	70.27	66.39
MLE	92.11	99.76	43.64
CRF	91.97	99.20	45.93
+ orthography	92.90	99.22	52.64
+ n-grams	96.51	99.43	78.19
+ closed-class	92.98	99.17	53.41
+ trans^2	93.77	99.36	58.49
+ distances	95.75	99.29	73.35
+ all features	97.55	99.64	**84.35**
Ensemble	**97.69**	**99.80**	**84.35**

(a) All Categories ($n = 7$)

Model	**All**	**Seen**	**Unseen**
Translation	73.89	73.68	75.15
MLE	92.13	99.78	43.64
CRF	98.26	99.61	**89.80**
Ensemble	**98.43**	**99.81**	**89.80**

(b) Merged Categories ($n = 5$)

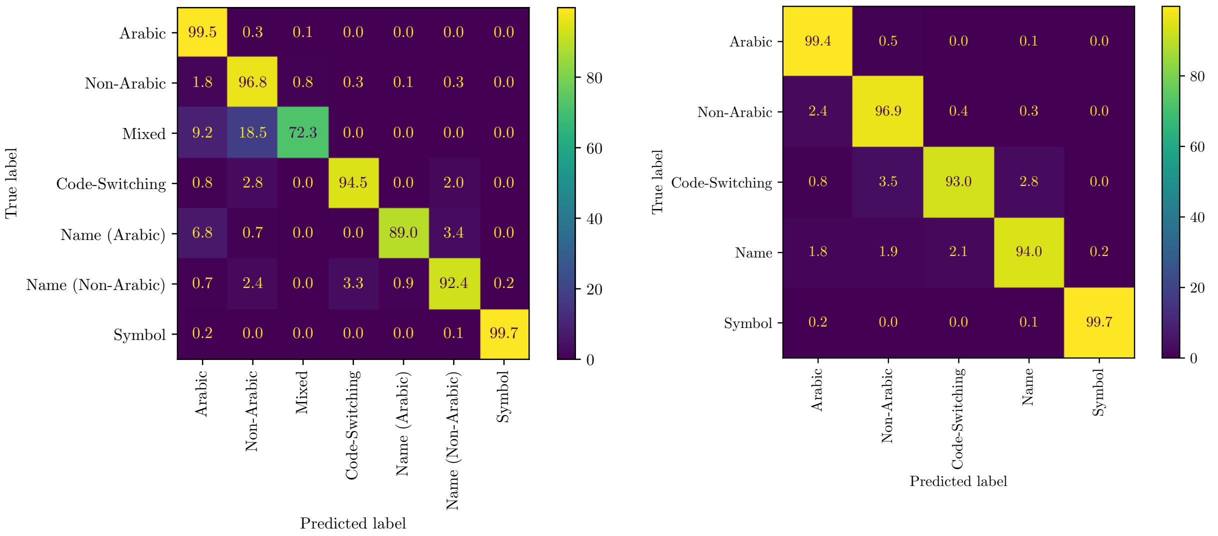
Table 2: Etymology classification accuracy across 10-fold cross-validation.

All features except for the trans^2 features, are based on the implementation from Osmelak and Wintner (2023).

Ensemble We combine MLE and CRF into one model. This favors the predictions from the MLE model whenever the token is seen exclusively with a single tag. Otherwise, the predictions from the CRF model are used.

All models are trained using 10-fold cross-validation, using the same splits. The results are shown in Table 2a, reporting the accuracy from all folds. For the CRF model, we contrast the performance of using no features, adding the orthography features only, adding every other group of features on top of this, and adding all of the features together. For the Ensemble, we show the results with all features. In addition to the scores for the entire data, we also show individual results for tokens that are seen in the corresponding training split versus tokens that are not seen.

With the exception of the Translation model, all models achieve over 91% accuracy, with Ensemble achieving the best results overall. While the Translation model performs relatively worse, it performs evenly across seen and unseen tokens. In contrast, the MLE model is heavily biased towards seen words. With no additional features, the CRF model performs worse overall than the MLE model,



(a) All Categories

(b) Merged Categories

Figure 1: Confusion matrices for the Ensemble classifier. Values are percentages and are obtained by adding predictions across folds and normalizing them by dividing by the total for a given class.

albeit obtaining a higher accuracy on unseen words. This highlights that many tokens in the data are unambiguous as a simple MLE baseline achieves close to perfect performance on seen tokens.

