UNIVERSITÄT
DES
SAARLANDES

With the support of the
Erasmus+ Programme
of the European Union

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

# MASTER'S THESIS

Christian Cayralat

# Orthography Standardization in Arabic Dialects

Department of Computational Linguistics and Phonetics

| | |
|---|---|
| Supervisors of the master thesis: | Prof. Dr. Josef van Genabith<br>Dr. Cristina España i Bonet<br>RNDr. Daniel Zeman, Ph.D. |
| Study programme: | Language Science and Technology |

Saarbrücken 2021

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In Saarbrücken date 05.08.2021 Christian Cayralat

Author's signature

This thesis topic tackles issues which are being studied to this day, issues with a degree of complexity such that it has prevented us from formulating definitive solutions to. Naturally, without the generous support and dedication of a number of key persons, this work would not have been possible.

First and foremost, I would like to thank my advisors Dr. Daniel Zeman, Prof. Dr. Josef van Genabith, and last but not least, Dr. Cristina España i Bonet for their continuous support, for taming a thesis the topic of which was broad enough to satiate the demands of a doctoral thesis, and for always keeping it bounded and down to earth, both technically and managerially.

Second in line, are the people which made a sizeable part of the thesis a reality and who went out of their way to help me out of sheer generosity and interest in the topic. Huge thanks go to the annotators Carine Khoury, Sara Tello, Wiaam Haddad, Rami Khaddaj, and Ghiwa Honein for believing in this project and having stuck with me till the end despite the very difficult circumstances we went through to produce the annotated corpus in this thesis. A very special thanks is also in order, to Mario Korban and Jawad Hamadani for helping with the adjoining annotation software which was used to produce the corpus.

I would also like to extend my gratitude to the following people which have helped me one way or another, and the help of which was essential in transcending the barriers of knowledge and logistics which impeded the research momentum: Dr. Nizar Habash, Cennet Oguz, Wissam Antoun, Fady Baly, Dr. Jirka Hana, Sunit Bhattacharya, Dr. Vladislav Kuboň, Dr. Marketa Lopatková, Akshay Aggarwal, Vilém Zouhar, Joseph Aziz, Danielle Inaty, Chadi Kahwaji, Rony Najem, Ziad Chelala, Bassem Riachi, Rana TN Riachi, Salim Riachi, Joelle Riachi, Rosette Feghali, Marie Claire Feghali, Beatrice Akowuah, Kevin Zakka.

Finally, to the people who were behind the scenes and who saw everything unfold from day one (and even before), the people that make up the backbone of my support, Mom and Dad.

I would like to dedicate this thesis to my people, the Lebanese people, and especially to the victims and survivors of the the August 4th 2020 Beirut Port blast, as it coincides with its painful commemoration. It avoidably killed, displaced, cast down, dejected, and crushed more people that I can bear to count. But in the face of all odds, this people rose up from the ashes, as it has done countless times before. A people whose relationship with its own language is as perplex as the perplexities that govern it. I hope with the whole of my heart that this work will compellingly do it justice.

Title: Orthography Standardization in Arabic Dialects

Author: Christian Cayralat

Department: Institute of Formal and Applied Linguistics (ÚFAL)

Supervisors:
RNDr. Daniel Zeman, Ph.D., ÚFAL
Prof. Dr. Josef van Genabith, Saarland University, Computational Linguistics and Phonetics, DFKI
Dr. Cristina España i Bonet, DFKI

Abstract: Spontaneous orthography in Arabic dialects poses one of the biggest obstacles in the way of Dialectal Arabic NLP applications. As the Arab world enjoys a wide array of these widely spoken and recently written, non-standard, low-resource varieties, this thesis presents a detailed account of this relatively overlooked phenomenon. It sets out to show that continuously creating additional noise-free, manually standardized corpora of Dialectal Arabic does not free us from the shackles of non-standard (spontaneous) orthography. Because real-world data will most often come in a noisy format, it also investigates ways to ease the amount of noise in textual data. As a proof of concept, we restrict ourselves to one of the dialectal varieties, namely, Lebanese Arabic. It also strives to gain a better understanding of the nature of the noise and its distribution. All of this is done by leveraging various spelling correction and morphological tagging neural architectures in a multi-task setting, and by annotating a Lebanese Arabic corpus for spontaneous orthography standardization, and morphological segmentation and tagging, among other features. Additionally, a detailed taxonomy of spelling inconsistencies for Lebanese Arabic is presented and is used to tag the corpus. This constitutes the first attempt in Dialectal Arabic research to try and categorize spontaneous orthography in a detailed manner.

Keywords: arabic NLP dialects diglossia spontaneous orthography standardization machine translation BERT corpus annotation augmentation morphology taxonomy linguistics machine learning multi-task

# List of Figures

---

[1] **1** is the Examples Menu. **2** is the Raw Text section where segmentation happens. **3** is the "Fix Sentence" button. **4** are the input fields resulting from CODA* segmentation, where spelling can be standardized. **5** are the input fields resulting from morphological segmentation where tagging happens. **6** is the "Search Previous Annotations" menu. **7** is the "Filter By Flag" button. **8** are the sentences available for annotation. **9** is the "Reset All" button. **10** is the "Reset State" button. **11** is the "Save and Next" button.

# List of Tables

---

[2] Unless otherwise specified, all transliterations in this report are realized using the Habash-Soudi-Buckwalter transliteration scheme.

---

[3] A tag with a star next to it refers to any tag that starts with what is right behind the star. For example PART* might mean PART, PART_DET, ADJ_INTERROG, etc. There are a few exceptions to those which are: ADJ_COMP, VERB_PSEUDO, PRON_INTERROG, and PRON_EXCLAM. Their names are misleading but they do not take any features.

[4] The lemmatization column holds the value of the respective feature which should be used to lemmatize a word.

# Contents

# Introduction to Arabic NLP

## The Story of Arabic

Modern Standard Arabic (MSA) is a Semitic (Afro-Asiatic) language based on Classical Arabic (CA), the language of the Qur'an, the holy scripture of Islam. It is extremely similar to it at all linguistics levels, i.e., phonological, morphological, syntactic, etc. It mainly differs with the latter in style and lexicon which tend to be more modern and simple, occasionally incorporating new structures from other languages, which would be considered incorrect in CA. Both use the Arabic script which is written from right to left, and are mostly written (c.f., spoken). What the Arabic people usually speak is what is referred to as Dialectal Arabic. MSA is generally the medium of choice for news, administrative, literary, and political topics, among others, i.e., *formal speech*. The same way Received Pronunciation is 'received' in England, MSA is taught at school formally to native Arabs, but they do not use it to communicate in day-to-day life. Habash (2010) even goes as far as stating that "standard Arabic is not any Arab's native language". Arabic is spoken by 400 million people worldwide.

Arabic NLP has recently witnessed great developments, as state-of-the-art methods were applied to very idiosyncratic problems related to the processing of the Arabic language. MSA is a morphologically rich language (MRL) which realizes its morphemes through both a concatenative process[5] and a templatic process[6]. While MSA is not resource-scarce, as a fair number of well-annotated treebanks are available for it (Hajič et al., 2004; Taji et al., 2017a), it does not enjoy as great an abundance in data as other well-studied languages. Coupled with its distinctive features, most notably, the difficulty of its script and great variation it holds, it poses many challenges to NLP researchers. In addition to being a MRL, the fact that diacritics (denoting vowels) are optional in Arabic text makes ambiguity scenarios even more abundant, so much so, that every token has on average 12 morphological analyses without diacritization (Habash, 2010).

After reaching high accuracy scores for many of the common NLP tasks for MSA such as tokenization, segmentation, lemmatization, diacritization, part-of-speech tagging, and parsing, and continuing to improve them, more attention in the recent past was put in some of the more high-level tasks using statistical, data-driven methods. The most prominent ones which are being researched for MSA today are Machine Translation (MT), Text Summarization, and Sentiment Anal-

---

[5] A base word is a concatenation of a word stem and its affixes, and can enclosed by clitics.

[6] Inflection and derivation of a root based on different templates.

ysis (Elsayed Abd Elaziz et al., 2019).

# Dialectal Arabic

The Arab world enjoys a wide array of dialects, which are the non-standard varieties of Arabic natively spoken and increasingly written on social media across the Arab world, i.e., *informal speech*. Over the centuries, as Arabic evolved alongside different middle-eastern and north-African languages, these languages had effects on each other in a way that would result in many of the varieties which we call dialects today. More recently, and with the rise of technology, and before the advent of smartphones and the plethora of keyboards that come embedded in them, it was difficult to type using the Arabic script. Therefore, people mostly used Roman script to communicate online using their dialects in their day-to-day interactions. This includes digits to account for phonemes which would be tough to convey using only roman letters. Writing this way is referred to as *Arabizi* (Arabic chat alphabet). To this day, many people still prefer to write this way despite the availability of Arabic keyboards on all smartphones.

## Diglossia

In contrast to MSA, the dialects are not taught at school, and are generally lumped together in the same "dialectal" basket as derivatives of MSA by people with no linguistic background, including natives. Some people also seem to think that the dialects are "incorrect" forms of MSA, as is the case with pidgin or creole languages, although the situation is quite different here. What native Arabs speak when they use their dialects seems to be quite different from what they perceive themselves as speaking (Ferguson, 1990). When they do, they sometimes cannot seem to dissociate between the said dialect and MSA, as in their minds, it is as if both are fused together in a kind of complementary symbiosis. And as these dialects were not commonly written for a long time, this helped feed into the layman hypothesis that they merely constitute abominations of their more standard progenitor, namely, MSA. This is also due to the fact that the use of informal speech (the dialects) and MSA is so intertwined – as imposed by day-to-day life – that it becomes so intuitive as to shun out the differences between them. This conundrum is linguistically referred to as *diglossia* (Shendy, 2019). For speech purposes though, most people – unknowingly – identify to their dialects and not to MSA. It is important to highlight that using DA and MSA interchangeably sometimes is not a matter of register shift, but should be regarded as code-switching because a big number of grammatical and phonological constructions in either variety completely clash with that of the other.

## Dialect Categories

In light of that, the *dialect* nomenclature is quite controversial among researchers as there has been ample evidence that is in favor of treating some of these varieties as individual languages. Many varieties of DA are not even mutually intelligible. As such, the process of determining how many dialects exist and how close they

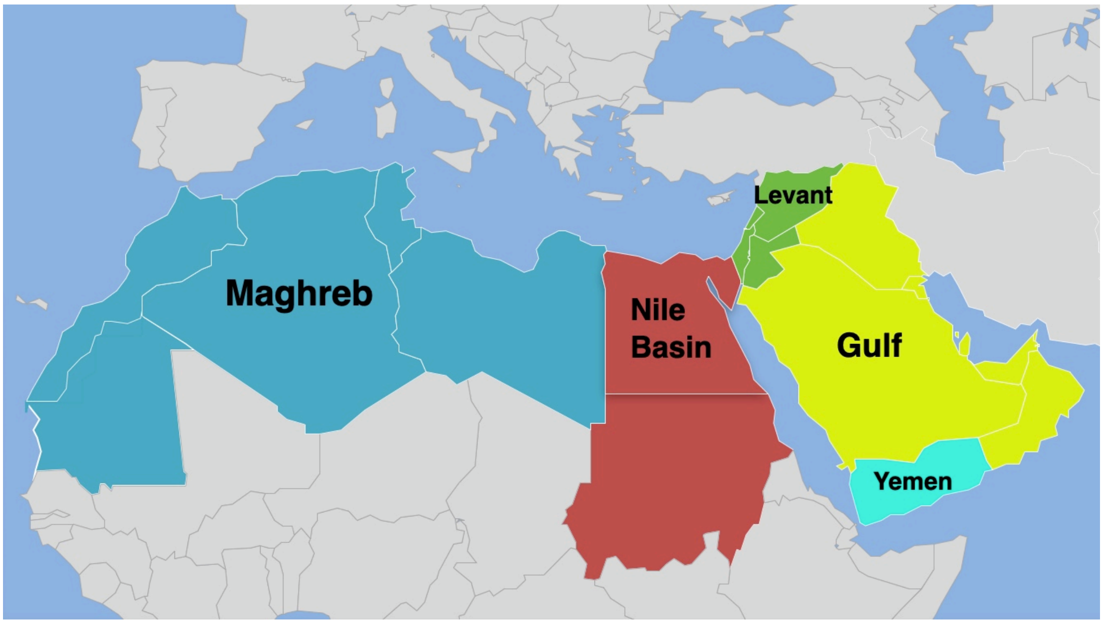


Figure 1: Map showing the different geographical regions of the dialects (template from `https://yourfreetemplates.com/free-mena-region-map-template/`).

are to each other and to MSA has been a daunting task. There are many schemes which attempt to group the dialects geo-linguistically, however, in this report, the scheme presented in Bouamor et al. (2018) is used for dataset compatibility reasons, and it divides the dialects both in a fine-grained (25 sub-dialects) and coarse-grained (5 regions) manner, namely (see Figure 1 for reference):

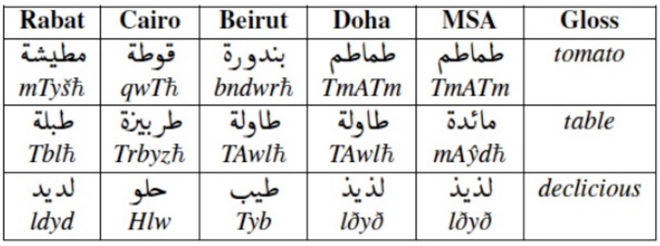
- Maghreb (Morocco, Algeria, Mauritania, Tunisia, Lybia)
- Nile Basin (Egypt, Sudan)
- Levant (Lebanon, Syria, Palestine, Jordan)
- Gulf (Iraq, Saudi Arabia, UAE, Qatar, Kuwait, Bahrain, Oman)
- Yemen

The differences between these dialects take place at all linguistic levels, the most prominent being the phonological, morphological, lexical, and syntactic levels, in decreasing order of importance, and they all differ at varying degrees from MSA (Zaidan et al., 2014). Table 1 shows a snippet of that variation, in which lexical changes from one dialect to the other are displayed. Additionally, Figures 2 and 3 give an idea about the morpho-syntactic changes that can be observed across the dialects or between them and MSA. The most notable changes – excluding the numerous phonetic and phonological shifts (colored in green), which is quite common for dialects – include[7]:

- The lexical change of the negative particle from لَمْ /lam/ ‘did not’ in MSA, to ما /maː/ for all dialects (Figures 3a, 3b, and 3c).

---

[7] Adapted from Bassiouney (2020).



| Rabat | Cairo | Beirut | Doha | MSA | Gloss |
|---|---|---|---|---|---|
| مطيشة<br>*mTyšħ* | قوطة<br>*qwTħ* | بندورة<br>*bndwrħ* | طماطم<br>*TmATm* | طماطم<br>*TmATm* | *tomato* |
| طبلة<br>*Tblħ* | طربيزة<br>*Trbyzħ* | طاولة<br>*TAwlħ* | طاولة<br>*TAwlħ* | مائدة<br>*mAŷdħ* | *table* |
| لديد<br>*ldyd* | حلو<br>*Hlw* | طيب<br>*Tyb* | لذيذ<br>*lðyð* | لذيذ<br>*lðyð* | *declicious* |

Table 1: Lexical correspondences between four urban Arabic dialects and MSA. Taken from Erdmann et al. (2018). Uses the Habash-Soudi-Buckwalter transliteration scheme[a]. See Appendix A for the mapping.

---

[a] Unless otherwise specified, all transliterations in this report are realized using the Habash-Soudi-Buckwalter transliteration scheme.

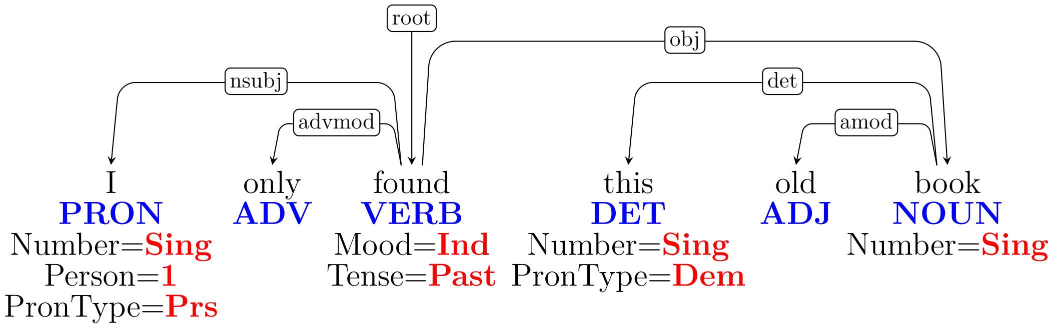


Figure 2: Dependency tree of the English sentence 'I only found this old book', using the UD v2.6 style (`https://universaldependencies.org/guidelines.html`).

- Along with the new negation particle, the usage of the perfective aspect (ماضي) to express the past tense instead of the imperfective (مضارع) aspect and jussive mood (مجزوم). This is a pattern across a large part of the dialects.

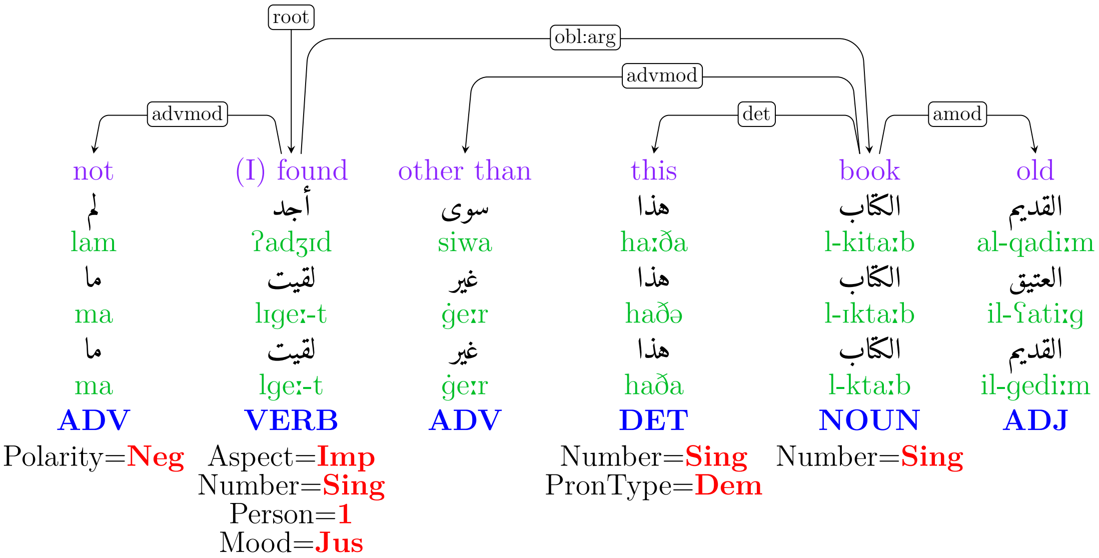
- The lexical change of the main verb in the phrase from إيجاد /iːdʒaːd/ 'to find' in MSA, to لقاء /liqaʔ/ 'meeting' in DA. Using the former while speaking in DA would be considered code-switching (Figures 3a, 3b, and 3c).

- The lexical change of the negative adverbial modifier سوى /siwaː/ 'except' in MSA, to غير /ġeːr/ 'other than' in Iraqi and Saudi Colloquial Arabic, and to إلاّ /ılla/ 'except' in Tunisian and Lebanese Colloquial Arabic (Figures 3a, 3b, and 3c).

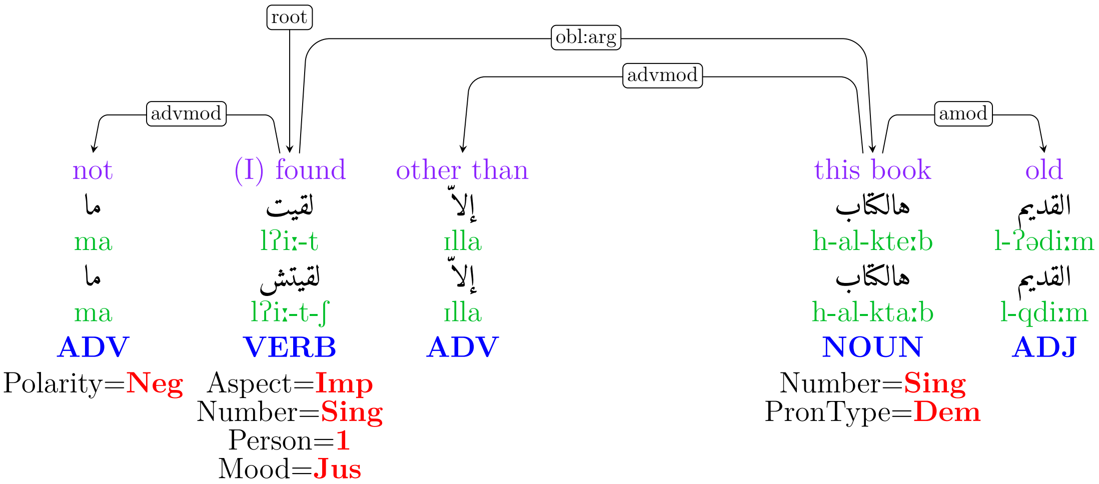
- The cliticization of the determiner هذا /haːða/ 'this' into the noun for Tunisian Colloquial Arabic and Lebanese Colloquial Arabic (Figure 3b).

- The addition of the enclitic ش /ʃ/ in Tunisian and Egyptian Colloquial Arabic, which along with the negative particle ما /maː/ forms one unit (Figure 3b and 3c).

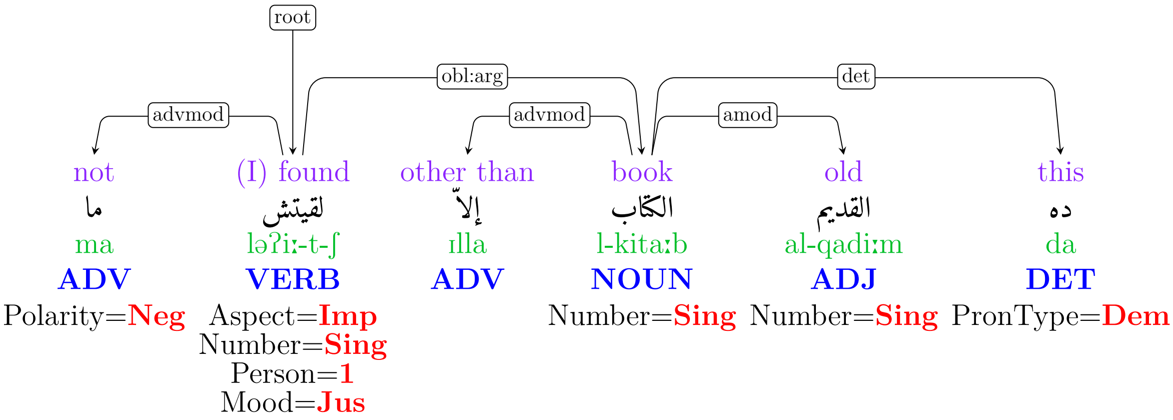
- The postponement of the determiner هذا /haːða/ 'this' in MSA to after



(a) (1) MSA, (2) Iraqi Arabic, and (3) Saudi Arabic, respectively across the rows.



(b) (1) Lebanese Arabic and (2) Tunisian Arabic, respectively across the rows.



(c) Egyptian Colloquial Arabic translation

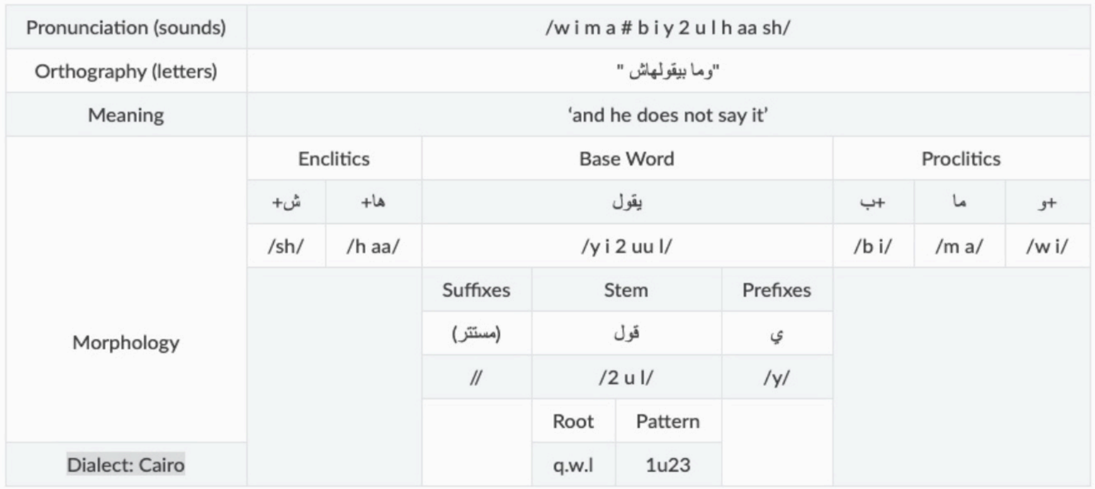
Figure 3: Dependency trees of the sentence ‘I only found this old book’ translated into MSA and multiple dialects using the UD v2.6 style (`https://universaldependencies.org/guidelines.html`).

the noun and its lexical change to ده /da/ in Egyptian Colloquial Arabic (Figure 3c).

In reality, the subdivision into 25 sub-regions does not depict the full picture as within each country, there is also a considerable amount of variation, although it limits itself to stylistic and lexical variation. Hence, the sub-regions were chosen so as to reflect the most prominent variant of the latter, usually the dialect of the capital.

## Dialectal Arabic Grammar

The English language has lost most of its grammatical features which were present in Old and Middle English, features such as case, gender, number markers, etc. On the same note, the grammatical inflection systems of Romance languages, i.e., French, Italian, Spanish, etc., which are direct descendants of Classical Latin, have been simplified as they lost most of the complex case structures found in it. To the extent that DA is a variant of the pluricentric MSA, it has not adopted many of the grammatical features that make up MSA morphology and syntax. Even though the contrast between the Arabic dialects is not as big as for the Romance languages, it remains non-trivial. Nevertheless, the dialects are still morphologically rich, and utterances will consist of a base word and various sets of affixes and clitics which will often denote number, gender, negation, aspect, etc. (see Table 2 for reference).



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Pronunciation (sounds) | /w i m a # b i y 2 u l h aa sh/ | | | | | | | | |
| Orthography (letters) | " وما بيقولهاش" | | | | | | | | |
| Meaning | 'and he does not say it' | | | | | | | | |
| Morphology | Enclitics | | Base Word | | | | Proclitics | | |
| | +ش | +ها | يقول | | | | +ب | ما | +و |
| | /sh/ | /h aa/ | /y i 2 uu l/ | | | | /b i/ | /m a/ | /w i/ |
| | | | Suffixes | Stem | | Prefixes | | | |
| | | | (مستتر) | قول | | ي | | | |
| | | | // | /2 u l/ | | /y/ | | | |
| | | | | Root | Pattern | | | | |
| Dialect: Cairo | | | | q.w.l | 1u23 | | | | |

Table 2: Morphological segmentation of an Egyptian Arabic utterance. Taken from https://camel-guidelines.readthedocs.io/en/latest/orthography/.

These kinds of differences also occur inter-dialectally and contribute to the potential unintelligibility between them. All of this is to demonstrate the intricacies that lie in trying to compare DA with MA, and this is just scratching the surface.

## Code-switching between MSA and DA

Code-switching between DA and MA is common if not natural. Different situations call for different registers (degrees of code-switching), and there have been many models which have tried to quantitatively delimit the boundaries of this phenomenon (Bassiouney, 2020). Nevertheless, they remain blurry, and the code-switching that might be encountered in either text or speech data adds yet another layer of complexity to processing DA. Because some elements of syntax change between DA and MSA, and because the changes vary from one dialect to the other, one is faced with tough design choices when it comes to accommodating for these dialects by sharing resources with MSA. Except for MSA, there are no references such as the Académie française for French which come to draw the line between the different dialects and registers within a dialect in order to classify something as orthographically and grammatically sound or not. In summary, lack of standardization within a language drastically increases the degree of complexity for NLP applications.

## Tools for DA

As a consequence of these differences, the tools developed for MSA work very poorly on DA. Until very recently, running Google Translate on a Levantine Arabic sentence to translate it into English would have yielded a very gruesome result. These days, it fairs much better at translating a DA sentence, although there is still a long way to go. While there has been considerable research dedicated to DA, especially Sentiment Analysis and Dialect Identification, there has been relatively little effort that directly addresses one of the most central problems in processing DA textual data, which is the lack of a standard orthography system. In addition to all the aforementioned differences between MSA and DA, the fact that DA does not have a standard orthography probably tops them all.

# Chapter 1

# Problem Statement and Task Definition

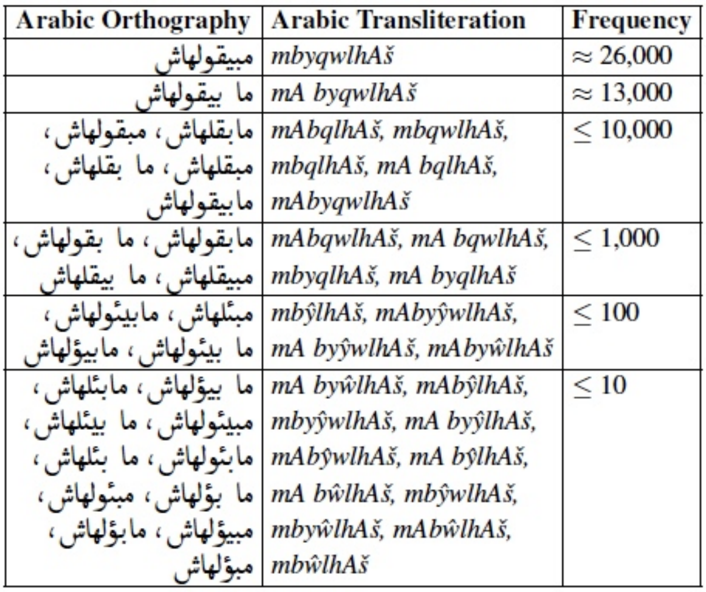
## 1.1 The Spontaneous Orthography Problem

Text and speech are two fundamentally different communication media. Speech is inherently loaded with considerably more content than text as it contains sounds and prosody. Trying to describe sounds using an alphabet is not an exact science, and especially to laypeople. This is distinctly true when the said alphabet leaves it to the reader to infer what vowels are to be produced in the corresponding morpho-syntactic context, as is the case with Arabic. Although it is said that Arabic has for the most part a phonemic (shallow) spelling system[1] like Czech or Spanish (Habash, 2010), this is actually very far from the reality of it, since vowels are for the most part specified by diacritics which are omitted as a common practice. It is left to the reader to infer the latter from the morpho-syntactic context. As an analogy, it is not easy for a beginner of English – which has a very deep (opaque) orthography – to correctly pronounce written words. The grapheme [a] in American English is mapped to many different phonemes like in *jail* /ɛ/, *ear* /i/, *pale* /a/, *bat* /æ/, or *paw* /ɔ/. When diacritics are used in Arabic, this very rarely happens. However, the many phonological differences between MSA and DA, compounded by the omission of diacritics, makes the confusion an order of magnitude larger as to how things should be spelled in DA.

Now, one cannot disregard the great amount of linguistic features shared by MSA and DA despite their big differences. Because DA resources are for the most part noisy and unstructured, it is quintessential to harmonize them in some meaningful way. The main reason why this problem is currently being faced is that DA lacks a standard orthography system, since it is mainly spoken. When written, it is left to the writer to make orthography choices based on an intricate interplay between the phonology of a word, and its closeness to its respective cognate in MSA when available. This will in turn produce various orthographic realizations of the same word, which will tend to vary between subjects and within subjects.

---

[1] In some very few cases, spelling is morphemic/lexical, in which case the graphemes do not faithfully represent the pronunciation, such as with the plural verb suffix (واو الجماعة) وا wA /u/, and the silent *t* (*T-marbuta*) ة $\hbar$ /t, $\emptyset$/.



| Arabic Orthography | Arabic Transliteration | Frequency |
|---|---|---|
| مبيقولهاش | *mbyqwlhAš* | ≈ 26,000 |
| ما بيقولهاش | *mA byqwlhAš* | ≈ 13,000 |
| مابقلهاش، مبقولهاش،<br>مبقلهاش، ما بقلهاش،<br>مابيقولهاش | *mAbqlhAš, mbqwlhAš,*<br>*mbqlhAš, mA bqlhAš,*<br>*mAbyqwlhAš* | ≤ 10,000 |
| مابقولهاش، ما بقولهاش،<br>مبيقلهاش، ما بيقلهاش | *mAbqwlhAš, mA bqwlhAš,*<br>*mbyqlhAš, mA byqlhAš* | ≤ 1,000 |
| مبئلهاش، مابيئولهاش،<br>ما بيئولهاش، مابيؤلهاش | *mbŷlhAš, mAbyŷwlhAš,*<br>*mA byŷwlhAš, mAbyŵlhAš* | ≤ 100 |
| ما بيؤلهاش، مابئلهاش،<br>مبيئولهاش، ما بيئلهاش،<br>مابئولهاش، ما بئلهاش،<br>ما بؤلهاش، مبئولهاش،<br>مبيؤلهاش، مابؤلهاش،<br>مبؤلهاش | *mA byŵlhAš, mAbŷlhAš,*<br>*mbyŷwlhAš, mA byŷlhAš,*<br>*mAbŷwlhAš, mA bŷlhAš,*<br>*mA bŵlhAš, mbŷwlhAš,*<br>*mbyŵlhAš, mAbŵlhAš,*<br>*mbŵlhAš* | ≤ 10 |

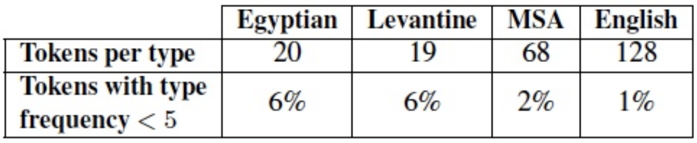
Table 1.1: 27 encountered ways to write the Egyptian Arabic utterance ما بيقولهاش *mAbyqwlhAš* /mabiʔulhaːʃ/ ‘he does not say it’ and their frequencies from Google Search. Taken from Erdmann et al. (2018). Uses the Habash-Soudi-Buckwalter transliteration scheme. See Appendix A for the mapping.

As such, the quality of data available for research purposes will mainly come in the form of very noisy social media web scrapes. To make it even more difficult to process DA, in addition to MSA-DA code-switching, the data is riddled with foreign language infusions (e.g., English, French, etc.).

For example, Habash, Eryani, et al. (2018) gives the results of analyzing the occurrence of the Egyptian Arabic utterance ما بيقولهاش *mA byqwlhAš* /mabiʔulhaːʃ/ ‘he does not say it’ via Google Search, and 27 different ways of writing this utterance were recorded (see Table 1.1). All these spellings faithfully represent the utterance phonetically, but because such an utterance does not exist in MSA, either morpho-syntactically or phonologically, then there is no way for people to know what the correct version is. Additionally, one can also take a look at the difference between the token-to-type ratio of two dialect varieties, MSA, and English (see Table 1.2). The low token-to-type ratio of the two DA varieties compared to MSA goes directly to show that there is a great deal of noise in the data which is not only due to rich morphology, but to the spelling inconsistencies of different words (Erdmann et al., 2018). From here on, this phenomenon shall be referred to as Spontaneous Orthography (SO) (Eskander et al., 2013).

## 1.2 CODA*

To remedy the SO problem, a Conventional Orthography for Dialectal Arabic (CODA*) was formulated as an attempt to provide a unified framework for future standardization efforts (Habash, Eryani, et al., 2018). It is used to specify how words from different varieties of DA are to be spelled in a standard way. While this might seem like a daring move, as many orthography systems have been proposed



| | Egyptian | Levantine | MSA | English |
|---|---|---|---|---|
| **Tokens per type** | 20 | 19 | 68 | 128 |
| **Tokens with type frequency < 5** | 6% | 6% | 2% | 1% |

Table 1.2: Token and type -based comparisons between two dialects of Arabic, MSA, and English in corpora of 13 million words each. Taken from Eryani et al. (2020).

for DA in the past and none have them have gained widespread recognition, CODA* was put together exclusively for ad-hoc reasons and was not intended to become an orthography reference. The aim of this scheme can best be summarized by the goals and design principles boasted by its creators:

1. Consistent and coherent convention for writing DA
2. Uses the Arabic script
3. Unified framework for writing all dialects
4. Balance between level of dialectal uniqueness and MSA-DA similarities
5. Easily learnable and readable (for high inter-annotator agreement)
6. Ad hoc convention (for computational purposes and not widespread use)
7. Comparable to English spelling (phonological, historical, with some exceptions)
8. Unique DA orthography, representative of its morpho-phonology
9. MSA-like orthographic decisions
10. Preserves the phonological form of dialectal words and morpho-syntax

Hence, this convention has been optimized to maximally expand the array of possibilities for processing DA. The authors do this while trying to strike a balance between trying to position the dialects relatively to MSA, all the while preserving their uniqueness. This convention will be helpful in setting some design choices later on in this thesis, as what it tries to achieve, is the automation of the conversion process from raw dialectal input to this CODA* form.

The MADAR CODA corpus was released recently (Eryani et al., 2020) and consists of 2000 parallel sentences for each of the five dialect region variants (one representative sub-region is picked for each). The statistics of this dataset shed some light on the different features of a corpus before and after being standardized to the CODA* form. Figure 1.3a shows for example that on average, 14% of tokens seen in a DA corpus will have to be standardized[2], while the remaining would already be in a standard form. This percentage varies from one dialect to the other, thus, underlining the difference between them, and more interestingly,

---

[2] This dataset has a relatively low amount of noise. Datasets from social media come with considerably more SO

| | No Edit | Sub | Split | Del |
|---|---|---|---|---|
| **Beirut** | 81.28% | 17.35% | 1.38% | 0.00% |
| **Cairo** | 85.98% | 12.44% | 1.54% | 0.03% |
| **Doha** | 94.90% | 4.79% | 0.30% | 0.01% |
| **Tunis** | 85.48% | 12.66% | 1.79% | 0.07% |
| **Rabat** | 83.66% | 14.67% | 1.66% | 0.01% |
| **Average** | **86.26%** | **12.38%** | **1.33%** | **0.02%** |

(a) Raw-to-CODA edit statistics in token space

| | Raw Types | CODA Types | Type Overlap | Vocabulary Reduction |
|---|---|---|---|---|
| **Beirut** | 4,114 | 3,877 | 80% | 6% |
| **Cairo** | 4,114 | 3,820 | 84% | 7% |
| **Doha** | 3,417 | 3,377 | 94% | 1% |
| **Tunis** | 4,044 | 3,834 | 84% | 5% |
| **Rabat** | 4,225 | 4,000 | 83% | 5% |
| **Average** | **3,983** | **3,782** | **85%** | **5%** |

(b) Vocabulary size in number of types and type overlap between the Raw and CODA corpora for each dialect.

Table 1.3: Statistics collected for the MADAR CODA corpus (Eryani et al., 2020)

their closeness to MSA, which is translated by a lower percentage. The closer these dialects are to MSA, the more clear-cut the orthographic decisions will be to the writers, hence, the lower the amount of SO. Figure 1.3b quantifies the reduction in vocabulary after standardization.