Adding low-level orthographic features is sufficient for the CRF model to perform better than the MLE baseline. The other features all contribute to some degree of further improvement, particularly on unseen words which boosts the overall accuracy. The n-gram and distance features give the most noticeable improvements overall with a 20-25% improvement on unseen tokens.

All features together yield the best performance for the CRF model. Despite this, accuracy scores on seen words are worse than those of the MLE model. This is mitigated by the Ensemble model, which gets an even better score on seen words than MLE, since it gets better predictions on tokens that are seen but with different labels.

Analyzing the predictions reveals that the model makes systematic errors, as shown in Figure 1a. In particular, the Mixed class is considerably mispredicted as non-Arabic or Arabic. To a lesser degree, Names are conflated with each other, with some confusion with non-name labels, namely, Non-Arabic and Code-Switching for Name (Non-Arabic), and Arabic for Name (Arabic).

Merged Categories To mitigate the possible negative effect of such mispredictions on our downstream task, we merge the Mixed tag with the **Non-Arabic** tag, and the Name tags together under a single **Name** tag. The total number of categories is thus reduced from 7 (All) to 5 (Merged). The decision to merge is motivated by system design and does not invalidate the importance of the various annotated categories, which were driven by linguistic insights. Furthermore, we note that tokens from the merged categories share a common set of properties and merging does not impact the decisions made in Section 4.2.

We report the results of the models using all features in Table 2b, showing similar trends to Table 2a. All models attain higher accuracy scores on these merged categories, with the CRF and Ensemble models getting close to 90% accuracy on unseen words. As shown in Figure 1b, the confusion is drastically reduced overall as well.

Going forward, we use the Ensemble model trained on all the data with merged categories.

4.2 Text Processing Pipelines

We now make use of the classifier outlined in Section 4.1 to make decisions on how to process Maltese text. We define processing pipelines that, given a sequence of Maltese tokens, output another sequence of the same length. A given token is processed in one of the following ways:

- **Pass ($\texttt{P}$)**: The original token is returned without any modification, so using Maltese as is.
- **Transliteration ($\texttt{X}_{\texttt{ara}}$)**: Transliteration into Arabic script.
- **Translation ($\texttt{T}^{\texttt{src}}_{\texttt{tgt}}$)**: Translation from a source

language $\mathtt{src}$ to a target language $\mathtt{tgt}$. We consider Arabic ($\mathtt{ara}$), Italian ($\mathtt{ita}$), and English ($\mathtt{eng}$) as different target languages.

We design several processing pipelines in which we apply one of these actions for a given token, depending on the token's etymology label. The different pipelines and corresponding actions are summarized in Table 4 and we also show their outputs after processing a sample sentence in Table 3.

The $\mathtt{P}$, $\mathtt{X_{ara}}$, and $\mathtt{T_{\star}}$ pipelines perform a pass, transliterate, and translate action indiscriminately, and hence, do not use etymology classifications.

The $\mathtt{X_{ara}/T_{\star}}$ pipelines mix transliteration and translation. $\mathtt{X_{ara}/T_{ara}}$ transfers every token to Arabic script by transliterating tokens of Arabic origin and Symbols, translating everything else. As highlighted in the human evaluation by Micallef et al. (2023), transliterations of Maltese words of Arabic origin are generally mapped to the Arabic cognate, whereas the transliteration system does not produce a coherent output for Maltese words of non-Arabic origin. Thus, we map tokens we expect to be distant from Arabic using translation instead of transliteration. Differently from the $\mathtt{T_{\star}}$ pipelines, Code-Switching tokens are translated from English instead of Maltese.[5]

The $\mathtt{X_{ara}/T_{ita}}$ and $\mathtt{X_{ara}/T_{eng}}$ pipelines similarly mix transliteration with Italian and English translations, respectively. This produces text that combines a mixture of scripts seamlessly. Differently from $\mathtt{X_{ara}/T_{ara}}$, we do not translate Code-Switching tokens, since these can already be considered as non-Maltese tokens and the output produced by the $\mathtt{X_{ara}/T_{ita}}$ and $\mathtt{X_{ara}/T_{eng}}$ pipelines already contains a mixture of scripts. Similarly, the $\mathtt{X_{ara}/P}$ pipeline produces mixed script text by combining transliteration with pass. The rationale for this pipeline is similar to the $\mathtt{X_{ara}/T_{ita}}$ and $\mathtt{X_{ara}/T_{eng}}$ pipelines. However, with this pipeline, the aim is to measure the impact of minimizing script differences between related words without using translation.