## 1.3 Spontaneous Orthography Taxonomy

One thing which Habash, Eryani, et al. (2018) do not make available is a taxonomy of SO, despite providing general yet discriminating rules on how to standardize it. This thesis will contribute towards that, although at the restricted scale of a single variety. The big question is however, what should and what should not be standardized. In other words, when is something considered as SO, and when is it considered as a plain typographic mistake, in which case we step into the realm of spelling correction. Spelling correction is a sizeable field in and by itself, and drawing the line between it and SO standardization is essential. Hence, this taxonomy definitely does that, and more. The top-most categories are laid out in this section (Figure 1.1) while the core of the taxonomy is presented in Chapter 4. It is loosely based on that of Himoro et al. (2020), as there will be four nested axes along which different categories will branch out, namely:

- **Intentionality**: Branched out from the *Spelling Inconsistencies* root category, it separates inconsistencies based on whether the writer was aware of the inconsistency or not, i.e., *Intentional Orthography* and *Unintentional Orthography*.
- **Awareness**: Branched out from *Unintentional Orthography*, it separates the latter based on whether unintentionality was due to lack of attention or to an absence of widely recognized orthography rules, i.e., *Random Errors* and *Non-random Errors*, or what we refer to here as SO.
- **Origin**: Branched out from *Spontaneous Orthography*, it separates the latter based on whether the standardization to be applied can be done using simple (MSA-existent) rules or not, i.e., *Regular Rules* and *Arbitrary Rules*.
- **Phonetic Plausibility**: Branched out from *Arbitrary Rules*, it separates the latter based on whether what was written is phonetically realizable

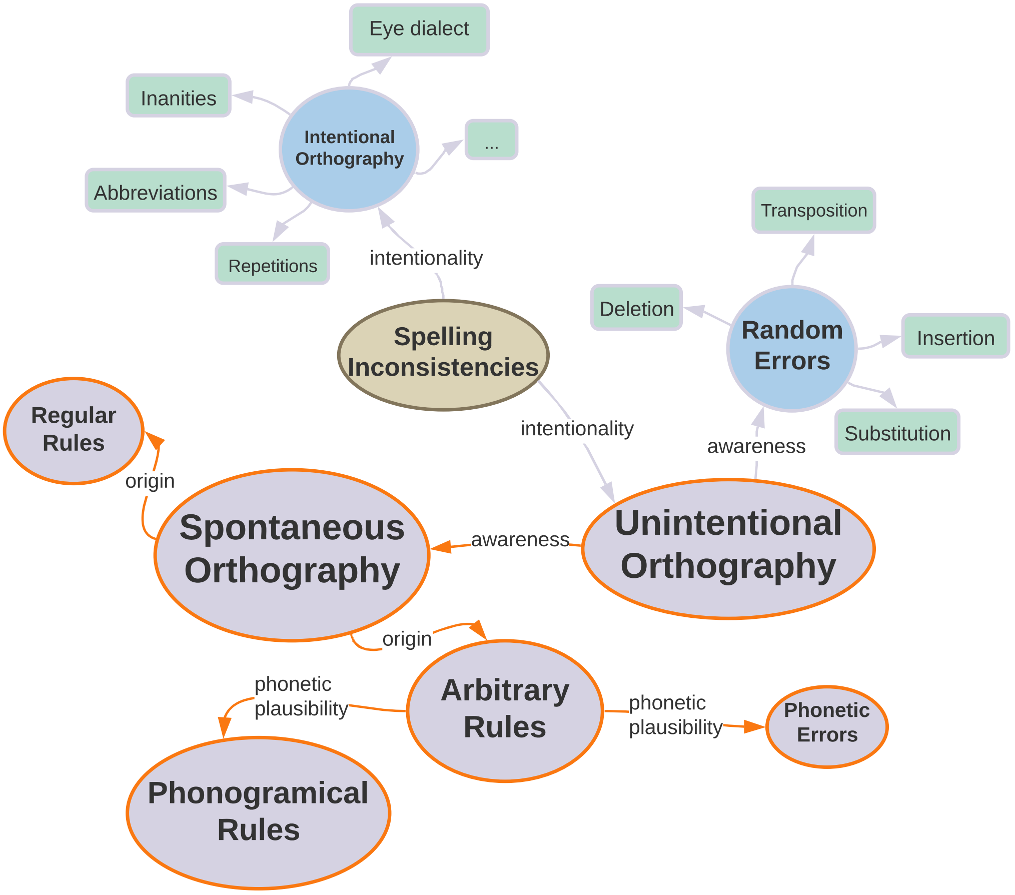


Figure 1.1: Summary of the Spelling Inconsistencies taxonomy which will be used throughout this thesis. Generated using `https://lucidspark.com/`.

given the utterance or not, i.e., *Phonogramical Rules* and *Phonetic Errors*. All complicated cases in SO lie in the *Phonogramical Rules* category.

Creating such taxonomies relies heavily on the state of mind of the writer at the time of writing, and it is essential to sort out different types of inconsistencies in order to know what should be corrected and what not, as this is not a spelling correction task. The standardization task is a non-trivial one, especially due to richness in morphology. For example, روحو *rwHw* /ruːħo/ as it is written might be interpreted as ‘go (command, plural)’ in which case it would have to be standardized as روحوا *rwHwA* since وا *wA* is the correct suffix for the masculine, plural, command verb. However, it could equally be interpreted as ‘his soul’, in which case it would be standardized as روحه *rwHh*, as the silent ه *h* denotes the masculine possessive pronoun ‘his’. Note that both words, before being standardized, are homographs and homophones. The decision that the system must make must therefore be contextually informed in order to have a chance at inferring the intent of the writer. All of this, to say that some design choices will be dedicated to this issue with the help of the taxonomy.

## 1.4 Literature Review

### 1.4.1 Previous DA Research on Spontaneous Orthography

| Task | Works | Orthography Standardization |
|---|---|---|
| Parsing | Chiang et al. (2006) | Manual annotation |
| Machine Translation | Zbib et al. (2012) | Pre-processing using a morphological segmentation tool; MAGEAD (Habash and Rambow, 2006) |
|  | Sawaf (2010)<br>Salloum et al. (2012)<br>Sajjad et al. (2013) | Converting DA input to closest MSA forms using dialect-specific morphological analyzers, handcrafted rules, and character-based n-gram models |
| Morphological Disambiguation | Khalifa, Hassan, et al. (2017) | Manual annotation |
|  | Zalmout, Erdmann, et al. (2018) | Character and word-level embeddings, embedding space mappings, and edit-distance weights |
| Morphological Segmentation | Zbib et al. (2012)<br>Samih et al. (2017) | No pre-processing |
| Pre-trained Embeddings | Abdul-Mageed et al. (2020) | No pre-processing |

Table 1.4: A sample of works in DA processing and how SO was indirectly handled for each of them.

**Addressing SO Indirectly** By indirectly, what is meant is that standardization was not the main output of the system, but was handled as a pre-processing step. As can be seen from Table 1.4 which is far from complete in terms of works, but more or less representative in terms of SO handling approaches, most attempts at handling SO have been limited to manual annotation, the use of morphological disambiguation tools for DA which usually rely on hand-crafted rules and shallow Machine Learning techniques, and finally, making use of the closeness between DA and MSA by using the latter as some sort of interlingua by and from which to transfer knowledge. Most notably, Zalmout, Erdmann, et al. (2018) address the noise indirectly through a morphological disambiguation task by making use of character- and word-embedding space mappings in conjunction with string similarity edit distance weights to rank closeness, and their system is shown to perform “as well could be expected without orthographic inconsistency”. Even though their results look promising, there was no way to investigate further into their work as access to both the dataset they used and their developed system was restricted. Furthermore, as introduced in Section 1.5.2 and explained in detail all throughout the report, good results on a dataset do not translate to good real-world performance.

**Addressing SO Directly** Other efforts have strived to directly address SO. Dasigi et al. (2011) use word clustering to identify variants of the same word using context similarity and Eskander et al. (2013) use multiple independent character classifiers to make conversion decisions across some pre-defined word and character features. Erdmann et al. (2018) train multi-dialectal word embeddings in order to facilitate bilingual dictionary induction, a crucial step in developing resources for any language. Watson et al. (2018) also deals with the spelling correction task directly, but for MSA, using a hybrid word-character-level bi-GRU NMT architecture to translate to a corrected form. In what can be considered closest to the present work and as an extension of their 2018 study mentioned above, Zalmout and Habash (2020) train a joint model (weight sharing) to predict both lexical features, i.e., normalization (standardization to CODA* form), diacritization, and lemmatization, and non-lexical features, i.e., morphological tagging. Their architecture is similar to one of the solutions proposed in this work.

**Incorporating Noise in the Model** Still, others choose to work with the data as is, with minimal pre-processing, incorporating the noise into the model (Samih et al., 2017), and more recently, building BERT embeddings out of a noisy multi-dialectal DA corpus (Abdul-Mageed et al., 2020). The latter has been shown to improve the performance of many tasks such as Dialect Identification and Sentiment Analysis.

Far from all of the above, a great deal of the research on DA has generally tended to revolve around Dialect Identification, resource creation, corpus and lexicon creation, and to a lesser extent, morphological disambiguation. Elnagar et al. (2021) in their recent survey on DA research notably point out that the amount of work that was put into research on identification is disproportionate when compared to resource creation, while Arabic dialects are still considered as heavily under-resourced languages. Another thing the survey draws our attention to is the scarcity of experimentation with Deep Learning techniques for DA, despite the increasing number of available data. As will be pointed out later on, increasing the amount of resources might still not be enough to achieve considerable advances in the state-of-the-art for DA.

### 1.4.2 Spelling Correction

In our efforts to quantify and try to ease SO, it might be beneficial to treat our problem, at least conceptually, as a spelling correction task, and much has been done in that regard. Automatic Spelling Correction (ASC) systems generally rely on three main components which are summarized in Equation (1.1) and Figure 1.2 (Hládek et al., 2020):

1. **Dictionary**: Proposes correction candidates $w_i \in C(s) \subset W$ for a given token $s$ from a list of all possible words $W$.
2. **Error Model**: Expresses the similarity between string $s$ and $w_i$. It usually comes in the form of a similarity measure (Levenshtein distance, LCS, Soundex, edit distance, etc.), and more recently in the form of neural

seq2seq models (bi-LSTM, RNN language model, Transformer). It is represented by $P(s|w_i)$ in (1.1).

3. **Context Model**: focuses on boosting the best candidate based on the context in which $s$ is. It is represented by $P(w_i)$ in (1.1).

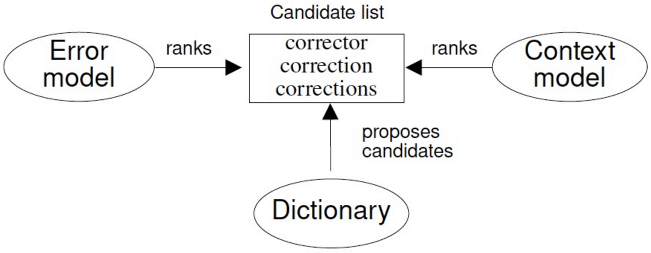


Figure 1.2: Components of an ASC system. Taken from Hládek et al. (2020).

$$w_b = \arg\max_{w_i \in C(s)} P(s|w_i)P(w_i) \tag{1.1}$$

According to Hládek et al. (2020), an architecture that has been especially successful over the recent years is one where we would translate from a noisy input to a corrected form using a Neural Machine Translation (NMT) setting at the character level, hence, with character sequences of words (Himoro et al., 2020; Etoori et al., 2018). For languages with large datasets, similar architectures have become the de-facto way of going about ASC.

Some work has also been carried out for Arabic ASC such as Attia et al. (2014) and Watson et al. (2018) for MSA, and Farra et al. (2014) for Egyptian Arabic. Both leverage annotated corpora and try to solve the ASC task (Equation 1.1), and the latter utilizes an old version of CODA*. At this point, it is not very useful to discuss the metric results of those studies, even though they are closely related to this thesis, since they both utilize different datasets, and work on different domains.

## 1.5 Methodology

In this section, the task at hand will be defined as it it hides many caveats. Then, an overview of the work performed will be structured chronologically to highlight the obstacles which were encountered in the efforts put forth towards engineering a viable solution for SO, all of this, while taking previous research and how the system would fare against real-world data into account. We start by presenting the variety which we will be dealing with throughout the thesis, namely, Lebanese Arabic.

### 1.5.1 Task Definition

When we say Lebanese Arabic, what is meant is the multiple regional varieties within Lebanon (Makki, 1983) as seen in Figure 1.3. Although Lebanon is a

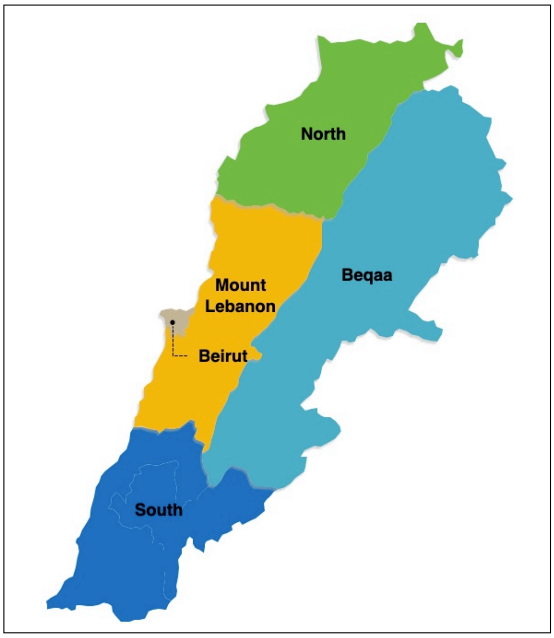


Figure 1.3: Map showing a rough segmentation of how the “dialectal” regions are divided in Lebanon.

very small country, the multiplicity of regional dialects within it is remarkable. It is not always clear where one ends and the other begins due to the small size of the country and the geographical proximity of the people, and consequently, the frequent intermingling of the different groups. Highlighting this is critical, because as we shall see, datasets can very often be non-representative of real-world data and hold biases. This is especially true for DA datasets, because the lines between the dialects are blurry and there is significant variation within a country’s own dialects. Not only that, but very often, there is even intra-subject variation in the way locals speak.

For the sake of this thesis, the scope shall be restricted to a single DA variety, namely, Lebanese Arabic[3], which is in turn part of the Levantine Arabic regional variety, along with Syrian, Palestinian, and Jordanian Arabic. The standardization task then consists of a system which would take in a text, the contents of which would have already been classified as belonging to the Lebanese Arabic variety (either manually or using a Dialect Identification module), and would standardize it into its CODA* form as seen in Figure 1.4.

At this point, there are a few points that should be elucidated. When one says Spontaneous Orthography, the term is very broad and we should be more precise in order to make the task clearer. As it is not in the intent of this thesis to perform spelling correction per se, we shall restrict the scope of “error” correction to unintentional, non-random errors, i.e., SO, as described by Section 1.3 and seen in Figure 1.1. In other words, intentional errors (e.g., abbreviations, inanities,

---

[3] From now on, Lebanese Arabic shall refer to all of the regional varieties combined as seen in Figure 1.3. In some datasets such as the MADAR CODA Corpus (Eryani et al., 2020), Beirut (the captial of Lebanon) is chosen as a representative of the whole Levant region, although, even within Lebanon, there is great variety outside of the “proto-Beiruti” dialect. Care must be taken not to confuse between those.

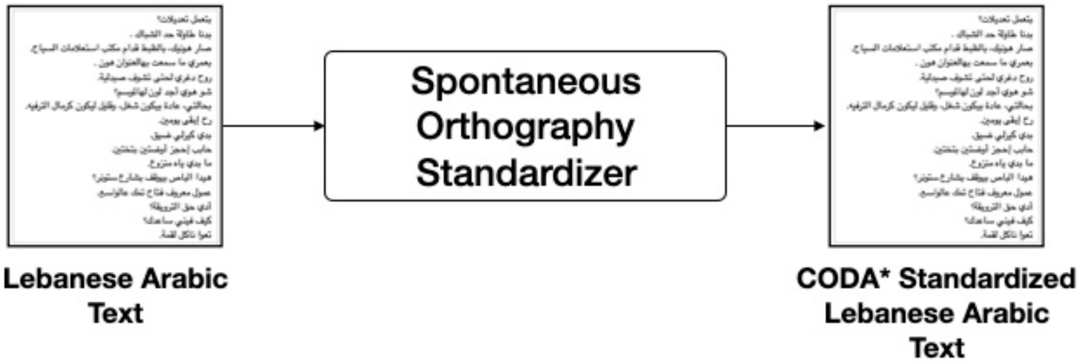

Spontaneous Orthography Standardizer
Lebanese Arabic Text
CODA* Standardized Lebanese Arabic Text

Figure 1.4: Diagram showing a high-level description of the proposed system's functionality.

etc.) and unintentional random errors, i.e., mistakes due to lack of attention or knowledge of the writer shall not be accounted for, meaning that the performance of our system shall not be evaluated depending on those cases.

Seeing that we are working in a low-resource environment, a logical step forward would be to increase the amount of data resources we have. However, doing so is not bound to solve the problem, due to the simple predicament which drives this thesis. Let us assume that we want to train a system to solve a specific task for DA (e.g., POS tagging). The system which will be put together will be trained on a some DA corpus which will most probably be pre-processed and its orthography standardized (most probably in a manual way). However, the data that it will be subjected to at inference time in the real world will have SO. Hence, increasing the amount of resources is not a solution.

One might then suggest to build our systems in such a way as to incorporate the noise distribution into the model. However, most recent state-of-the-art systems that tend to leverage different kinds of pre-trained models such as BERT (Devlin et al., 2019), and which incorporate a huge amount of prior information, and which can be fine-tuned to (supervised) datasets of much smaller sizes, have been shown not to be robust when it comes to dealing with noisy data (Jin et al., 2020; Alabi et al., 2020; Heigold et al., 2018), and hence they are not optimal and sometimes not even suitable for DA data.

Therefore, what this thesis proposes is three-fold. First, the aim is to gain a better understanding of the problem at hand – as it has been rather underrepresented in the past – by studying the distribution of inconsistencies and getting up-close and personal with them. Second, it will put forward a much needed taxonomy which will account for inconsistencies that are idiosyncratic to Lebanese Arabic and the skeleton of which could be extended to other varieties. And finally, it will propose a system which attempts to reduce the amount of noise in Lebanese Arabic data by leveraging multiple neural architecture, all of this, hoping that the methods used will be scalable to the other DA variants.

### 1.5.2 Chronological Overview

Section 1.4 outlined the most prominent studies that previously dealt with SO. However, most of them use different datasets and there is no guaranteeing that the datasets they used were adequate in terms of the amount or quality of noise they capture. On another hand, it is very hard for deep learning models to generalize predictions under a low-resource setting as they are data-hungry. This is exacerbated by the morphological richness and degree of ambiguity of DA, and last but not least, the lack of standard orthography. In this section, the followed methodology to surpass this is explained.

In the early stages, different standardization models were developed. These models are generally variations of the regular sequence-to-sequence (seq2seq) Neural Machine Translation (NMT) architecture which consists of an encoder, decoder, and attention mechanism. They fall under three classes, namely, *word-level* models which deal with word-level representation of data, *character-level* models which deal with character-level representations, and *hybrid* models which do both.

**Bad Performance on Real World Data** The best models are the hybrid character-level models, and specifically the ones which make use of context. However, it soon became clear that the good performance on development sets would not translate well to similar performance on real world data, mainly due to the size and coverage of the available datasets. For example, the 27 spellings of the Egyptian Arabic utterance ما بيقولهاش /mabiʔulhaːʃ/ ‘he does not say it’ in Table 1.1 were fed into the best model trained on the Cairo (Egypt) portion of the MADAR CODA corpus. Without morphological context, the system was not able to standardize any of the different forms even though the accuracy achieved on the development set surpassed that of previous studies. Hence, the available datasets coupled with the available resources in our possession (morphological models, etc.) do not make for good training candidates.

**Corpus Annotation** This realization made the thesis take a different turn, and it was decided that a new corpus would be annotated for the following reasons:

- First and foremost, it would be an opportunity to release an annotated corpus (for morphology and CODA* standardization) solely for Lebanese Arabic which would otherwise be hard to find publicly.
- The available corpora are non-comprehensive and non-representative of real world noise, and require data augmentation at the source side (noisy text). Furthermore, their current format makes augmentation difficult since no already existing treebank annotates for both the CODA* form and morphological segmentation (and tagging) at the source side. The hypothesis here is that having a corpus which is both annotated for morphological segmentation (at the source side) and orthography standardization (at the target side) would greatly benefit the standardization model. The available DA corpora fall under the following categories:

  - they are annotated for orthography standardization but are not morphologically segmented (Eryani et al., 2020)
  - they are annotated for orthography standardization and morphological segmentation, but at the target side, i.e., only the CODA* standardized versions are annotated for segmentation (Khalifa, Habash, et al., 2018). Performing morphologically-driven augmentation (Saito et al., 2017) to make source-side noise more abundant for training is difficult.
  - they are just annotated for morphological segmentation at the source side and no CODA* standard forms are provided as gold labels (Darwish et al., 2018).

- It will facilitate the development of a SO taxonomy of inconsistencies (Section 1.3).

Hence, the Annotated Shami Corpus (ASC) was developed along with the ANNOTATIO annotation platform which was used to annotate the corpus.

**Spontaneous Orthography Taxonomy** Once the corpus was annotated, the SO standardization source-target-pairs were used as a reference to manually construct a taxonomy of spelling inconsistencies, which in and by itself, is a useful reference to have in one's possession for comparative studies between MSA and Lebanese Arabic. But in fact, it lends itself to the even more useful task of tagging different source-target pairs using the leaf categories of the taxonomy and training a system to predict what kinds of inconsistencies reside in a pair. Furthermore, this taxonomy can be very helpful in making a morphologically-driven data augmentation process much clearer and more structured. Hence, each source-target pair was also annotated using the tags from this taxonomy.

**Training on ASC** As a final step, the gathered annotations were used to train multiple neural models, ranging for a morphological segmenter, to a joint morphological tagger and SO standardizer. Some experimentation was also carried out with the SO taxonomy tags. Unfortunately, the data augmentation step could not be completed due to time constraints, but should be experimented with since it has been rather overlooked in the past and especially in Arabic NLP.

**Author's Note** Standardization is a broad term. For example Zalmout and Habash (2020) proved that their system's standardization is so good, to the point where feeding in automatically standardized data to a downstream task is equivalent to feeding in manually standardized data. However, the general performance of DA systems, e.g., for the POS tagging task, is still considerably subpar to that of systems trained on MSA data (Darwish et al., 2018). Therefore, the goal of this thesis is not so much to create the best standardizer system, as it is to get a better understanding of the structure of DA and of what can be done to make it more responsive in downstream tasks by getting a better in-depth understanding of the standardization task in general. Our task is therefore as linguistic as it is computational and this is reflected all throughout the thesis. For instance, the taxonomy is a clear testament to that. The conjugation table for Lebanese Arabic verbs in Appendix D is another. Many research questions will be opened, and many will be expanded and left open, as what is sought here is an all-comprehensive account of the standardization task which in my opinion, is still badly understood.

### 1.5.3 Thesis Structure

#### No Man Left Behind

Below, are outlined the goals that were set at some point in the research phase, but were not included due to inconclusive results – to leave space for more notable ones – or to lack of time.

#### Linguistics

- Quantify inter-personal vs. intra-personal noise.
- Inter-dialectal vs. intra-dialectal differences.
- Describe the differences between the grammar of one of the dialects and MSA in detail.

#### Machine Learning

- Try to experiment with a joint multi-dialectal setting to see if the different varieties can learn from each other.
- Try data augmentation techniques on available datasets. These include:
    - leveraging character-level language model distributions
    - masking words at the input of word-level pre-trained language models to predict most likely words given a context and ranking outputs based on a string similarity metric
    - use the newly created highly structured corpus and especially the taxonomy to make augmentation decisions
- Prepend the standardizer model to systems found in previous studies and evaluate possible improvement in performance.
- Inspect character embeddings generated by training to see if the word-embeddings they form have a structure in a reduced 3-dimensional space, e.g., using t-SNE, which could reveal spontaneous orthography types or word patterns.
- Use probing explanation networks to inspect the standardization mechanism at a finer level (Conneau et al., 2018).

We must not forget that what drives all of this is our disposition towards being able to share resources between DA and MSA, and this thesis tries to describe all the obstacles which pose themselves in our way. Ultimately, improving the quality of DA resources in terms of standardization will:

- first and foremost, curb the amount of noise (SO) in incoming DA data from the real word before feeding the input to any system
- help improve the quality of pre-trained embeddings for DA
- be a great asset for linguists conducting comparative studies between DA variants or with other languages
- help improve language model scoring in the Speech Recognition and Optical Character Recognition (OCR) and pipelines

### Thesis Outline

This thesis report will be distributed as follows. Chapter 2 exposes the first phases of experimentation with orthography standardization models on the Beirut portion of the MADAR CODA corpus. Then, the ASC annotation process and the accompanying SO Taxonomy are thoroughly documented in Chapters 3 and 4. Finally, the experiments carried out on the ASC, as well as the conclusions drawn from those experiments are outlined in Chapter 5.

While reading the thesis, if some section feels somewhat difficult to follow, reading the next sections should most often elucidate the misunderstandings as it was written in a highly interconnected way.

# Chapter 2

# Preliminary Modeling

This chapter describes the work that was done before the creation of the Annotated Shami Corpus (Chapter 3). Several neural architectures were used, be it word-level or character-level, and only some of them yielded satisfactory results.

## 2.1 Sequence-to-Sequence Models

The powerful generalization properties of neural networks on sequences make them an attractive choice for the spelling correction task. For that reason, a sequence-to-sequence (seq2seq) Recurrent Neural Network (RNN) will be used in a Neural Machine Translation (NMT) setting, in which we will be “translating” from our raw inputs to the standard CODA* form. This applies to all models used in this section. What might differ between them is whether the unit of input to the seq2seq network is a sentence or a word.

Seq2seq RNN models Sutskever et al. (2014) have enjoyed great success in a variety of tasks involving sequence classification, and most notably, NMT. A model consists of two main neural networks: an *encoder* that processes the input and a *decoder* that generates the output. In an NMT setting, the encoder captures a meaningful and compact feature representation of the source sentence, while the decoder takes the latter as input and learns a data-driven mapping between the source and target side. Because such a model inherently fails to capture long-range dependencies between the sequence elements, and because of the single-direction processing (left to right) the architecture imposes, the RNN cells are augmented to Long Short-Term Memory (LSTM) cells and bidirectional processing is used respectively. Finally, as is now customary in all seq2seq models, an attention layer is added between the encoder and the decoder in order to allow the decoder to selectively focus on specific parts of the input at each decoding step based on previous decoding steps (auto-regressively), instead of relying on a condensed mixed signal from the last hidden layer of the encoder.

The models built for this thesis are divided into three categories as follows (see Section 2.3 for a detailed description):

- **Word-level Models**

  - WL: Standard word-level bidirectional LSTM-RNN NMT architecture.
  - WL-BERT: BERT2BERT architecture Rothe et al. (2020). Similar to the first architecture but using a pre-trained transformer architecture.

- **Character-level Models** (Figure 2.4)
  - CL-NO-CTX: Bidirectional LSTM-RNN NMT architecture but with sequences of characters and not sequences of words and where individual examples are sequences of word and not sentence characters. This is similar to Himoro et al. (2020) and Etoori et al. (2018).

- **Hybrid Models**
  - WL-HYBRID: Hybrid character-word-level-embedding bidirectional LSTM-RNN with a character decoder for each decoded word at the word level. This architecture is similar to Luong et al. (2016) (Figure 2.3).
  - CL-CTX: Same as CL but adding context (see Section 3). This is similar to Zalmout and Habash (2020) but using a BERT word-embedding context and without the morphological context (Figure 2.4).

The word-level models fared badly against their character-level counterparts for the simple reason that they would not train properly due to some defect in the implementation. Furthermore, working under the very naive assumption that words can be standardized without the use of any context (i.e., surrounding words), CL-NO-CTX proved to be less performant than the hybrid CL-CTX. Note that no model takes in full sentences as character representations. This helps keep the input length bounded as LSTM seq2seq models' performance is known to degrade with excessively large sequence lengths due to the vanishing gradient problem. Intuitively, this architectural set-up should be able to take in a noisy input, and accordingly map it into a de-noised form, potentially looking at the latter as a different language.

## 2.2 Data Preprocessing

### 2.2.1 Data and Preprocessing

Throughout the experiments, the Beirut portion of the MADAR CODA Corpus is used. It contains parallel sentences of noisy (with spontaneous orthography) source, and CODA*-standardized target sentences. After splitting the corpus into training and development data, we get the sizes shown in Table 2.1a. Some preprocessing specific to dealing with Arabic text was also applied, namely:

- stripping the data from diacritics since they can be inferred by the word's context and many writers omit them
- stripping punctuation as it is used inconsistently and the goal is not to standardize orthography
- removing examples with a source word length greater than 30 characters to prune out inanities from the data

| Train | Development |
|---|---|
| 9,000 | 1,050 |

(a) Dataset sizes in number of tokens (space delimited spans of characters).

| Source | Target |
|---|---|
| 4,100 | 3850 |

(b) Number of unique types in source and target sides of the data.

| Equal | Not Equal | | |
|---|---|---|---|
| 81% | 19% | | |
| | Split | Merge | No split/merge |
| | 13% | 2% | 85% |

(c) Type of source-target pairs (relationship direction is from source to target)

Table 2.1: Counts and statistics related to the Beirut portion of the MADAR CODA corpus.

### 2.2.2 Word-aligning the Data

First of all, each example is a *source-target pair*[1] (STP) which either contains at least one space character in the source side (**Split**), in the target side (**Merge**), or no space in either (**Equal** or **No split/merge**). See Figure 2.1 and Table 2.2 for an example of a **Split** STP. The distribution of STPs is shown in Table 2.1c. According to these statistics, 81% of the tokens in our corpus are spelled in an already standard way, i.e., according to CODA* guidelines. Pre-processing this corpus for the CL-NO-CTX and CL-CTX models was not an easy task due to the fact that it is aligned at the sentence level, while what is sought is word-aligned STPs. Remember that we are dealing with batches of words for these two models, and hence, mapping which tokens belong to each other on the source and target side is hindered by the presence split and merged tokens. The way in which the STPs are formed, i.e., the data is word-aligned, is described in detail in Appendix B.

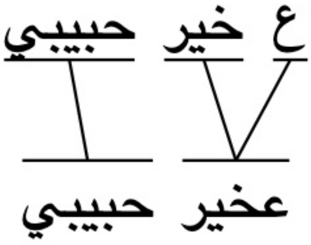


Figure 2.1: Correct alignment of the source ς␣xyr␣Hbyby (top) - target ςxyr␣Hbyby (bottom) pair.

| | Source | | | Target | |
|---|---|---|---|---|---|
| **Indexes** | 1 | 0 | 0 | 1 | 0 |
| **Aligned** | حبيبي<br>*Hbyby* | خير<br>*xyr* | ع<br>*ς* | حبيبي<br>*Hbyby* | عخير<br>*ςxyr* |
| **Example 1** | ع خير<br>*ς␣xyr* | | | عخير<br>*ςxyr* | |
| **Example 2** | حبيبي<br>*Hbyby* | | | حبيبي<br>*Hbyby* | |

Table 2.2: Generated training examples from the pair in Figure 2.1 which will be fed in to the character-level models.

---

[1] A source-target pair is a base word and its surrounding clitics. See Section 3.4.5) for a definition of those terms.

To avoid all of this, one could have built a character-level model which takes a fixed range of characters per input and which would have to intrinsically figure out which of those are related by a token relationship. Another way to solve this is to opt for models which take in batches of sentences as input (WL-HYBRID and WL-BERT) rather than words. However, those proved difficult to train as explained in Section 2.3.1. What made more sense in terms of intuitiveness was to feed in one word utterance at a time (CL-NO-CTX and CL-CTX models), i.e., a base word and its surrounding clitics. As clitics and/or affixes are split or merged approximately 15% of the time (Table 2.1c) when SO occurs, we have to group those together before feeding them the two CL models.

### 2.2.3 Merger Module

**Rationale** Now that we have the alignments, we can use the STPs to train our model. However, in the real world, we will neither have a target or an alignment, and consequently, we will not be able to know whether two source tokens actually belong the same token, as seen in the example shown in Figure 2.1. Therefore, a merger model (MERGER) is built to perform this action right before feeding in inputs to the standardizer (Figure 2.2). Note that MERGER and the CL models are independent and that it can be viewed as a pre-processing step before feeding the input into the standardization models. In other words, the loss does not back-propagate through it. The reason why there is a merger model but not a splitter is because the input to the standardizer model is a batch of tokens, and if two tokens are split, then we need to provide both of them to our standardizer model for it to merge them, i.e., remove the space character between them. If one token needs splitting as part of CODA* standardization, then the character-level standardizer can simply add a space character as part of the decoding stage.

**Implementation** Empirically speaking, the Beirut portion of the MADAR CODA Corpus does not contain any case which requires merging more than two tokens together. Theoretically speaking, one could think of a case where six tokens might have to be merged in the worst case before feeding them to the standardizer (i.e., five clitics and/or affixes and one base word). Seeing such a case in any corpus is highly unlikely, and in a realistic worst-case, one might see up to three or four (to be conservative) tokens that would require merging. For this model though, tokens were chosen to be processed one bi-gram at a time. As can be seen in Figure 2.2, a bi-gram (two tokens separated by a space character) is input to the merger model. The latter is a character bi-LSTM encoder with a fully connected layer at the output, which narrows down to two final outputs that are supposed to determine whether the tokens in the bi-gram should be merged or not. Additionally, to provide context for this operation, the encoder is initialized with the representation of the sentence which contains that bi-gram via the output [CLS] token of a multi-dialectal Arabic BERT model[2]. For training, details, refer to Appendix C.

---

[2] The BERT model used (MARBERT) was trained on 15.6B tokens of both unstandardized multi-dialectal Arabic data and MSA (Abdul-Mageed et al., 2020).

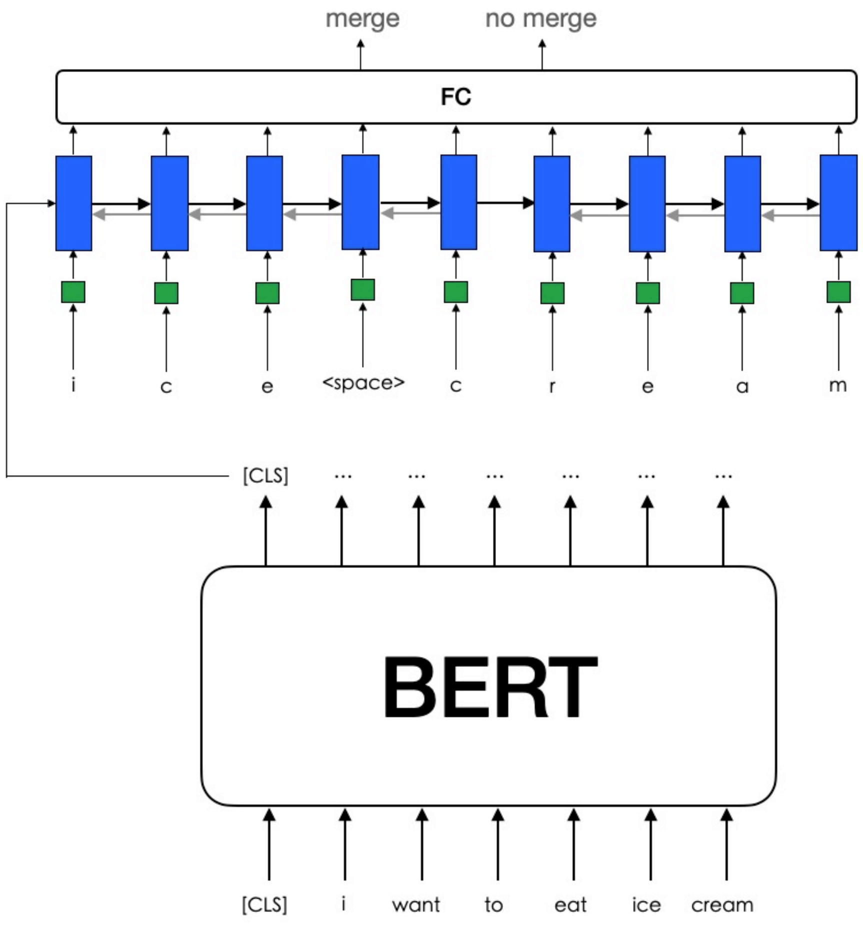


Figure 2.2: Diagram of the MERGER model. This model is trained independently from and then prepended to the CL-CTX model as shown in Figure 2.4.

**Results** Unsurprisingly, the accuracy of this module shot up to 100% as the only cases in this corpus which required merging were when the clitic و *w* 'and' was spontaneously detached from its corresponding base word. However, other corpora may display more varied cases. The Beirut portion of the MADAR Corpus is quite well-behaved. Now that this module is training, it can be added as a pre-processing step to the character-level models as shown in Figure 2.4.

## 2.3 Models

In this section, the models will be described. In all the model diagrams (Figures 2.3 and 2.4), the color blue refers to the encoders, light red to decoders, green to character embeddings, yellow to word embeddings, and brown to character-level decoders (when a word-level decoder is also present).

**Metric Used** To measure the performance of our systems, we will be using a special flavor of *F1 score*. Character accuracy is not very useful as it will always be high and make it more difficult to intuitively assess how the models are performing. However, using F1 score would require word-aligned data, and fortunately, this was already done as described in Appendix B. The way F1 score is calculated here is by calculating precision and recall in the following way:

$$precision = \frac{ne_{correct}}{ne_{total}} \tag{2.1}$$

$$recall = \frac{e_{correct} + ne_{total} - ne_{incorrect}}{e_{total} + ne_{total}} \tag{2.2}$$

where:

- $e_{total}$ = number of source tokens that should not have been altered (**Equal**)
- $ne_{total}$ = number of source tokens that should have been altered (**Not Equal**)
- $e_{correct}$ = those from $e_{total}$ that were left unaltered
- $ne_{correct}$ = those from $ne_{total}$ that were standardized correctly
- $ne_{incorrect}$ = those from $ne_{total}$ that were altered and standardized incorrectly

$$F1score = 2 \times \frac{recall \times precision}{recall + precision} \tag{2.3}$$

One could have used plain accuracy, or word-error rate (WER) to evaluate this task. However, these two would have also turned out high most of the time because about 81% of the input tokens in the dataset are already in standard form (see Table 2.1c. Therefore, this flavor of F1 score was designed in a way to best suit an intuitive interpretation for measure of performance in this specific task, i.e., SO standardization.

### 2.3.1 Word-level Models

**Rationale** The idea behind using word-level models potentially augmented with character decoders/encoders is rooted in the need to avoid the splitting and merging problem – which had to be solved using the MERGER model (Section 2.2.3). However, due to technical problems, these models did not train properly. The three word-level models are WL, WL-HYBRID, and WL-BERT. WL and WL-BERT are similar to WL-HYBRID which is shown in Figure 2.3. WL-HYBRID is inspired by Luong et al. (2016) which purpose is to deal with out-of-vocabulary (OOV) occurrences and include sub-word information to the NMT process. The three models have the same back-bone which is the word-level encoder, decoder, and attention modules. The only difference is that WL, the most basic model, is not augmented with a character decoder, and WL-BERT, the most sophisticated of the word-level models, has a word-level Transformer architecture (Vaswani et al., 2017) instead of plain bi-LSTM. It was built using the BERT2BERT architecture (Rothe et al., 2020) from the Hugging Face 🤗[3] Transformers library with MARBERT pre-trained weights (Abdul-Mageed et al., 2020).