5 Downstream Task Evaluation

In this section, we conduct an extrinsic evaluation on four downstream tasks: Part-of-Speech tagging (POS), Dependency Parsing (DP), Named-Entity Recognition (NER), and Sentiment Analysis (SA). Refer to Section 5.1 for further details on the tasks.

Each dataset is processed using all of the pipelines presented in Section 4.2, keeping the corresponding labels/tags the same. The processed datasets are then used to fine-tune pre-trained language models. We run fine-tuning 5 times with different random seeds and report the mean performance. The language models used are the multilingual model mBERT (Devlin et al., 2019), the Arabic model CAMeLBERT-Mix (Inoue et al., 2021), the Italian model ItalianBERT (Schweter, 2020), the English model BERT (Devlin et al., 2019), and the Maltese model BERTu (Micallef et al., 2022).

Due to the large number of combinations, we do not fine-tune every model on all the pipelines. Instead, we only fine-tune models on the pipelines which produce data in a language that it has been intentionally pre-trained on. So we fine-tune CAMeLBERT on all pipelines which do a $\mathtt{T_{ara}}$ and/or $\mathtt{X_{ara}}$ action, ItalianBERT on all pipelines which do a $\mathtt{T_{ita}}$ action, and BERT on all pipelines which do a $\mathtt{T_{eng}}$ action. mBERT is fine-tuned on all pipelines since it is multilingual. Additionally all models are fine-tuned on the $\mathtt{P}$ and $\mathtt{X_{ara}/P}$ pipelines to test their capabilities on Maltese using only fine-tuning data. The results are presented in Section 5.2

5.1 Tasks

We follow all fine-tuning architectures and hyperparameters suggested by Micallef et al. (2022). See Appendix B for further details.

We use the MUDT (Čéplö, 2018) dataset for the DP task. For the POS task, we use the MLRS POS dataset (Gatt and Čéplö, 2013) with the same splits from Micallef et al. (2023). The dataset from Martínez-García et al. (2021) is used for SA, tokenized as in Micallef et al. (2023) to allow for the token-level actions used to process the data.

We use the MAPA NER data (Gianola et al., 2020) for the NER task using only the level 1 tags. However, we normalize this data to be in line with the tokenization scheme used in the MUDT and MLRS POS datasets (see Appendix A for further details). This step is crucial since the original dataset splits off the **-** and **’** characters as separate tokens. These characters carry important linguistic features in Maltese which are used by the transliteration system (Micallef et al., 2023) and can at times change the meaning of the token.[6]

[5] We do not consider translating from Italian since almost all cases of code-switching observed during our annotation in Section 3 are in English.

[6] For example, *fil-* ‘in the’ and *fil* ‘mortar joint’ or *ta’* ‘of’ and *ta* ‘he gave’.

etymology label	Arabic	Non-Arabic	Name	Arabic	Symbol	Arabic	Code-Switching	Non-Arabic	Symbol
token	Il-	karozza	Porsche	tal-	2022	għandha	speed	fenomenali	!
P	Il-	karozza	Porsche	tal-	2022	għandha	speed	fenomenali	!
X_{ara}	ال	كردزة	برسكهي	تاع ال	٢٠٢٢	عندها	صباد	فنملي	!
T_{ara}	ال	ترام	بورشه	ل	2022	هو	سرعة	هائل	!
T_{ita}	IL	tram	Porsche	Di	2022	Esso	velocità	fenomenale	!
T_{eng}	The	streetcar	Porsche	of	2022	it	speed	phenomenal	!
X_{ara}/P	ال	karozza	Porsche	تاع ال	2022	عندها	speed	fenomenali	!
X_{ara}/T_{ara}	ال	ترام	بورشه	تاع ال	٢٠٢٢	عندها	سرعة	هائل	!
X_{ara}/T_{ita}	ال	tram	Porsche	تاع ال	2022	عندها	speed	fenomenale	!
X_{ara}/T_{eng}	ال	streetcar	Porsche	تاع ال	2022	عندها	speed	phenomenal	!

Table 3: An example sentence and the corresponding etymology labels passed through the pipelines outlined in Table 4. The raw sentence is: *Il-karozza Porsche tal-2022 għandha speed fenomenali!* 'The 2022 Porsche car has phenomenal speed!'.

	Arabic	Non-Arabic	Code-Switching	Name	Symbol
P	P	P	P	P	P
X_{ara}	X_{ara}	X_{ara}	X_{ara}	X_{ara}	X_{ara}
T_{ara}	T^{mlt}_{ara}	T^{mlt}_{ara}	T^{mlt}_{ara}	T^{mlt}_{ara}	T^{mlt}_{ara}
T_{ita}	T^{mlt}_{ita}	T^{mlt}_{ita}	T^{mlt}_{ita}	T^{mlt}_{ita}	T^{mlt}_{ita}
T_{eng}	T^{mlt}_{eng}	T^{mlt}_{eng}	T^{mlt}_{eng}	T^{mlt}_{eng}	T^{mlt}_{eng}
X_{ara}/P	X_{ara}	P	P	P	P
X_{ara}/T_{ara}	X_{ara}	T^{mlt}_{ara}	T^{eng}_{ara}	T^{mlt}_{ara}	X_{ara}
X_{ara}/T_{ita}	X_{ara}	T^{mlt}_{ita}	P	T^{mlt}_{ita}	P
X_{ara}/T_{eng}	X_{ara}	T^{mlt}_{eng}	P	T^{mlt}_{eng}	P

Table 4: Data processing pipelines and the action performed for each corresponding etymology class: transliteration (X_{ara}), translation (T^{src}_{tgt}), and pass/nothing (P).

5.2 Results

The results are summarized in Table 5. As expected, BERTu remains the best-performing model across all tasks due to its pre-training on Maltese. With P, mBERT performs worse than BERTu. However, it does better than the other monolingual models. This is largely due to its multilinguality, as it was exposed to several different languages, including those related to Maltese. In contrast, the other monolingual models only include some of the languages with relevance to Maltese. Moreover, CAMeLBERT performs the worst on the P pipeline due to the script difference. Hence, we designate P BERTu as the topline setup to close the gap with, and P mBERT as the baseline setup to beat. Similar to the findings from Micallef et al. (2023), X_{ara} CAMeLBERT performs better than X_{ara} mBERT on POS and SA and P mBERT on DP and SA.

A discussion of the other pipelines and their results follows. Unless explicitly mentioned, we do not include BERTu in the rest of this discussion.

5.2.1 Translations over Transliterations

Using mBERT, the T_* pipelines give worse performance on POS and DP compared to P and X_{ara}. Conversely, the monolingual models generally give better performance on these tasks, with the exception of CAMeLBERT which gives worse performance than X_{ara}. However, mBERT performs better overall than the monolingual models with the T_* pipelines. A jump in performance is observed on the NER task, using T_* compared to the X_{ara} pipeline, but only mBERT T_{eng} gives better performance than mBERT P.

On the other hand, on the SA task T_* give better results than P and X_{ara}, regardless of the model used. In fact, the best scores overall are attained with the T_* pipelines for the SA task, with BERT T_{eng} performing the best across all pipelines. T_{eng} is, in general, the best-performing pipeline across all T_* pipelines, likely due to the prevalence of Maltese-English parallel data compared to other language pairs,[7] which, in turn, results in better translation performance between this language pair compared to other pairs.

These findings highlight that while training with translated data can be an effective solution for low-resource languages, it is largely dependent on the

[7] For Maltese, OPUS (Tiedemann and Nygaard, 2004) reports 27.9K, 6.0M, and 34.1M parallel sentences with Arabic, Italian, and English, respectively, at the time of writing.