---

[3] `https://huggingface.co/transformers/model_doc/encoderdecoder.html`

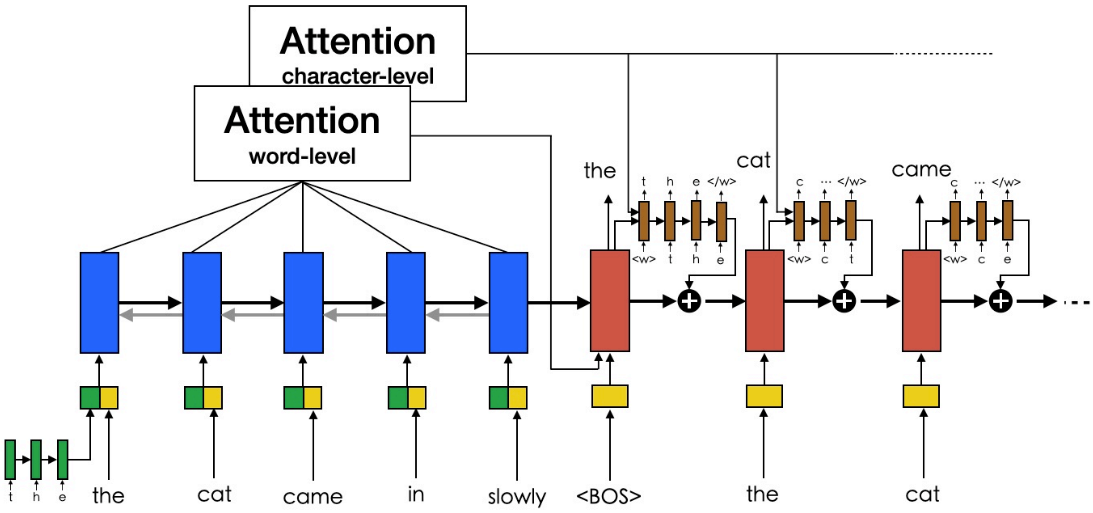


Figure 2.3: Diagram of the WL-HYBRID model. WL is similar it to but without the character decoders, and WL-BERT is the same except that the encoder decoder and attention modules are replaced with their BERT counterparts.

**Implementation** Intrinsically, this model does not require to deal with merged or split tokens as it takes in the whole sentence as input. But because it deals with words as units, it has a coarse granularity, which can nevertheless be solved by hybridizing the model through augmentation with character encoding/decoding. This is done, as shown in Figure 2.3 by concatenating character embeddings which are sourced from a character encoder (in green), itself, a bi-LSTM RNN, to word embeddings (in yellow) at the input of the word encoder (blue). For the hybrid model (WL-HYBRID), each decoder word output is fed as initialization to a character decoder (brown), the output of which is fed back into the word pipeline by adding it to the hidden state vector for that same word and using it as hidden input to the next time step. Note that as specified by Luong et al. (2016), word- (red) and character-level decoders have their own attention modules to avoid putting too much load on either and having a mismatch of gradient granularity. The character decoder and word decoder loss are added together to form the total loss. Training details are available in Appendix C.

**Results** For some reason, the loss related to the character decoders would not decrease, while the word loss seemed to be decreasing as expected. The cause of this is unknown, and further debugging is in order. However, due to the strict time constraints, this phase was aborted and was left for future inquiry. The metric used to evaluate performance in Table 2.3 (F1 score) is described in §2.3. The reason why no results are unavailable for WL-HYBRID is because the character decoders from which we are supposed to get the output were not training correctly (the loss was not decreasing). On another hand, as expected, WL has a substantially lower F1 score (47.0%) than all the character-level models. Performing regular NMT on a small and noisy dataset is bound to fail due to the high number of OOV tokens. Finally, at the word level, WL-BERT did not fare any better and even “standardized” some sentences in a peculiar way by

| Model | Recall | Precision | F1 score |
|---|---|---|---|
| WL | 61.1 | 38.2 | 47.0 |
| WL-HYBRID | - | - | - |
| WL-BERT | - | - | - |
| CL-NO-CTX | 98.2 | 82.5 | 89.7 |
| CL-CTX (BERT WLE concatenation) | 98.1 | 84.8 | 91.0 |
| CL-CTX (BERT CLE initialization) | **98.7** | **85.7** | **91.7** |

Table 2.3: Report of the F1 scores gathered across the multiple models trained.

translating source tokens to ones which are semantically related to the source tokens, as MARBERT incorporates a large amount of prior knowledge about DA and MSA.

### 2.3.2 Character-level Models

**Rationale** Character-level models, if not practically, intuitively make the most sense to use for the task of spelling standardization, as it is a task whose patterns inherently lie at a sub-word level. However, one caveat against this approach lies in the inability of sequence neural networks to handle very long sentences, ones that would be brought about by treating characters as units instead of words. Because of the vanishing gradient problem (Hochreiter, 1998), we use batches of tokens rather than sentences. Ngo et al. (2019) talks about how Transformers deal much better with long sequences than RNNs, and how character-based NMT is possible with those. This was not investigated for the moment due our satisfactory results for the best model (CL-CTX). However, it should be investigated in future work.

**Implementation** It was decided that the standardization model would take in sentences and standardize them one token at a time as seen in Figure 2.4. First, the input sentence is taken through the merger network (Section 2.2.3) one bi-gram at a time, in order to determine which tokens should be merged, i.e., have the space character between them deleted, before feeding them in to the standardization model one token at a time. If tokens need to be split, it is dealt with at the decoder side. The encoder-decoder setup works with batches of tokens and not sentences. The BERT model used for contextualization takes in the whole sentence, however BPE tokenization will often split tokens before inputting them to the model. Therefore, word embeddings belonging to the same raw token at the BERT last hidden layer output are summed together to "restore" the raw token. This can be seen in the example given in Figure 2.4. CL-CTX and CL-NO-CTX are the same, except that the BERT contextualization module is taken out for CL-NO-CTX. Training details are available in Appendix C.

**Results and Error Analysis** It does not come as a surprise that this is the best model since patterns are being extracted by the NN at the sub-word level, aided in CL-CTX's case by the contextualized word embedding of the token to

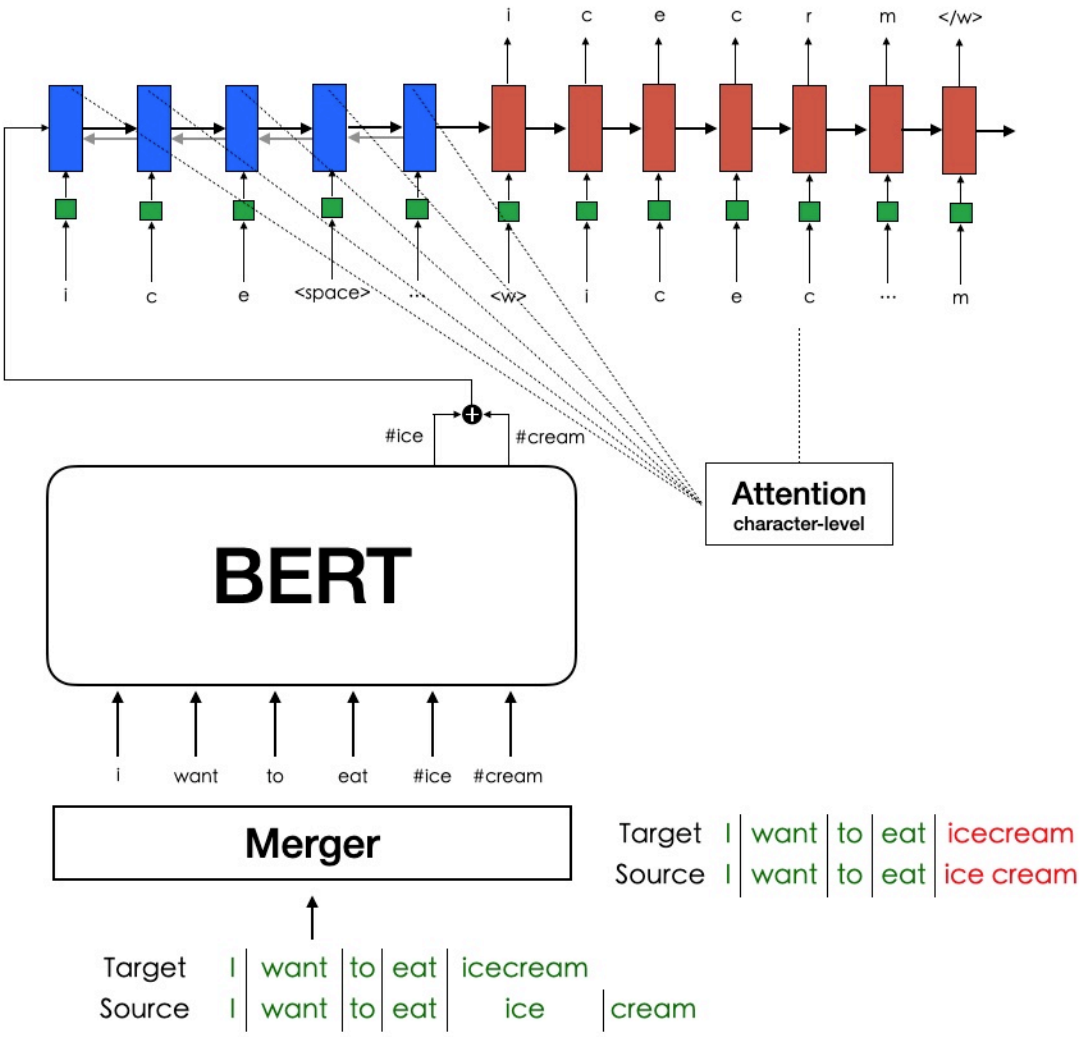

i
c
e
c
r
m
</w>
i
c
e
<space>
...
<w>
i
c
e
c
...
m
#ice
#cream
Attention
character-level
BERT
i
want
to
eat
#ice
#cream
Merger
Target I want to eat icecream
Source I want to eat ice cream
Target I want to eat icecream
Source I want to eat ice cream

Figure 2.4: Diagram of the CL-CTX model. The Merger module is trained before training the main standardizer module, and the standardization output does not back-propagate through it. It is described in Section 2.2.3.

standardize. Now, relativizing the results with results of other studies might be futile. All the other studies (Farra et al., 2014; Eskander et al., 2013; Zalmout and Habash, 2020) are either not very clear about the accuracy metric they are using, they are using different metrics, or they are using different datasets. In our case, recall is usually higher than precision because the majority of words are already in their standard form. The error in recall is due to those tokens that should not have been altered but were altered, and vice versa. As expected, adding context to the analysis increases both the recall (+0.6) and precision (+3.2) by a substantial amount (Table 2.3). CL-CTX solves many cases where context would have been necessary to predict correctly, mostly with function words. For example, predicting whether في *fy* should be standardized to فيه *fyh* depends on whether this token was meant as a copular verb, a preposition, or a modal verb. The last two require a standardization to فيه *fyh* while the the first one does not.

Now, most errors are cases where there was no occurrence of that token in the training set. However, in some cases, even zero-shot tokens (that were not seen in the training set) were successfully standardized, but the level of spontaneous orthography in them was quite low and could have been easily dealt with using a simple rule-base system, such as replacing the *hamzated* variations of the word-initial ا *A* (e.g., أ إ آ *ĀǍÂ*) to the non-hamzated version, or changing the word-final ه *h* to ة *ħ* when necessary. In contrast, there are some zero-shot examples, the standardization of which was not straightforward for the system seeing the small size of the dataset. Those were not standardized correctly, like for example standardizing تزكرة *tzkrħ* 'ID Card' to تذكرة *tðkrħ*. In Arabic, ذ *ð* /ð/ is simply the inter-dental version of the fricative ز *z* /z/, and since in the Lebanese dialect the former sound is always replaced by the latter, there is no way for the system to know this if it has never seen this token before. It may have been able to do it had the size of the dataset been larger, by inferring some kind of pattern between nouns that exhibit the same phenomenon.

With the current features of the dataset and its size, coverage is crucial in determining the accuracy of the system. As another experiment, the 27 examples of the Egyptian Arabic utterance ما بيقولهاش /mabiʔulhaːʃ/ 'he does not say it' (Table 1.1) were fed into the best model trained on the Cairo (Egypt) portion of the MADAR CODA corpus. The system was not able to standardize any of the different forms even though the accuracy achieved on the development set surpassed that of previous studies.

### 2.3.3 Conclusion

A qualitative comparison with other studies' results is not possible because access was restricted to the datasets they used, namely the ARZ Egyptian Dialect corpus from the LDC (Maamouri et al., 2012). However, to put things to into perspective quantitatively, the *word accuracy*[4], which is the same metric used in all of Farra et al. (2014), Eskander et al. (2013), and Zalmout and Habash (2020)[5], was calculated for the best system (CL-CTX), and was recorded at 95.2% for the Beirut portion. It is higher than the accuracies in all three papers. This does not necessarily mean that CL-CTX performs better than the other systems. Rather, this shows that the dataset we are dealing with is quite an "easy" dataset in terms of the forms of SO that it contains. Furthermore, taking into consideration the results on the ما بيقولهاش /mabiʔulhaːʃ/ 'he does not say it' utterance and the error analysis, this might be an indicator that the MADAR CODA corpus has low noise coverage and does not readily generalized to unseen cases, or that our system needs additional morphological context. This corroborates our propensity to develop an in-house dataset for two reasons:

1. Morphological analysis of the unstandardized text like in Zalmout and Habash (2020) might be very helpful for this task, especially in cases where

---

[4] A simple string matching metric which enforces a word alignment that pairs words in the reference to those of the output. It is calculated by dividing the number of correct output words by the number of words in the input (Farra et al., 2014).

[5] Even though Zalmout and Habash (2020) use full-fledged morphological analysis as context (alongside other features), they manage to report an accuracy which is lower than Eskander et al. (2013), which use a less sophisticated system. This warrants further investigation.

there are multiple morphemes per word such as with the Egyptian example.

2. The size of the dataset is too small, hence, data augmentation is in order. To do proper data augmentation, a morphologically segmented text would be very beneficial.

These two reasons are are the main driving forces which led to the creation of the Annotated Shami Corpus, described in Section 3.

# Chapter 3

# Lebanese Arabic Corpus Annotation

## 3.1 Rationale

**Low-resource Setting** While building corpora is a non-trivial and costly task, one which is becoming less and less popular due to state-of-the-art developments in distributed representation learning, it is sometimes essential when dealing with low-resource domains. One cannot disregard the (still ongoing) superiority of human supervision over the generalization properties of neural networks. After having built several models for the task of SO standardization, it has come as a realization that the task at hand is definitely non-trivial and requires specific types of resources, which currently are not readily available in a structured way that would allow for supervised training. Additionally, corpora like the MADAR CORDA Corpus (Eryani et al., 2020) which are annotated for spelling standardization seem to have low coverage of colloquial structures and expressions of a whole country as they only focus on one sub-dialect of that country. Consequently, it was decided that a corpus would be annotated for specific features which group together many features found in other similar datasets, although never together.

**Choosing the Right Features** Excluding sentence context from the input variables definitely is not in favor of correct SO standardization as seen in Section 2.3.1. Zalmout and Habash (2020) have even found that jointly training for lemmatization and morphological features alongside standardization leads to gains in accuracy. However, they relied on already available morphological analyzers, which in occurrence, work only for a specific dialect. Hence, the corpus described in this section will allow us to train a neural morphological analyzer for Lebanese Arabic, and at the same time, use those morphological analyses in the training for SO standardization. What is interesting about it is what makes it differ from the rest of similar datasets. First of all, morphological segmentation and tagging happens at the source (noisy) side of the corpus, in contrast with Khalifa, Habash, et al. (2018) which do it at the target (CODA*) side. Darwish et al. (2018) segment at the source side, but do not provide CODA* annotations.

And finally, corpora like Eryani et al. (2020) only annotate for CODA* without morphology. This is sub-optimal considering the absence of morphological analyzers for all the dialects[1]. On another hand, all of the mentioned corpora feature sentence-level alignment, and none of them make token-aligned sentences available. This is an issue because of split and merged tokens. Last but not least, segmenting at the source side instead of the target side makes morphologically-driven data augmentation possible, as explained in Section 3.2.

**Lebanese Arabic Dialects** When we say Lebanese Arabic, this encompasses a wide array of sub-dialects, despite the small size of the country (Section 1.5.1). However, when it comes to text data, the differences between those sub-varieties are heavily diluted due to the common practice of diacritic[2] omission. Alongside the fact that the sub-dialects do not substantially differ, or at least very mildly, in morpho-syntactic structure, this plays to our advantage. But nevertheless, the differences are non trivial, and the annotation is carried out in a fashion that captures all of those in a unified and consistent way.

Hence, this justifies creating a “corpus of Lebanese Arabic”, which will by extension, allow real-world systems to deal with more variation related to the same country, as it makes sense to evaluate systems’ performance at the country-level, by virtue of the geographic proximity and the resulting resemblance between the sub-varieties. It is worthy to note that it is not easy to find corpora with this kind of diversity as the latter are usually well controlled to suit one specific sub-region or one specific task, e.g., dialect identification.

## 3.2 Corpus Description

### 3.2.1 Annotated Shami Corpus

The corpus which was annotated is called the Shami Dialect Corpus (SDC) (Abu Kwaik et al., 2018). It is a Levantine Arabic corpus, divided into its four main varieties, namely, Jordanian, Lebanese, Palestinian, and Syrian. It is an unstructured corpus of tweets scraped from Twitter, is not confined to a specific domain, and features the kind of informal, highly dialect-centric use of language which is sought-after for standardization, without being too noisy in terms of random and intentional errors (see Section 1.3), but abundant in terms of non-random errors, i.e., Spontaneous Orthography. Both a pre-processed and raw version of the corpus are made available, and the raw version was used to annotate the present corpus. For the annotations in this thesis, only the Lebanese portion is used, and the resulting annotated corpus is called the Annotated Shami Corpus[3] (ASC). The ASC is annotated for:

- **Spontaneous Orthography Standardization**

---

[1] See Elnagar et al. (2021) for a recent survey on DA resources.

[2] Diacritics specify a large majority of vowels’ pronunciation in Arabic.

[3] The corpus will be made publicly available from the following GitHub repository: `https://github.com/christios/annotated-shami-corpus`

    - **Segmentation** STPs are aligned taking into account split and merged tokens[4].
    - **Standardization** Then, the source tokens are standardized into their CODA* form using the CODA* guidelines[5].
    - **Tagging** Using the taxonomy in Section 4, the individual STPs are tagged for the different cases in the taxonomy.

- **Morphological Segmentation and Tagging** This stage is independent from the last one and involves segmenting tokens at the clitic-base word boundary and tagging the different segments for:
    - *POS* (i.e., verb, noun, adjective, preposition, etc.)
    - *Obligatory Features* (i.e., aspect, gender, number, person, mood, etc.)
    - *Verb Forms* (i.e., templates of verb)
    - *Lemmatization* (i.e., dictionary form of a word)
    - *Dialectalness* (i.e., the extent to which a word belongs to a dialect)

## 3.3 Annotation Platform

### 3.3.1 Software Description

**Customized Platform** In order to ensure a smooth and efficient annotation process, a platform was built specifically for the types of annotations that we require. Other annotation platforms such as *brat rapid annotation tool*[6] and *WebAnno*[7] are great for annotating syntactic dependencies and named entities, but none of them allow for an expedient morphological annotation process, or the type of segmentation that is needed to map split/merged tokens to their raw version, or even for spelling standardization. The only platform which came closest to the needed requirements is *MADARi* (Obeid et al., 2018), however it was not publicly available online. Thus, a customized web application was developed for the purposes stated in Section 3.1.

**Speed, Accuracy, Consistency** ANNOTATIO[8] is a customized annotator which serves two main purposes. First and foremost, it allows annotators to efficiently

---

[4] In the Preliminary Modeling section (Section 2), it was found that a token- rather than sentence-based (inputs are tokens and not sentences) model provided the best results. However, one obstacle in the way of such models is the fact that some utterances are spontaneously split into several tokens at the time of writing, while they should be written as one token considering their standard form. Also, others are written as one token while they should be written as multiple tokens (more on this in Section 3.4.2). While Zalmout and Habash (2020) also use a token-based model, they do not specify how they deal with these cases, keeping in mind that about 15% of all spontaneities are split/merge errors (see Table 2.1c in Section 2.2).

[5] `https://camel-guidelines.readthedocs.io/en/latest/orthography/`

[6] `http://brat.nlplab.org/`

[7] `https://webanno.github.io/webanno/`

[8] The platform will be made publicly available from the following GitHub repository: `https://github.com/christios/annotatio`

annotate for spelling correction, and more specifically in our case, SO standardization. Second, it also allows annotators to efficiently segment sentences morphologically and tag the segments based on any tag-set. *Flask* was used for the back-end, *JavaScript-CSS* for the front-end, and a make-shift database was put together using the *git* framework by way of pushing and pulling, to synchronise all of the annotations. The skeleton of the web app was started by an experienced developer, while most features, testing, and debugging were developed in-house. Figure 3.1 shows ANNOTATIO's interface.

This interface was designed with many considerations in mind, the most prominent being:

- *Speed of Annotation*: When one needs to go through two rounds of segmentation (for SO standardization and morphological segmentation), finding the right way to do it is crucial. ANNOTATIO provides an easy way to do so which is explained in Section 3.3.2.
- *"Back-work" Compatibility*: Annotators can at any time go back to their previous annotations and make changes as needed.
- *Annotation Helpers*: Two search menus were implemented to help with the annotation process. One was linked to a database of examples of previous annotations from various annotated corpora, and the other was linked to the annotators' previous annotations.

All of the above features make for a platform which facilitates the creation of consistent, accurate, and expedient annotations.

### 3.3.2 Two Annotation Phases

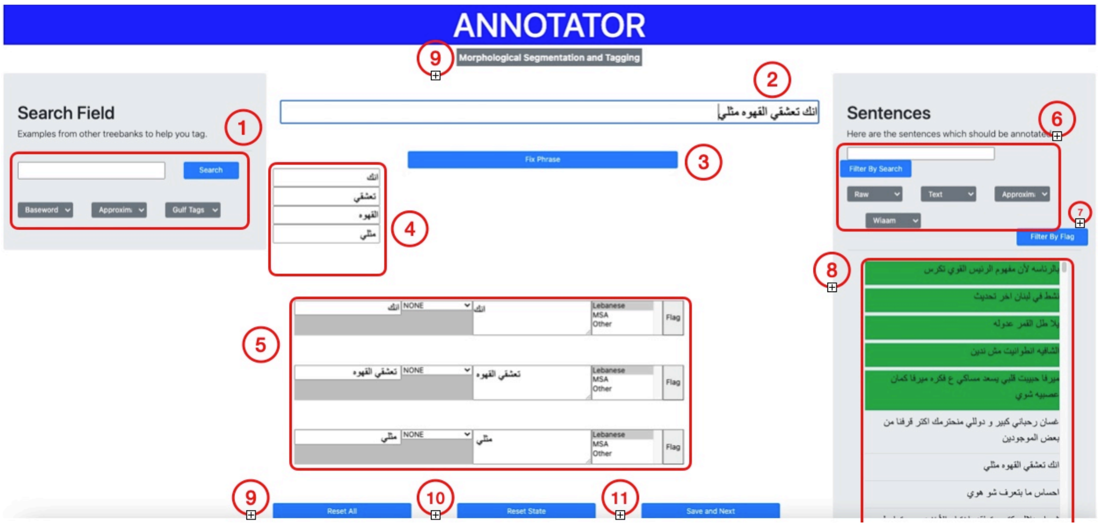
The annotation happens in two phases. The bold numbers inside parentheses refer to the corresponding section in Figure 3.1. The two phases are:

1. **Spontaneous Orthography Standardization** To get to this state, a sentence from the sentences on the right-hand-side of the screen (**8**) must be selected. Once in this state, the text on top of the middle sentence field where segmentation happens (**2**) will read "CODA Segmentation and SO Standardization" (**9**). In this state, you go through a first round of segmentation which will pave the way for spelling standardization[9]. In this state, every segmentation click generates one STP[10] (STP). To segment, one simply clicks after the last character of the source side of the pair one wants to create and an editable text field containing the target side of the pair appears below (**4**) so that one can standardize the spelling. Section 3.4.2 explains why we segment before standardizing the spelling. Clicking after the last character in this sentence triggers the second phase.

---

[9] Orthography Standardization happens according to the CODA* guidelines.

[10] The field being segmented (2) contains the source side, and the associated CODA* field (**4**) is the target side. STPs might contain spaces and Section 3.4.2 elaborates the reason why.



انك تعشقي القهوه مثلي

انك

تعشقي

القهوه

مثلي

انك | NONE | انك

تعشقي القهوه | NONE | تعشقي القهوه

مثلي | NONE | مثلي

بالرئاسه لأن مفهوم الرئيس القوي تكرس

نشط في لبنان اخر تحديث

يلا طل القمر عدوله

الشافيه انطوانيت مش ندين

ميرفا حبيبت قلبي يسعد مساكي ع فكره ميرفا كمان عصبيه شوي

غسان رحباني كبير و دوللي منحترمك اكتر قرفنا من بعض الموجودين

انك تعشقي القهوه مثلي

احساس ما بتعرف شو هوي

Figure 3.1: View of the annotation software[a].

---

[a] **1** is the Examples Menu. **2** is the Raw Text section where segmentation happens. **3** is the “Fix Sentence” button. **4** are the input fields resulting from CODA* segmentation, where spelling can be standardized. **5** are the input fields resulting from morphological segmentation where tagging happens. **6** is the “Search Previous Annotations” menu. **7** is the “Filter By Flag” button. **8** are the sentences available for annotation. **9** is the “Reset All” button. **10** is the “Reset State” button. **11** is the “Save and Next” button.

2. **Morphological Segmentation** This phase is automatically triggered when annotators click after the last character in the sentence in the middle field (**2**). (**9**) changes to “Morphological Segmentation and Tagging”. Once in this state, annotators must segment again, the same way they did for the first state, except that here, they would be segmenting for morphological segmentation, but for producing morphemes[11]. Refer to Section 3.4.5 for more details on how to segment morphologically. We call each morpheme a *segment*. Multiple segments constitute one STP.

### 3.3.3 Software Functionalities

Below are some of the most important features available in the interface. Again, the bold numbers inside parentheses refer to the corresponding section in Figure 3.1.

- **Save and Next Button (11)** At any time during either of the segmentation phases, one can click “Save and Next” to save the work for the sentence in the working space. As a consequence, the sentence is highlighted in green (**8**), and the next sentence is automatically selected.
- **Fix Sentence Button (3)** It is used to pre-process a sentence before even starting to segment it for CODA* standardization or morphological

---

[11] The following link specifies the guidelines followed for segmentation: `https://camel-guidelines.readthedocs.io/en/latest/morphology/`.

segmentation. Please read a more detailed explanation in Section 3.4.1 as this functionality is important.

- **Reset All and Reset State (9 & 10)** The first button is used to unregister an annotation, and the second one is used to zero-out the morphological segmentation stage, in case the annotator wants to retain work on orthography standardization.
- **Examples Menu (1)** A smart search bar and drop-down menus that provide a querying/filtering interface which enables annotators to efficiently search for annotation examples which were fetched from other datasets, namely, the MADAR CODA Corpus (Eryani et al., 2020), the Annotated Gumar Corpus (Khalifa, Habash, et al., 2018), and the NYUAD Arabic UD treebank (Taji et al., 2017b).
- **Search Previous Annotations Menu (6)** A smart search bar and filter drop-down menus which can be used to search for previous annotations done by the annotator, and by the other annotators.
- **Flag Button (5 & 7)**: A button available for each created segment in morphological segmentation which allows the annotator to flag it. It is used in cases when the annotator is unsure about an annotation or when a difficult annotation case is faced. It prevents the annotators from wasting too much time on specific things which would later be solved in common discussions.
- **Undo (4 & 5)**: Annotators can undo the most recent segmentations they did by pressing on a specific key combination.
- **Add/Remove STP/Segment (5)**: In the morphological segmentation state, annotators can add or remove segments or STPs between any two consecutive elements in a spreadsheet fashion. This feature greatly improved revision time and allowed annotators to make local changes in segmentations without having to reset whole sentences.

## 3.4 Annotation Pipeline

### 3.4.1 Pre-processing: SO vs. Typos

Since we are not doing spelling correction, we need to root out typographic mistakes (typos) from the raw corpus. To do that, annotators were provided with a “Fix Sentence” feature to pre-process the sentence before entering any of the two stages of annotation[12]. When using this option, one needs to exercise a fair amount of judgement, because it essentially lends itself to discerning typos from SO. The sole purpose of this feature is the following: we are standardizing “spontaneous orthography”, which is the result of DA not having any standard orthography; we are not correcting mistakes which are the result of the lack of attention of the writer, i.e., typos. Therefore, this feature is exclusively used to

---

[12] See “Fix Sentence” button in Section 3.3.3.

correct typos, and is done before actually starting to correct for SO following the CODA* guidelines. This is done to avoid the wasteful discarding of sentences with a perfectly sound syntactic structure due to some typo. Table 3.1 shows some examples. The following are the general guidelines that were given to the annotators:

- Minimal use should be made of this feature.
- Our goal is to capture the essence of the Lebanese Arabic variety, and not to describe typos.
- If a sentence is really unrealistic and incomprehensible, and fixing it would require drastically editing it, then discarding it is the best choice.
- If there are two parts (clauses) of a tweet that are completely independent, a comma should be added between them.
- If orthography can be directly inferred from MSA unambiguously and was written in a non-standard way, it is considered a typo if it is not at a clitic/affix boundary[13].

### 3.4.2 Orthography Segmentation

Segmenting before starting spelling standardization is an essential step which might not seem very obvious at first sight. However, we segment for one main reason, and this is to keep a mapping between the source- and target-side of each STP (raw to CODA*), since some spontaneous orthographies will involve merging and splitting STPs. In general, the spontaneous orthographies fall under one of the four main categories outlined below[14]:

1. One token which was written as multiple tokens, e.g., ح يكون *H␣ykwn* 'he will be' (spontaneous version) instead of حيكون *Hykwn* (**4**; standardized version). In this case we take the spontaneous version as one segment and remove the space character in the editable field to standardize.
2. Multiple tokens which were written as one token, e.g., مابدّي *mAbd∼y* 'I don't want' (spontaneous version) instead of ما بدّي *mA␣bd∼y* (standardized version). In this case we take the spontaneous version as one segment and add a space in the editable field to standardize.
3. Various types of spontaneous orthography that will not involve merging or splitting tokens, e.g., بدّو *bd∼w* 'he wants' should be standardized to بدّه *bd∼h*.

---

[13] The latter generally create much confusion for writers as they are the breeding ground for many phonological changes between the dialect and MSA.

[14] Transliteration in the Habash-Soudi-Buckwalter scheme (described in the Appendices) happens character-for-character and can be used to follow the logic in case one is unfamiliar with the Arabic script.

| **Example** | **Comment** |
|---|---|
| شيماء هلالي كتير متوفقه با ختيار الأغنيه<br>*šymA' hlAly ktyr mtwfqh bA xtyAr AlÂγnyh*<br>Shima' Hilali really did well with the choice of that song | If a space character occurs at a boundary other than a base word-clitic or affix-stem boundary, then it is usually considered a typo. Here for example, ب+اختيار *b+AxtyAr* 'in+choosing' had a space within the word اختيار 'choosing'. Had the space occurred between ب *b* 'in' and ا *A* (first letter of the word 'choosing', then it would have been considered SO. |
| نغس اه يا تايتا لطيفه حكمني مغس عا هل خبريه<br>*nγs Ah yA tAytA lTyfh Hkmny mγs ςA hl xbryh*<br>Stomach ache, oh Teta Latifeh, that's what got hold of me when I heard this. | In this example, مغص *mγS* 'stomach ache' is written as نغس *nγs* . In this STP, there are two things to note. First, there is a typo in substituting the م *m* /m/ with a ن *n* /n/ which is considered a typo because it would be hard for a writer to "spontaneously" mistake one for the other, and it is most probably due to the lack of attention of the writer. On another hand, the ص *S* is substituted with a س *s*. This can be considered as SO because the emphatic and pharyngealized voiceless alveolar fricative *S* /$s^ʕ$/ is often backed off to a plain *s* /s/ in DA. On the other side, تايتا *tAytA* 'grandma' is considered as SO because the word does not exist in MSA and there is no standard way to spell it. To standardize a word like this, all annotators should agree on a fixed form. |
| ما خلّونا نعمل تغير واصلاح<br>*mA xl∼wnA nςml tγyr wASlAH*<br>The did not let us push change and reform | If the orthography of a word can be unambiguously inferred from MSA, and the spelling inconsistency is not at an affix/clitic boundary, then it is considered a typo. |
| كتبنها للورقة<br>*ktbnhA llwrQħ*<br>We did write the paper | If the inconsistency lies at an affix/clitic boundary (morphological boundary), then it might be considered SO, depending on the severity of the inconsistency. Here, it can be considered as SO. |
| صعب انك تحس يجوز بتحس بس انت عديم احساس كيف بدك بتحس<br>It's difficult for you to feel it could be that you feel but you are without feeling how would you feel? | There is no explicit typo in this example, but it highlights a case where annotators were instructed to add a comma between two completely independent clauses. In this example, there are three independent clauses. |

Table 3.1: Table of examples showing differences between Spontaneous Orthography (orange) and typographic mistakes (red).

4. Some spontaneous orthographies might still involve various combinations of the above, e.g., مبدّو *mbd∼w* 'he doesn't want' should be standardized as ما بدّه *mA␣bd∼h*. Note that this example contains a merging spontaneity (second bullet) and a regular substitution spontaneity (third bullet).

### 3.4.3 Spontaneous Orthography Standardization

To perform CODA* standardization, the annotators are referred to the SO taxonomy presented in Section 4 and to the CODA* guidelines[15]. Many standardization cases can be inferred from the CODA* guidelines. However, the latter are very general as they aim to be dialect agnostic. This is both an advantage and a disadvantage, as it sponsors inter-dialectal research but at the same time fails to provide clear enough guidelines for all the ambiguous cases which arise for each dialect. Also, because we are trying to capture the whole Lebanese dialect and not just the Beiruti sub-dialect, some cases arise which are not specified by CODA*. Below are outlined the guidelines which CODA* does not explicitly attend to for the Lebanese dialect. Reading §3.4.5 first (definition of base word, affix, and clitic) is recommend for non-Arabic speakers before reading the below.

- **Intra-dialectal divergence** Because we are dealing with all the sub-dialects of Lebanese Arabic, some words are bound to be pronounced differently. The difference generally takes place at the vowel or semi-consonant[16] level. For example, at the vowel level, the utterance 'he gave' can either be pronounced as /ʕəte/ in which case a good spelling choice would be عطي *ςTy*, or /ʕata/ in which case the corresponding spelling would be عطى *ςTý*. At the consonant level, 'she put him to sleep' can either be pronounced as /najjameto/ in which case the obvious spelling choice would be نيّمته *ny∼mth*, or /nawwameto/ in which case the corresponding spelling would be نوّمته *nwwmth*. In these cases, we always choose to spell it the same way as its MSA cognate. Also, depending on the sub-dialect, some clitics that are otherwise silent, might actually be realized as a semi-consonant and a vowel such as in 'he can' فيه *fyh*. It might either be pronounced as /fiː/ or /fijo/ which might lead someone to write it as فيو *fyw*. We always standardize these cases to one form – the one that makes the most sense morphologically, i.e., فيه *fyh*. However, if the difference involves adding a hard consonant such as with فيني *fyny* /fiːne/ instead of فيي *fyy* /fiːjje/, the hard consonant is retained during standardization, and this results in two forms of the same clitic.

- **Letter switch** In the Lebanese dialects, some very particular commonly used day-to-day words are pronounced by switching the position of two sounds. For example, ملعقة *mlςQħ* /malʕaʔa/ 'spoon' and معلقة *mςlQħ* maʕlaʔa. The one which corresponds the most to the MSA form is usually chosen – in our case, the former.

---

[15] `https://camel-guidelines.readthedocs.io/en/latest/orthography/`

[16] The semi-consonants (or defective letters) are ا و ي *A w y* and can either function as vowels in the phonetic sense, i.e., /a/, /u/, and /i/, or consonants, i.e., /ʔ/, /w/, and /j/ respectively.

- **Cognate Backoff** Some very common words are sometimes heavily altered phonologically like امرأة *AmrÂħ* /ʔimraʔa/ 'woman' in MSA which is pronounced /mara/ in Lebanese Arabic. If such an occurrence is found in a sentence which would typically be considered as Lebanese Arabic morpho-syntactically, it is spelled the way it is pronounced, while retaining the MSA spelling as much as possible – in our case, مرا *mrA*. We call this *Cognate Backoff* because the writer most probably felt awkward writing it the way it was pronounced and wrote it in its MSA form.

- **Verb Conjugation** Appendix D lays out rules on how to conjugate spelling-ambiguous verbs based on aspect, voice, person, gender, number, and the presence or not of the *b-* prefix in the imperfective. The conjugation tables presented are an ad-hoc solution which served as a reference to conjugate Lebanese Arabic verbs. They were created by morpho-phonologically analyzing verbs that are common in Lebanese Arabic and MSA, according to the verb templates they adhere to, as MSA and Lebanese Arabic verbs follow the same set of templates[17]. In summary:

  - **Active Perfective** If a verb form contains a *connecting hamza* همزة وصل, then it retains it in its standardized form. If it does not – and this applies to all verb forms including the quadri-literal verb forms and their augmented versions – then it does not take this hamza since it is not pronounced in Lebanese Arabic.

  - **Passive Perfective** These verb are written as specified by the verb form (template) used. See Appendix D.1.1 for a more detailed explanation.

  - **Active Imperfective** We only specify rules for the third/first person, and masculine singular versions of the verbs with a *b-* imperfective prefix[18] because only those pose phonological problems[19]. In all verbs, the third person ي *y* prefix[20] should be added if the *b-* prefix is used in conjunction, even though it is not always pronounced as /j/[21]. For the singular with the *b-* prefix, the ا *A* imperfective prefix is never written after the former except in the case of hamza-beginning verbs (افعال مهموزة), in which case it gets assimilated by using the *madda* آ.

  - **Assimilated Imperfective Prefix** If the prefix to be used with the verb is the same as the first radical of the verb, then it sometimes gets assimilated into it. This sometimes leads to the omission of the prefix. It should hence be restored.

---

[17] Verbs and their spellings are especially cumbersome since their morphology is very productive. Adding to this the phonological changes of Lebanese Arabic, spelling certain verbs consistently while retaining the morpho-phonological structure of the dialectal form should be handled carefully.

[18] This prefix does not exist in MSA and poses some pronunciation/spelling problems here as it interacts with the verb's phonology.

[19] The pronunciation and spelling of the rest (feminine, plural, and second person) are inferable unambiguously from MSA grammar rules.

[20] Inflects based on gender, number, and person.

[21] see Appendix D.1.1

- **Command** All tri-literal non-augmented verbs exhibit middle vowel elongation in the command aspect (صيغة الأمر). The only exception is the doubled tri-literal form (افعال مضعّفة), in which no elongation takes place. This should be reflected in the spelling. Additionally, tri-literal verbs drop the connecting hamza as it is not pronounced in Lebanese Arabic. As an exception, augmented tri-literal verbs retain the connecting hamza.
- **Attached Enclitic** In the command verbs, the vowel elongation that happens with no attached clitic (previous bullet) is retracted except for the defective verbs (الافعال المعتلّة الآخر) which have a semi-consonant as the final radical.

Section 4 summarizes all the categories of spontaneous orthography as viewed from the CODA* guidelines' and our extended guidelines' perspective in the form of a detailed taxonomy.

### 3.4.4 Spontaneous Orthography Tagging

After generating the STPs by segmenting at STP boundaries, if the target side of the pair is altered by the annotators, it can be given a tag from the leaves of the taxonomy described in Chapter 4. Each STP can receive at least one tag, as many SO phenomena might be taking place within each one. The tagging decision process is described more depthfully in the taxonomy chapter.

### 3.4.5 Morphological Segmentation

In order to segment correctly, the annotators were asked to have good command of the concepts of *base word*, *clitic*, and *affix* (Habash, 2010). All the guidelines related to morphological segmentation are divided into the below paragraphs.