Pipeline	POS	DP	NER	SA
	Acc.	LAS	F1	F1
$\mathtt{P}$ (Topline)	98.3	88.1	84.0	83.1

(a) BERTu (Micallef et al., 2022)

Pipeline	POS	DP	NER	SA
	Acc.	LAS	F1	F1
$\mathtt{P}$	88.9	61.8	75.5	61.9
$\mathtt{X_{ara}}$	96.0	77.3	74.1	67.5
$\mathtt{T_{ara}}$	91.2	69.6	77.2	73.8
$\mathtt{X_{ara}/P}$	95.6	76.6	76.9	65.0
$\mathtt{X_{ara}/T_{ara}}$	95.7	77.3	76.2	70.2
$\mathtt{X_{ara}/T_{ita}}$	95.5	76.5	76.2	66.2
$\mathtt{X_{ara}/T_{eng}}$	96.0	77.4	78.2	64.1

(b) CAMeLBERT-Mix (Inoue et al., 2021)

Pipeline	POS	DP	NER	SA
	Acc.	LAS	F1	F1
$\mathtt{P}$	89.7	64.3	80.1	62.3
$\mathtt{T_{ita}}$	92.7	71.9	79.6	70.9
$\mathtt{X_{ara}/P}$	44.9	14.6	76.1	58.2
$\mathtt{X_{ara}/T_{ita}}$	47.9	17.9	76.2	64.1

(c) ItalianBERT (Schweter, 2020)

Pipeline	POS	DP	NER	SA
	Acc.	LAS	F1	F1
$\mathtt{P}$	96.1	73.0	79.7	64.2
$\mathtt{T_{eng}}$	93.6	74.7	79.9	**75.2**
$\mathtt{X_{ara}/P}$	96.0	72.5	77.9	63.7
$\mathtt{X_{ara}/T_{eng}}$	96.4	73.9	79.1	69.4

(d) BERT (Devlin et al., 2019)

Pipeline	POS	DP	NER	SA
	Acc.	LAS	F1	F1
$\mathtt{P}$ (Baseline)	96.7	77.3	81.0	67.3
$\mathtt{X_{ara}}$	95.8	77.4	75.7	62.5
$\mathtt{T_{ara}}$	91.4	71.3	77.5	74.3
$\mathtt{T_{ita}}$	92.6	72.9	79.6	71.3
$\mathtt{T_{eng}}$	94.2	75.8	81.5	73.1
$\mathtt{X_{ara}/P}$	96.6	78.8	80.3	66.2
$\mathtt{X_{ara}/T_{ara}}$	95.4	77.4	76.5	66.2
$\mathtt{X_{ara}/T_{ita}}$	96.5	**79.2**	79.7	67.3
$\mathtt{X_{ara}/T_{eng}}$	**96.8**	**79.2**	**82.2**	67.7

(e) mBERT (Devlin et al., 2019)

Table 5: Results using the data processing setups defined in Table 4, grouped by language model. Accuracy, Labelled Attachment Score (LAS), span-based F1, and macro-averaged F1 are reported for the POS, DP, NER, and SA tasks, respectively. Each value is an average of 5 runs with different random seeds. For each task, the best scores (excluding the Topline) are **bolded**, and all scores better than the Baseline are shaded.

type of task and the performance of the translation model.

This trend is also observed for $\mathtt{X_{ara}/T_{ara}}$. For the POS and DP tasks, the added translations give worse performance than $\mathtt{X_{ara}}$ but better performance than $\mathtt{T_{ara}}$ due to the decreased translations. Conversely, the opposite is true for the NER and SA tasks where $\mathtt{X_{ara}/T_{ara}}$ performs better than $\mathtt{X_{ara}}$ but worse than $\mathtt{T_{ara}}$.

5.2.2 Multilingual Models, Multilingual Text

With mBERT, all $\mathtt{X_{ara}/T_{*}}$ pipelines give better performances than the corresponding $\mathtt{T_{*}}$ pipelines and $\mathtt{X_{ara}}$ on POS, DP, and NER with the exception of $\mathtt{T_{ara}}$ which performs better than $\mathtt{X_{ara}/T_{ara}}$ on NER. $\mathtt{X_{ara}/T_{ara}}$ yields the worst results of all $\mathtt{X_{ara}/T_{*}}$ pipelines, since, similar to $\mathtt{T_{ara}}$, this is not fully exploiting the multilinguality aspect of the model. mBERT $\mathtt{X_{ara}/T_{ita}}$ achieves the best overall performance on the DP task.