**Base Words, Affixes, and Clitics** In Lebanese Arabic, words are formed as shown in Table 3.2. Put simply, the difference between a clitic and an affix is that "affixes are needed to specify the obligatory features for [...] the POS of a base word which is [...] a stem and the minimal number of concatenative affixes". Hence, affixes are considered part of the base word. The obligatory features are specified in Table 3.3 for each POS. If a particle does not specify any of those features for a POS, then it is a clitic. Clitics are "syntactically independent but phonologically dependent morphemes that are attached to the word phonologically. Words can be base words or base words with added clitics." Both affixation and cliticization in Arabic are purely concatenative phenomena.

**Segmentation Boundaries** Morphological segmentation in our case happens at the boundary between clitics and base words, or between clitics and affixes. Segmentation does not happen between a stem and its corresponding affix(es). For example, in the utterance حيروح *HyrwH* 'he will go', the way to segment it is

| Enclitic(s) | Base word | | | Proclitic(s) |
|---|---|---|---|---|
| PART* | **Suffix(es)** | **Stem** | **Prefix(es)** | PART*<br>PREP |
| | PRON | VERB*<br>NOUN*<br>ADJ* | PRON | |

Table 3.2: Table depicting the structure of Lebanese Arabic nominals (nouns and adjectives) and verbs. Other POS generally consist of just a base word.

into two parts, namely, the future particle ح *H* ‘will’ and يروح ‘go (he)’. The prefix ي *y* (imperfective masculine particle) is attached to the stem روح ‘go’ (stem), and they both form the base word. The reason why ي *y* was not treated as a segment of its own is because it is an affix and not a clitic. And since affixes are part of the base word, we do not segment them out of it, while in return, clitics are segmented out.

**Contracted particles** As a matter of convenience, we consider that some particles come in an expanded and contracted (cliticized) form. For example, one can say that the future particle’s expanded form is رح *rH* /raħ/ and its contracted form is ح *H* /ħa/. The contracted form always attaches as a clitic to the word that comes after it.

**Bound Pronouns** One thing to note about bound pronouns[22] is that they can be mistaken for suffixes as they seem like they specify some obligatory features like gender, number, and person. However, they are not affixes, but clitics. Bound pronouns attach to other entities in three cases (bound pronoun underlined):

1. when attached to verb, they are used to mark:
    - its direct object, e.g., أكلها *ÂklhA* ‘he ate it’.
    - its subject when the verb is in the passive form[23], e.g., أكلت *Âklt* ‘it was eaten’.
2. when attached to a noun[24], they are used to mark possession to that noun, e.g., تفّاحتها *tf∼AHthA* ‘her apple’.
3. when attached to a preposition, e.g., فيها *fyhA* ‘in it’.

---

[22] Bound pronouns are morphemes that cannot occur independently of another morpheme. They are related to other words called their hosts

[23] In the active form, if an explicit unattached pronoun is not used, there is virtually no pronoun for the subject, so there is no clitic to segment out. This is because Arabic is a null-subject language.

[24] Bound pronouns are specific to the possessing entity when attached to a noun, and not to the noun itself.

| POS | Obligatory Features | Lemmatization |
|---|---|---|
| ADJ*, NOUN* | GENDER, NUMBER, STATE | M, S, I |
| ABBREV, ADV*, CONJ, DIGIT, FOREIGN, INTERJ PART*, PREP, PUNC | None | Does not change |
| PRON* | PERSON, GENDER, NUMBER | Does not change |
| VERB* | ASPECT, PERSON, GENDER, NUMBER, VOICE, MOOD | P, 3, M, S, A, I |

Table 3.3: Table showing the different POS[a] tags and their corresponding obligatory features[b]. Refer to Section 3.2 in the webpage referenced in Footnote 11 for what the values mean.

---

[a] A tag with a star next to it refers to any tag that starts with what is right behind the star. For example PART* might mean PART, PART__DET, ADJ__INTERROG, etc. There are a few exceptions to those which are: ADJ__COMP, VERB__PSEUDO, PRON__INTERROG, and PRON__EXCLAM. Their names are misleading but they do not take any features.

[b] The lemmatization column holds the value of the respective feature which should be used to lemmatize a word.

Bound pronouns related to subjects are considered as part of the base word and are not segmented out because they specify gender, number, and person.

**Exceptions** While we follow the above logic throughout all of the annotation process, there are some exceptions that apply:

- The ال *Al* ‘the’ determiner prefix which denotes state, while being a prefix (since it is necessary for denoting state), was segmented out of the base word.
- The ب *b* prefix marker (Naïm, 2016), is an auxiliary particle that can be attached to the imperfective aspect of every verb, and which inflects to م *m* in the plural. Although it goes hand in hand with the ي *y* imperfective suffix, it was segmented out of the base word as some syntactic structures make use of it, while others don’t.

### 3.4.6 Morphological Tagging

#### Morphology

**CAMEL POS Tag-set** For morphological segmentation and tagging, the CAMEL POS tag-set[25] was used to tag segments gathered as described in Section 3.4.5. It claims to have been designed to facilitate research on adaptation between MSA

---

[25] `https://camel-guidelines.readthedocs.io/en/latest/morphology/`

and DAs, to support backward compatibility with previously annotated resources, e.g., using the Universal Dependencies[26] or the ARZATB tag-set (Maamouri et al., 2012), and finally, that the inherent morphology it captures is more compatible with Arabic morpho-syntactic rules than other tag-sets.

The difficulty of this tagging task lies in that many word categories in Arabic can be confused for syntactic functions. For example, in Lebanese Arabic, (predicate) nouns often fulfill the function of a verb if translated to English. Furthermore, the existential particle فيه *fyh* ‘there is’ is actually a preposition attached to a pronominal enclitic[27]. Other similar examples include the frequently used modal-like auxiliaries بدّه *bd∼h* ‘he wants’ and لازم *lAzm* ‘should’ which are actually nouns. Many such cases made the annotation process quite bumpy. Care was taken to annotate each word on the basis of its closest MSA-grammar category and not its English translation, the way it is prescribed in the morphology section of the CODA* guidelines. On another hand, the tags which posed the most issues were VERB_PSEUDO, VERB_NOM[28].

All in all this tag-set lived up to its purpose and was used for its simplicity, inclusiveness, and to promote shared resources within the Arabic NLP community.

### Lemmatization

Lemmatization is “the process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word’s lemma, or dictionary form”[29]. For example, in English, the lemma of *dogs* is *dog*. English is a morphologically poor language so there will not be many words mapped to the same lemma. In Arabic, it is quite the opposite. For example, كانا، كانت، كانوا، يكون، يكونون، يكونوا، يكونان، كنّا، كنت ‘{I, you, he, she, we, they} {was, were}’, etc. all map to the same lemma which is كان. For each category of POS, Table 3.3 also specifies how to lemmatize a specific segment. For example, to lemmatize the segment مزارعون ‘farmers’, we take it to its masculine, singular form, as specified by the table. The reason why we are not considering state even though Table 3.3 dictates that it is an obligatory feature lies in the exception from Section 3.4.5. The ال *Al* article will always be segmented out, so there will be no need to worry about state for the purposes of lemmatization.

**Lemmatization Process** As specified by Table 3.3, to lemmatize a nominal or a verb (NOUN*, ADJ*[30], VERB*), one simply converts the token to its “dictionary form” by changing its features to what is prescribed. No lemmatization is required for POS which are neither nominals nor verbs. Finally, and most importantly,

---

[26] `https://universaldependencies.org/u/pos/index.html`

[27] In Arabic, more weight seems to be placed on syntactic rather than morphological processes to create meaning.

[28] Pseudo verbs and verb nominals are words that have the same syntactic behavior as verbs in that they take a subject and a predicate, or a sentential complement.

[29] Taken from `https://en.wikipedia.org/wiki/Lemmatisation`.

[30] Comparative and superlative adjectives are lemmatized by being taken to their positive form (base adjective).

lemmatization happens to the dialectal CODA* feature-stripped root and not to the MSA cognate root of a token.

### Dialectalness

This feature is expected to help the model deal with standardization cases where code-switching[31] is happening between Lebanese Arabic and MSA. There is a drop-down menu for each segment (**5**) with "Lebanese" (default), "MSA", and "Other" as values. This is for when a word has the structure of an MSA (فصحى) or other dialect word (e.g., Egyptian). For example, take the sentence:

اساسا لو تعرفوا انو هل عرسان رح يسعدوكن ما بتحكو هيك
'Had you known in the first place that these newlyweds would be helping you, you wouldn't be talking like this'

The utterance لو تعرفوا *lw tςrfwA* 'had you known' does not sound very Lebanese in any of the majority of the Lebanese dialects, as the way it would properly be said is لو بتعرفوا *lw btςrfwA* with the ب *b* prefix marker. And since it does not have an MSA structure (i.e., لو عرفتم *lw ςrftm*), then it is neither "Lebanese" or "MSA" and should be tagged as "Other". That is not to say that if a word is common between Lebanese and MSA that it should be annotated as MSA. Most words should have a value of Lebanese. But if a word really sounds "un-Lebanese", then the annotators were instructed to choose either "MSA" if the structure used is purely MSA and feels awkward in Lebanese (in this case, we say the sentence is code-switched), or "Other" if it comes from another dialect. If the sentence is fully written in MSA or in an other dialect, then the instructions were to discard it.

## 3.4.7 Corpus Information

### Logistics

Putting together this corpus was no easy task. The biggest challenge was building and maintaining the annotation software which was essential for consistent and rapid annotation. The second biggest challenge was to figure out how to deal with ambiguous annotation cases, training the annotators and keeping them connected. The corpus was annotated by five annotators[32]. The annotation process followed a divide-and-conquer strategy. First, orthography and morphology segmentation, and orthography standardization were completed together, and then POS and orthography tagging followed. The corpus in its final version consists of 9,600 tokens, as seen in Table 3.4a. After 5,000 tokens were annotated, the remaining 4,600 were annotated automatically by the system described in Section 5, and corrected by one of the annotators.

To assess the degree of consistency, an inter-annotator agreement score was to be calculated by infusing the tweets of each annotators with 50 common tweets.

---

[31] Code-switching is when two different languages are used within the same sentence. Here we consider Lebanese and MSA to be two different languages.

[32] The annotators were students in linguistics-related fields.

| Train | Development |
|---|---|
| 8,600 | 1,000 |

(a) Dataset sizes in number of tokens (space delimited spans of characters).

| | Source | Target | Source$_{\text{seg}}$ | Target$_{\text{seg}}$ |
|---|---|---|---|---|
| **ASC** | 4,356 | 4,344 | 3,293 | 3,205 |
| **MCC (Beirut)** | 4,100 | 3,850 | 2,987 | 2,836 |

(b) Number of unique token and segment types in source and target sides of the data.

| Equal | Not Equal | | |
|---|---|---|---|
| 74% | 26% | | |
| | **Split** | **Merge** | **No split/merge** |
| | 11% | 1% | 88% |

(c) Type of source-target pairs (relationship direction is from source to target)

Table 3.4: Counts and statistics related to the ASC (without diacritics).

However, due to the turbulence of the annotation process[33], the annotations had to be revised many times, often by different annotators. So, calculating this score would not be a very good measure of consistency.

On another hand, the sentences that were annotated were not of very high quality as many of them were parts of sentences which were taken out of context. Furthermore, many tokens were split most probably due to the preprocessing of the authors of the Shami Corpus, most often at places where there would have been a *shadda* diacritic in the original tweets. This corpus was chosen because it is one of the few publically available corpora which divide the data at the country level rather than at the region level. However, in future work, a cleaner corpus should be considered. Nevertheless, the results we get are satisfactory as seen in Section 5. In a future iteration of this annotation task, the plan is to work with a more wholesome corpus, and automatically annotate sentences for them to be simply revised by annotators.

### Corpus Statistics

**Word Counts** The ASC is slightly smaller in size when one compares the type counts with that of the Beirut portion of the MADAR CODA Corpus (MCCB) as seen in Table 2.1. The most striking result is that the decrease in unique types from source to target in the ASC (−0.3%) is much smaller than that of the MCCB (−6%), as seen in Table 3.4b. This might be due to multiple reasons, one of them being that the annotations of our corpus are much more noisy than that of the MCCB. Another would be that there are more morphological phenomena taking place in the ASC. The latter are very often concatenative in nature, hence, contributing to this low decrease in types. This is corroborated by the segment types gathered for the two corpora by morphologically segmenting them using the SEGMENT model[34] described in Section 5.2.1. The result is two-fold. First,

---

[33] This includes the very steep learning curve of figuring out annotation details, making sense of Lebanese Arabic grammar and structure, and software issues.

[34] Being 94% accurate on Lebanese Arabic data, it can give us a fairly good idea of what is happening morphologically on both sides.

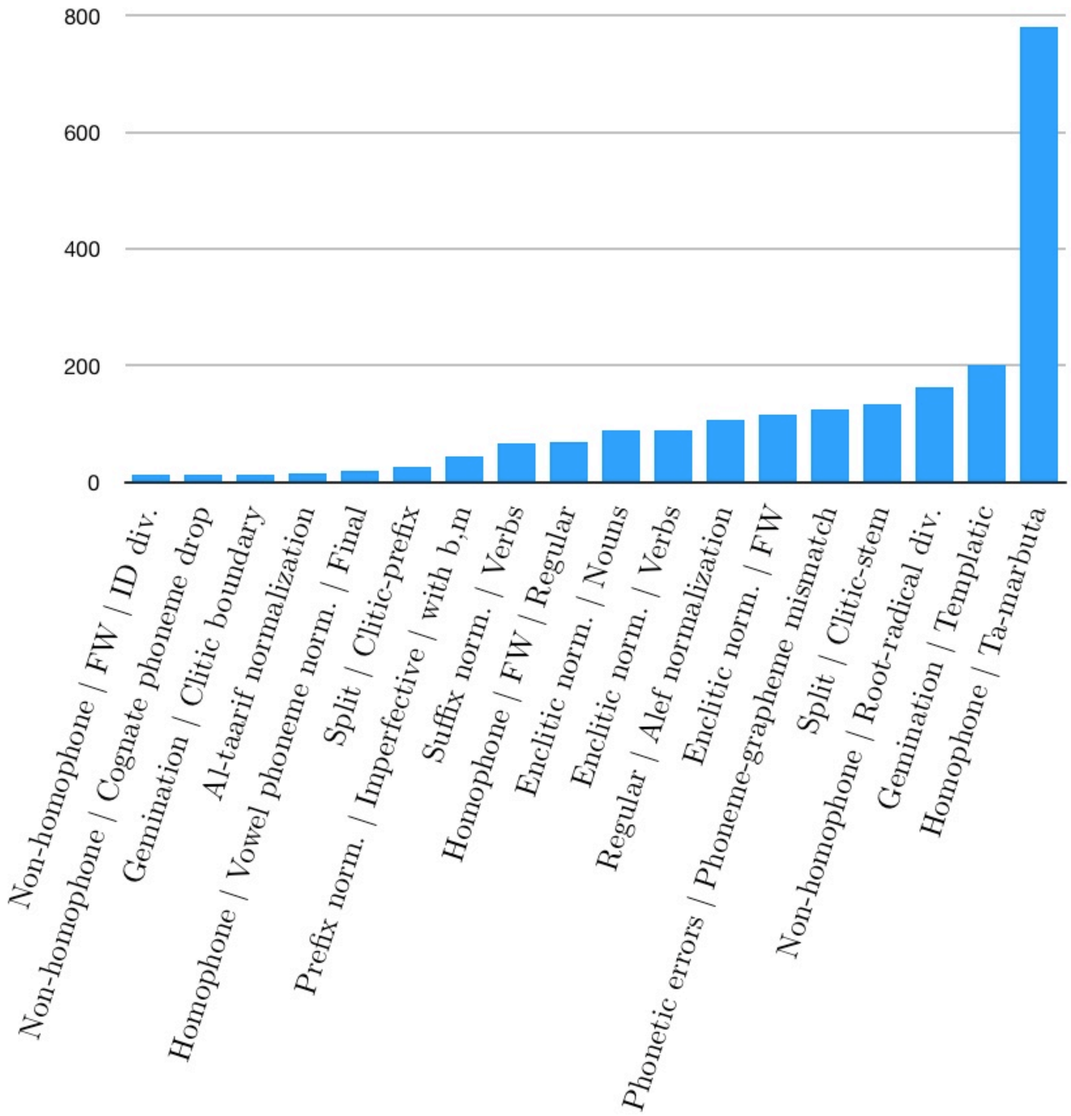


Figure 3.2: Histogram showing the number of occurrences per taxonomy category, only for categories which contain at least 10 occurrences.

there are 10% more segment types in the ASC than in the MCCB, which might be an indication that when concatenation is taken out of the picture, the ASC is richer in types. Second, the reduction in segment types for the MCCB ($-5\%$), is not as great as it was for the token types ($-6\%$). But more importantly, while the reduction in token types was near zero for the ASC, it is now much closer ($-3\%$) to that of the MCCB for segment types. While this should be investigated further, this might suggest that morphological phenomena in the ASC are more abundant. On another hand, the standardization annotation process followed for the ASC might have been the problem. This should be inspected by running more detailed tests, but is left for future research. Despite all of that, the ASC still contains about 10% more token/segment types than the MCCB. Finally, and most interestingly, the ASC also brought about a 7% increase in the number of source-target pairs in which source and target are not equal, as seen in Figure 3.4c. This means that it probably contains more types of non-standard orthographies.

**Lemmatization and Dialectalness** Lemmatization was done for individual segments, and was done based on the raw section of the data instead of the CODA* standardized section. Hence, the task of lemmatization latently had an orthography standardization component embedded in it. For this reason, the quality of the data could not be confirmed, and the lemmatization process should be revised in future iterations. Also, lemmatizing particles and pronouns was perplexing as some of them were very similar versions of each other with the same meaning, yet, are pronounced differently sometimes. Whether or not, and how these should be mapped to the same version was not clearly addressed and hence, this data was left out from the experiments in Section 5. Dialectalness was a very hard measure to asses because it is very subjective from person to person, in addition to being prone to heavy class imbalance. For this reason, it was also left out of the experiments.

**Taxonomy** Figure 3.2 shows the distribution of the different taxonomy tags[35]. The *Homophone ta-marbuta* category probably warrants being subdivided into more categories as it contains a disproportionate amount of occurrences which, after closer inspection, seem to be different in nature. All in all, there are 46 different leaves of the taxonomy tree, and hence, the same number of taxonomy tags. Those that contain less than 10 occurrences out of the 2,500 non-equal STPs were not included in Figure 3.2. A more detailed analysis of what is inside each category by way of sequence matching could very well elucidate the way for a morphologically-driven data augmentation process which is left for future research.

## 3.5 Conclusion

All in all, the aim of creating this corpus was to have a clear reference morpho-syntax of Lebanese Arabic. Since rules were never enforced for it, the amount of variation in people's writing and even speech is astounding. We postulate that this is one of the main contributors to the low performance of DA systems, which regardless of the low-resource setting, have to deal with these dialectal inconsistencies. The Annotated Shami Corpus (ASC) – a Lebanese Arabic corpus – presents unique features which are not available in other similar corpora. It is an all-inclusive corpus which allows training of a standardizer, segmenter and POS tagger at the source side. Creating this corpus was faced with many obstacles as the grammar of the dialect was learned along the way and as our knowledge of Arabic grammar grew with it. Both this corpus, the annotation platform used to create it, i.e., Annotatio, and the guidelines described serve as a blueprint for future efforts in this task, and the plan would be to build a much larger, and even more comprehensive corpus of Lebanese Arabic using the amassed knowledge and methods learning here. They will be publicly released and will be accessible via the links stated in the beginning of the chapter, hoping that they will serve as a motivation for further DA resource creation.

---

[35] See Section 4 for a detailed explanation of the different categories.

# Chapter 4

# Spontaneous Orthography Taxonomy

Himoro et al. (2020) put forward a taxonomy of spelling errors found in a variety of Philippine Creole Spanish called Zamboanga Chabacano. While an official orthography was established for it by the public authorities in 2012, it has not picked up traction and a barrier still stands between native speakers and a standard way of writing. As this resembles our case with SO, we base the backbone of our taxonomy on theirs. One of the main differences is that a standard orthography for DA was never officially established, hence, while many of the error categories in the Zamboanga Chabacano paper lie in the *Regular Errors* category[1], ours come across rather as *Arbitrary Errors* in the authors' definition of the term. However, a slightly different nomenclature is used here to reflect the fact that these are not errors, since there is no clear reference to fall back to, and use *Inconsistencies* instead of *Errors*. The top-most categories of inconsistencies were explained in Section 1.3. Here, we turn to the lower-hanging branches of the taxonomy.