$\mathtt{X_{ara}/T_{eng}}$ mBERT achieves better results than $\mathtt{P}$ mBERT on all tasks and achieves the best scores across all pipelines in the POS, DP, and NER tasks. Besides English being the dominant language in mBERT's pre-training data, we hypothesise that the performance of Maltese-English translation models (as highlighted in Section 5.2.1) also plays a role in this result. Furthermore, as the gap in performance with BERTu is further reduced, this offers a viable option to further give performance improvements over standard fine-tuning for low-resource languages with similar mixing to Maltese.

Although its pre-training data does not include Maltese, mBERT obtains better results on POS, DP, and NER when trained with $\mathtt{X_{ara}/P}$ instead of $\mathtt{X_{ara}}$ and $\mathtt{T_{*}}$ (except for $\mathtt{T_{eng}}$ on NER). mBERT $\mathtt{X_{ara}/P}$ also achieves a better score than mBERT $\mathtt{P}$ on DP. This finding supports the evidence from Muller et al. (2021) who show that transliteration to the same script as the related language in the pre-training data improves cross-lingual transfer. Additionally, mBERT $\mathtt{X_{ara}/P}$ performs competitively with mBERT $\mathtt{X_{ara}/T_{*}}$, performing slightly better than $\mathtt{X_{ara}/T_{ara}}$ and $\mathtt{X_{ara}/T_{ita}}$ on the POS and NER tasks.

5.2.3 Monolingual Models, Multilingual Text

The trends from Section 5.2.1 do not hold entirely for $\mathtt{X_{ara}/T_{ita}}$ and $\mathtt{X_{ara}/T_{eng}}$. ItalianBERT with $\mathtt{X_{ara}/T_{ita}}$ performs worse, sometimes significantly, compared to $\mathtt{P}$ and $\mathtt{T_{ita}}$. $\mathtt{X_{ara}/T_{eng}}$ BERT performs worse than $\mathtt{T_{eng}}$ on all tasks except POS.

CAMeLBERT generally performs worse with $\mathtt{X_{ara}/T_{ita}}$ than $\mathtt{X_{ara}/T_{ara}}$, $\mathtt{T_{ara}}$, and $\mathtt{X_{ara}}$. The exceptions are $\mathtt{X_{ara}}$ on NER due to the reduction in performance that we observe when transliterating names into Arabic script (Section 5.2.4), and $\mathtt{T_{ara}}$ on POS and DP since increasing transliterations and decreasing translations show improved performance for these tasks (Section 5.2.1).

Overall these results make sense since we are giving the respective models less of the type of language they were pre-trained on: transliterations in the case of ItalianBERT and BERT and Italian translations in the case of CAMeLBERT. Similarly, $\mathtt{X_{ara}/P}$ gives worse performance for the non-Arabic monolingual models, since they were not pre-trained on Maltese, although the discrepancy with $\mathtt{X_{ara}/T_{ara}}$ is negligible.

5.2.4 Arabic Script on Names

Overall we observe that changing Name tokens to Arabic script ($\mathtt{X_{ara}}$, $\mathtt{T_{ara}}$, and $\mathtt{X_{ara}/T_{ara}}$) gives among the worst results in the NER tasks. A big factor for this is the lack of casing information not present in the Arabic script, supported by the findings from Mayhew et al. (2019).

5.2.5 Multilingual Presence in Monolingual Models

Despite having less Arabic text overall, $\mathtt{X_{ara}/T_{eng}}$ yields the best performance for CAMeLBERT on the POS, DP, and NER tasks. This could be explained by the presence of the Latin script in CAMeLBERT's pre-training data, which is being exploited by the modeling.

Similarly, BERT has likely seen some Arabic text in its pre-training, since with $\mathtt{X_{ara}/T_{eng}}$ it achieves better performance compared to $\mathtt{T_{eng}}$ on POS. It is also not significantly worse on the other tasks, particularly when compared to the results of $\mathtt{X_{ara}/T_{ita}}$ ItalianBERT.

These results support the findings by Blevins and Zettlemoyer (2022) and Muennighoff et al. (2023) who identify that large-scale pre-training corpora contain language contamination, resulting in languages that are unintentionally seen at pre-training.

6 Conclusion

In this work, we analyze how partially transliterating Maltese has an impact on downstream task performance. We present a newly annotated dataset with word etymology labels and build classifiers to predict these labels. Using these classifiers, we design various pipelines to make decisions on which tokens to transliterate or otherwise.

Our evaluation using mBERT shows that by exclusively transliterating words of Arabic origin, downstream task performance improves. The best results are achieved by mixing transliterations with translations, where including English translations yields better results than fine-tuning on the original data on all tasks. These findings corroborate with those from Muller et al. (2021), but we show this further by only transliterating words that would aid cross-lingual transfer.

Future work should explore language adaptation techniques (Chau et al., 2020; Pfeiffer et al., 2020) using the pipelines presented here, to further improve the cross-lingual transfer capabilities of multilingual models. It is also interesting to apply this method during inference of few-shot and zero-shot settings. We also hope that our newly annotated dataset can be used as a resource to further support the understanding of Maltese. While we have reduced the script difference between Maltese and its related languages, other linguistic properties can also impact the cross-lingual performance (Philippy et al., 2023) and future work should investigate these facets.

7 Limitations

The pipelines using translation are limited by the performance of the models used. We did not systematically evaluate different translation systems to find out the best-performing system.

Although word-level translations allow us to reduce computing requirements, sentence-level translations are bound to produce more accurate translations. Aligning translations to get the corresponding translation for a word is particularly challenging, especially with varying levels of morphological richness and limited tools for low-resource languages. Even if these pipelines are combined as a single model that produces the output sentence, this still needs to be aligned to the original token in the data for token classification tasks such as Part-of-Speech tagging, Dependency Parsing, and Named-Entity Recognition.

In our pipelines, we have not treated Names much differently from other tokens. While some names can be handled by transliteration, especially those of Arabic origin, it is more challenging for others, particularly since many names we annotated use English orthographic rules.

Our findings are also limited to the task results presented here. As such, a wider variety of tasks, including higher-level semantic tasks, is desirable to verify the generalizability of such a method. Moreover, having this evaluation on a wider variety of language models would be ideal to assess how factors such as pre-training data and model architecture influence the results.

8 Ethics Statement

The biases present in the data and language models we used are inherited. We acknowledge that some performance errors may be due to introduced ambiguities or errors in the techniques we studied. That said, we do not foresee any major risks.

Acknowledgements

We acknowledge support from the LT-Bridge Project (GA 952194) and DFKI for access to the Virtual Laboratory. We further acknowledge funding by Malta Enterprise.

References

Joseph Aquilina. 1987. *Maltese-English Dictionary Vol. I, A-L.* Midsea Books, Valletta, Malta.

Joseph Aquilina. 1990. *Maltese-English Dictionary Vol. II, M-Z.* Midsea Books, Valletta, Malta.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explains the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Joseph M. Brincat. 2017. Maltese: blending Semitic, Romance and Germanic lexemes. *Lexicographica*, 33(2017):207–224.

Slavomír Čéplö. 2018. *Constituent order in Maltese: A quantitative analysis*. Ph.D. thesis, Charles University, Prague.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online. Association for Computational Linguistics.

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

Kareem Darwish. 2014. Arabizi detection and conversion to Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 217–224, Doha, Qatar. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ramy Eskander, Mohamed Al-Badrashiny, Nizar Habash, and Owen Rambow. 2014. Foreign words and the automatic processing of Arabic social media text written in Roman script. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 1–12, Doha, Qatar. Association for Computational Linguistics.

Albert Gatt and Slavomír Čéplö. 2013. Digital Corpora and Other Electronic Resources for Maltese. In *Proceedings of the International Conference on Corpus Linguistics*, pages 96–97. UCREL, Lancaster, UK.

Lucie Gianola, Ēriks Ajausks, Victoria Arranz, Chomicha Bendahman, Laurent Bié, Claudia Borg, Aleix Cerdà, Khalid Choukri, Montse Cuadros, Ona de Gibert, Hans Degroote, Elena Edelman, Thierry Etchegoyhen, Ángela Franco Torres, Mercedes García Hernandez, Aitor García Pablos, Albert Gatt, Cyril Grouin, Manuel Herranz, Alejandro Adolfo Kohan, Thomas Lavergne, Maite Melero, Patrick Paroubek, Mickaël Rigault, Mike Rosner, Roberts Rozis, Lonneke van der Plas, Rinalds Vīksna, and Pierre Zweigenbaum. 2020. Automatic removal of identifying information in official eu languages for public administrations: The MAPA project. In *Proceedings of the 33rd International Conference on Legal Knowledge and Information Systems (JURIX'20)*, pages 223–226. IOS Press.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, pages 15–22. Springer, Netherlands.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages

2779–2795, Hong Kong, China. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics – Doklady*, 10:707–710.

Antonio Martínez-García, Toni Badia, and Jeremy Barnes. 2021. Evaluating morphological typology in zero-shot cross-lingual transfer. 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 3136–3153, Online. Association for Computational Linguistics.

Stephen Mayhew, Tatiana Tsygankova, and Dan Roth. 2019. ner and pos when nothing is capitalized. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6256–6261, Hong Kong, China. Association for Computational Linguistics.

Kurt Micallef, Fadhl Eryani, Nizar Habash, Houda Bouamor, and Claudia Borg. 2023. Exploring the impact of transliteration on NLP performance: Treating Maltese as an Arabic dialect. In *Proceedings of the Workshop on Computation and Written Language (CAWL 2023)*, pages 22–32, Toronto, Canada. Association for Computational Linguistics.

Kurt Micallef, Albert Gatt, Marc Tanti, Lonneke van der Plas, and Claudia Borg. 2022. Pre-training data quality and quantity for a low-resource language: New corpus and BERT models for Maltese. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 90–101, Hybrid. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Doreen Osmelak and Shuly Wintner. 2023. The denglisch corpus of German-English code-switching. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 42–51, Dubrovnik, Croatia. Association for Computational Linguistics.

Kartikey Pant and Tanvi Dadu. 2020. Towards code-switched classification exploiting constituent language resources. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 37–43, Suzhou, China. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Fred Philippy, Siwen Guo, and Shohreh Haddadan. 2023. Identifying the correlation between language distance and cross-lingual transfer in a multilingual representation space. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 22–29, Dubrovnik, Croatia. Association for Computational Linguistics.

Stefan Schweter. 2020. Italian BERT and ELECTRA models.

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.

Jörg Tiedemann and Lars Nygaard. 2004. The OPUS corpus - parallel and free: `http://logos.uio.no/opus`. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

A MAPA Data Fixes

The Maltese data from Gianola et al. (2020) is fixed to have consistent tokenization with the other token classification datasets used in Section 5. We do this by re-tokenizing the raw text using the MLRS Tokenizer.[8] Further to this, we also manually split off trailing - and ' for tokens that do not carry the linguistic meaning for Maltese. For instance, marking number ranges with - or using ' for quotation marks.

Since some of the tokens from the original data are merged into a single token, the corresponding labels are also merged. Whenever the merged tokens contain different target labels, we keep them separate.

While doing this process, we went through the inconsistencies between the tokens in the data and the new tokens. While there were legitimate cases where the source tokenization made sense, we identified certain entity spans that were incorrectly marked, typically a missing character in the whole word. In these cases, we fix the annotation so that the span is consistent with the tokenization.

Lastly, we also fixed some of the labels which contained errors during our conversion. For cases where the entity span was marked but no label was present, we added the labels. When there were inconsistencies between the level 1 and level 2 tags, we fixed the incorrect tag appropriately.

We make this dataset publicly available.[9]

B Experimental Setup

The number of parameters for language models used in Section 5 is summarized in Table 6.

Model	**Parameters**
BERTu	126M
mBERT	179M
CAMeLBERT	109M
ItalianBERT	111M
BERT	109M

Table 6: Number of parameters for the language models used in Section 5

We use NVIDIA A100 GPUs (40GB and 80GB, depending on memory requirements) on a compute cluster. Fine-tuning time depends on the model used and the pipeline from Section 4.2 with which the data was processed with, but a single GPU was always used. Giving a rough estimate for each task: Part-of-Speech tagging takes around an hour and a half, Dependency Parsing takes around 1 hour, Named-Entity Recognition takes around 6 hours, and Sentiment Analysis takes around 30 minutes. Named-Entity Recognition takes significantly longer since the dataset used is larger and we use gradient accumulation to ease memory requirements while keeping the same effective batch size from Micallef et al. (2022). The figures reported here include all of the runs with different random seeds, the test evaluation for each run, and any initial setup necessary for startup.

[8] https://mlrs.research.um.edu.mt/

[9] https://huggingface.co/datasets/MLRS/mapa_maltese