## 4.1 Classes of Spontaneous Orthography

In this section, the overarching categories of SO are listed and defined. To the right of every category name lie its parent categories in decreasing depth. This section constitutes a high-level overview of the categories. One may find it more suitable to start reading Section 4.2 – which contains the leaves of the taxonomy – and then refer back to this section as reference for the overarching categories. It is left to the reader to proceed as they wish.

### 4.1.1 Non-random Errors

Also known as *Spontaneous Orthography*, this category is divided into two main categories which classify inconsistencies based on whether or not standardization can be performed using a deterministic rule-based process.

---

[1] Definitions may be found in Section 1.3.

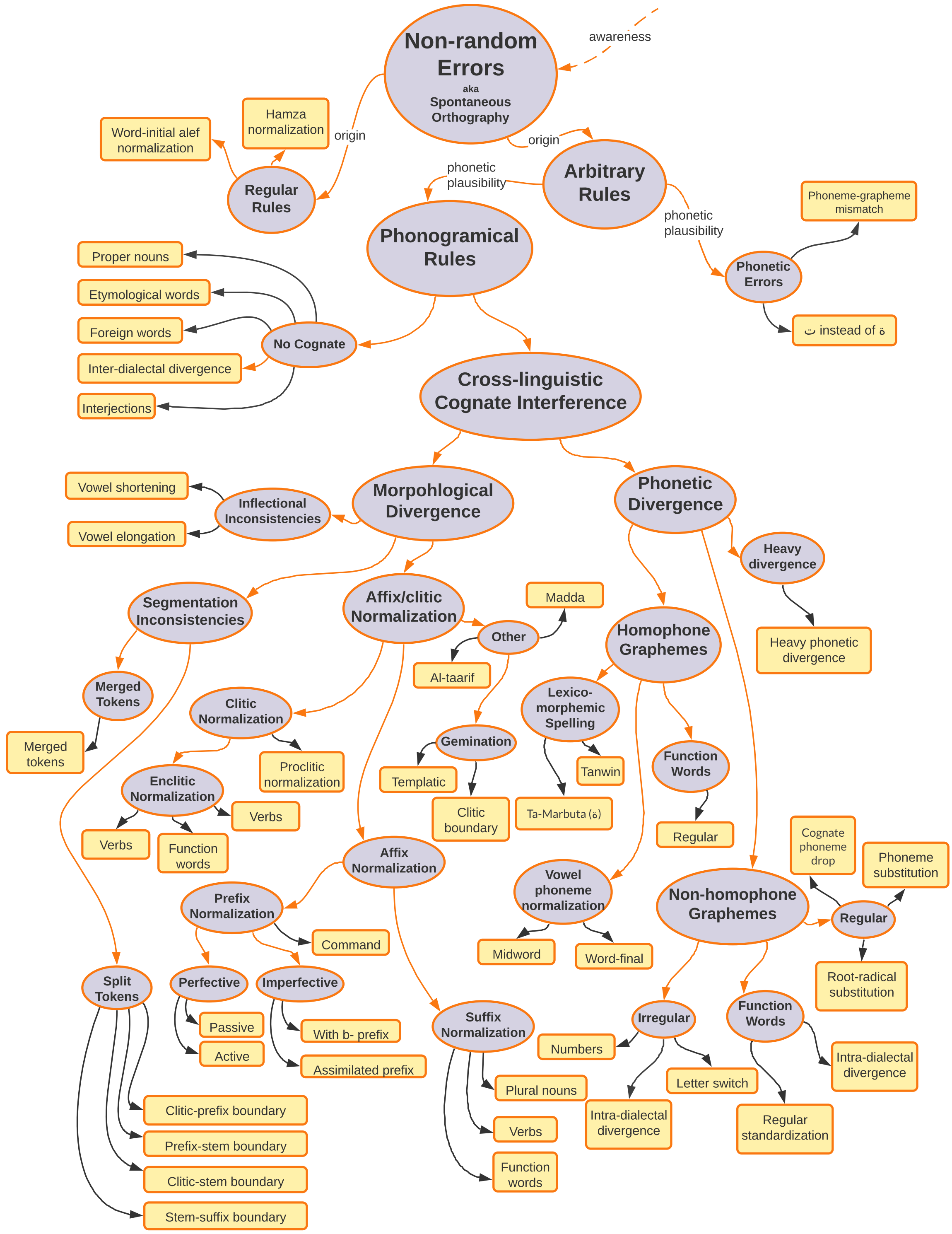


Figure 4.1: Spontaneous Orthography taxonomy manually created by inspecting contents of the ASC. Completes the diagram in Section 1.3 (Figure 1.1).

### 4.1.2 Regular Inconsistencies, Non-random Errors

Groups inconsistencies that could be standardized by using simple rules or rules which already exist from MSA and which could be directly applied here. This category does not carry a lot of weight in terms of how many other categories it groups.

### 4.1.3 Arbitrary Inconsistencies, Non-random Errors

Groups non-trivial inconsistencies which require more than simple rules to standardize. This branch splits into spelling inconsistencies that are non-standard yet realizable (reasonably inferable) phonetically, i.e., *Phonogramical Inconsistencies* and those that are not, i.e., *Phonetic Inconsistencies.* Most inconsistencies lie in this branch.

### 4.1.4 Phonetic Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups spellings, the intention of which is not realizable phonetically under any reasonable consideration. This branch does not carry a lot of weight since phonetically unrealizable spellings usually lie in the *Random Errors* branch[2], i.e., typographic mistakes.

### 4.1.5 Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups spellings, the intention of which is realizable phonetically but is not standard according to the CODA* guidelines. This category contains the bulk of the other categories. It is divided into those cases which exhibit cross-linguistic phonogramical interference with their MSA cognates and those that have no MSA cognate.

### 4.1.6 No Cognate Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups those cases which are hard to spell for a writer since they do not even have an MSA cognate to refer back to.

### 4.1.7 Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups those cases which are hard to spell for a writer since their phonetic realization in DA conflicts with their MSA cognate orthography. It branches out based on whether that interference is more phonologically or morphologically based. The phonological-morphological distinction here is not clear-cut and should be

---

[2] See Section 1.3

taken as lying on a spectrum[3]. All IPA transcriptions in parentheses pertain to the pronunciation of the MSA cognate and are not a transcriptions of the standardizations. They are meant to make the reader appreciate how the phonological difference between the MSA and DA word (IPA transcription on the left side) makes the spelling choice difficult.

### 4.1.8 Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups inconsistencies in which the deviation between the pronunciation and MSA spelling makes the spelling-pronunciation relationship unstable.

### 4.1.9 Morphological Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups Inconsistencies which are due to the interplay between phonology and morphology between DA and MSA.

### 4.1.10 Non-homophone Graphemes, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups cases where the grapheme used on the source side is not a homophone of the target-side grapheme.

### 4.1.11 Homophone Graphemes, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups cases where the grapheme used on the source side is a homophone of the target-side grapheme.

### 4.1.12 Heavy Divergence, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups cases in which the dialectal word has heavily diverged phonologically from its respective cognate.

---

[3] Many words in this category are standardized in way that often does not reflect their pronunciation, although doing so benefits consistency (lower token-to-type ratio) in standardization when considering multiple sub-dialects of Lebanese Arabic, or even DA in general.

### 4.1.13 Affix/clitic Normalization, Morphological Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups awkward spelling due to affixes, the pronunciation of which changes in DA.

### 4.1.14 Segmentation Inconsistencies, Morphological Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups inconsistencies presented in Section 3.4.2, i.e., in which a space character was used inconsistently. The space character is considered as a regular character and any segmentation that happens at a boundary which is not a stem-affix or clitic-affix-base word boundary is considered a *Random Error* as described in Table 3.1.

### 4.1.15 Inflectional Inconsistencies, Morphological Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups inconsistencies in which the inflection of a word into some specific grammatical category brings about awkward spellings due to the resulting mismatch between inflectional phonology and the spelling of the uninflected form.

## 4.2 Cases of Spontaneous Orthography

In this section, the lowest hanging branches of the taxonomy are described, and clear examples are provided for reference. The IPA transcriptions on the source side describes the intended pronunciation of the utterance. The words to the left and right of the arrows are the source and target respectively. Finally, note that any source-target pair can exhibit one or more of these cases.

### 4.2.1 Regular Inconsistencies, Non-random Errors

Groups inconsistencies that could be standardized by using simple rules.

**Word-initial *alef* normalization** There are multiple forms of the ا *A* letter[4] (*alef*) that contain the ء *'* (*hamza*) glottal stop. These should all be normalized to the *alef* without a ء *'* (*hamza*). The reason for this is that this letter is one of the very few other letters which have multiple "versions". Many writers omit the ء *'* (*hamza*) since it can be inferred from context. Example:

- أهلي *Âhly* 'my parents' → اهلي *Ahly*

---

[4] Forms of word-inital *alef*: آ إ أ *ĀǍÂ*. This letter is a placeholder for the ء *'* (*hamza*).

***Hamza* normalization** In the middle or end of the word, the ء ’ (*hamza*) letter has multiple forms[5] and which one to choose is governed by strict phonological rules. Example:

- مئلوف *mŷlwf* ‘familiar’ /maʔluːf/ →مألوف *mÂlwf* (/maʔluːf/)

### 4.2.2 Non-homophone Function Words, Non-homophone Graphemes, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

This groups all the different cases of closed-class function word orthographies whose spelling does not reflect the way the word is pronounced in Lebanese Arabic[6].

#### Regular Standardization

Groups cases in which there is no notable phonetic shift between the intended pronunciation and the spelling. It is usually limited to vowel or semi-consonant sounds[7]. Example:

- اليك *Alyk* /ʔılak/ ‘for you’ →الك *Alk* (/ʔilajka/)

#### Intra-dialectal Divergence

Groups cases in which the function word can be pronounced differently across the different Lebanese varieties, in which case, all variants are standardized to the same form. In this category, standardization only restricts itself to adding or removing semi-consonants and the ء ’ (*hamza*). If a change involves a valid dialectal hard consonant addition, then the latter is maintained. Examples:

- ياه *AyAh* /jjeː/ ‘(give) it (to)’ →اياه *Alk* (/ʔijjaːh/)
- هدا *hdA* /heda/ ‘this’ →هيدا *hydA* (/haːða/)
- والا *wAlA* /walla/ ‘or else’ →ولا *wlA* (/waʔılla/)
- هك *hk* /heːk/ ‘like this’ →هيك *hyk* (/haːkaða/)

### 4.2.3 Regular Non-homophone Graphemes, Non-homophone Graphemes, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups cases where the difference between the MSA and DA version usually involves either an addition, a dropping, or a substitution of a sound.

---

[5] ء ’ (*hamza*) forms in the middle or end of the word: ء أ ؤ ئ *ŷ ŵ Â* ’

[6] The standard orthography of many of these words had to be pre-decided with the annotators as there are often intra-dialectal differences (between the different Lebanese dialects).

[7] Also called defective letters, the semi-consonant are (ا و ي ى *ýy w A*). They are called this way because they can phonetically act as either a vowel or consonant sound.

### Phoneme Substitution

Groups inconsistencies in which DA words have an MSA cognate, one of the phonemes of which is substituted. They are standardized so that the grapheme reflects the pronunciation of the dialect. Examples:

- يد *yd* /ʔiːd/ ‘hand’ →ايد *Ayd* (/jad)

### Cognate Phoneme Drop

Groups inconsistencies in which DA words having an MSA cognate dropped one of the phonemes which constitute the latter. They are standardized to reflect the pronunciation of the dialect. The most frequent case of dropping is for the ء *’* (*hamza*). Examples:

- مساء *msA’* /masaː/ ‘evening’ →مسا *msA* (/masaːʔ/)
- اوفياء *AwfyA’* /ʔawfija/ ‘loyal (plural)’ →اوفيا *AwfyA* (/ʔawfijaːʔ/)
- اولاد *AwlAd* /wleːd/ ‘kids’ →ولاد *wlAd* (/ʔawlaːd/)

### Root Radical Substitution

This category groups the dialectal words which are very similar to their MSA cognates, but which have one of their root radicals[8] substituted in a one-to-one fashion according to a specific set of common sound changes[9]. They are standardized to their MSA cognate form, except for dialectal words substituting the mid-word ء *’* (*hamza*) glottal stop for another sound. Examples:

- متلك *mṯlk* /mətlak/ ‘like you’ →مثلك *mθ̱lk* (/mɪθlika/)
- نائب *nAŷ̱b* /neːjɪb/ ‘member of parliament’ →نايب *nAy̱b* (/naːʔɪb/)
- زبط *ẕbT* /zabat/ ‘it worked’ →ظبط *ẔabT* (/θˤbata/)
- أهوة *Â̱hwħ* /ʔahwe/ ‘coffee’ →قوهة *Q̱hwħ* (/qahwa/)
- سورة *s̱wrħ* /sawra/ ‘revolution’ →ثورة *θ̱wrħ* (/θawra/)
- دهب *ḏhb* /dɪhab/ ‘gold’ →ذهب *ð̱hb* (/ðahab/)

## 4.2.4 Irregular Non-homophone Graphemes, Non-homophone Graphemes, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups cases in which the difference between the MSA and DA versions can be influenced by factors which are slightly more complex than for regular non-homophone grapheme cases.

---

[8] Almost every Arabic word – except proper nouns and foreign words – has a *root* that consists of three or four letters. These are called *radicals*.

[9] Section 4.4.5 in `https://camel-guidelines.readthedocs.io/en/latest/orthography/`.

#### Numbers

Groups cases in which the pronunciation of numbers in DA diverges from the MSA spelling. Example:

- واحدة *wA̱Hdħ* /wɪħde/ ‘one (feminine)’ →وحدة *wHdħ* (/waːħidatun/)

#### Intra-dialectal Divergence

Groups cases in which the non function words can be pronounced differently across the different Lebanese varieties in which case all variants are standardized to the closest MSA form. All of these variants have exactly the same meaning. Examples:

- عطي *ςTy* /ʕəte̱/ ‘he gave’ →عطى *ςTý* (/ʕataː/)
- نيّمته *ny̱∼̱mth* /nay̱y̱ameto/ ‘she put him to slepp’ →نوّمته *nw̱∼̱mth*

#### Letter Switch

Groups cases in which some very particular commonly used day-to-day words are pronounced by switching the position of two sounds. They are standardized to the closest MSA form. Examples:

- معلقة *mς̱ḻQħ* /maʕ̱ḻaʔa/ ‘spoon’ →ملعقة *mḻς̱Qħ* /mɪḻʕ̱aqa/
- جوزته *j̱wẕth* /ʒ̱awẕto/ ‘his wife’ →زوجته *ẕwj̱th* /ẕawʒ̱atuhu/

### 4.2.5 Homophone Function Words, Homophone Graphemes, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

This groups all the different cases of closed-class function word orthographies whose spelling does reflect the word’s pronunciation in Lebanese Arabic but is not standard. It is one of the most productive categories.

#### Regular Standardization

Groups all cases in which the intended pronunciation can be easily realized by the spelling, but the latter is not standard according to CODA*. Examples:

- نحن *nHn* /nəħna/ ‘we’ →نحنا *nHnA* (/naħnu/)
- هوي *hwy* /huwwe/ ‘he’ →هو *hw* (/huwa/)
- عل *ςl* /ʕala/ ‘on’ →على *ςlý* (/ʕala/)

- {ال, لي, ل} {*Al*, *ly*, *l*} /lli/ ‘that[10]’ →اللي *ςlý* (/ʔallaði/)
- ايمت *Aymt* /ʔajmta/ ‘when’ →ايمتى *Aymtý* (/ʔajji mataː/)
- بقه *bQh* /ħbaʔa/ ‘(stop it) already!’ →بقا *bQA*

### 4.2.6 Homophone Vowel Phoneme Normalization, Homophone Graphemes, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

This groups all cases of words, the vowels’ spelling of which can result in a homophonic realization, but the choice of which is not standard. This phenomenon is due to the presence of defective letters in the cognate. We standardize to the form which resembles the cognate form the most.

#### Word-final

Groups cases in which the standardization should happen at the end of the word. Examples:

- موسيقا *mwsyQA* /musiːʔa/ ‘music’ →موسيقى *mwsyQý* (/musiːqa/)
- خره *xrh* /xara/ ‘shit’ →خرا *xrA* (/xaraːʔ/)

#### Mid-word

Groups cases in which the standardization should happen in the middle of the word. Example:

- مسيقى *msyQý* /musiːʔa/ ‘music’ →موسيقى *mwsyQý*

### 4.2.7 Lexico-morphemic Spelling, Homophone Graphemes, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

This category also groups spellings which can faithfully symbolize the respective pronunciation, but are not standard. Here, they are due to spellings in MSA which are lexical or morphemic, i.e., the spelling-pronunciation relationship is not one-to-one as it usually is.

#### Tanwin

In MSA, depending on whether a nominal is definite or indefinite and on the case grammatical feature, a noun can be realized using *tanwin*, i.e., adding the /n/

---

[10] Subordinating conjunction

sound at the end of the word, and this is realized through the use of a diacritic. This selectively transfers to some DA words. However, since diacritics are usually omitted, we get awkward spellings. Example:

- حدن *Hdn* /ħadan/ 'someone' → حدا *HdA* (احداً *AHdA* /ʔaħadan/)

#### t-marbuta

This groups all the different cases where spelling requires a *t-marbuta*[11] *ħ* /∅/ (silent *t*). Examples:

- دينيه *dynyh* /diːnijje/ 'religious[12]' → دينية *dynyħ* (/diːnijjaton/)
- محرزي *mHrzy* /məħərze/ 'worthwhile[13]' → محرزة *mHrzħ* (/muħrizaton/)
- راكبي *rAkby* /reːkbe/ 'is riding[14]' → راكبة *rAkbħ* (/raːkibaton/)
- ضهرا *DhrA* /dˤahra/ 'outing[15]' → ضهرة *Dhrħ* (/dˤahraton/)

### 4.2.8 Heavy Divergence, Homophone Graphemes, Phonetic Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

This section groups cases in which the dialectal word has heavily diverged phonologically from its respective cognate.

#### Cognate Backoff

This groups the inconsistencies in which the writer most probably backed off to writing a word in its cognate form even though writing it this way would make it heavily diverge from its actual dialectal pronunciation. Examples:

- امرأة *AmrÂħ* /mara/ 'woman' → مرا *mrA* (/ʔımraʔa/)

### 4.2.9 Clitic Normalization, Affix/clitic Normalization, Morphological Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups all cases which exhibit awkward spelling due to dialectal cliticization. Either the clitics are written phonetically but in a non-standard way which does not reflect correct morphology, or they are written in a way that corresponds to the correct way in MSA, but does not reflect true phonology in Lebanese Arabic. For the purposes of standardization, a balance is struck between both.

---

[11] This letter is used abundantly in MSA and is usually a feminine marker.
[12] Feminine predicate adjective
[13] Feminine adjective
[14] Feminine predicate noun
[15] Feminine noun

### Enclitic Normalization

Groups all cases in which enclitics are written in a non-standard way based on what the POS of the word they are attached to is.

**Verbs** Groups cases in which an enclitic is attached to a verb to mark a direct object. Examples:

- روحو *rwHw* /ruːħo/ ‘go (command plural)’ →روحوا *rwHwA* (/ruːħu/)
- بتعرفن *btςrfn* /btaʕrıfon/ ‘you know them’ →بتعرفهن *btςrfhn* (/taʕrifuːnahom/)
- خلّاكم *xll∼Akm* /xalleːkon/ ‘he let you’ →خلّاكن *xl∼Akn* (/xallaːkum/)
- بيضربا *byDrbA* /bjodroba/ ‘he hits her’ →بيضربها *byDrbhA* (/jadrubuhaː/)
- نسمعوا *nsmςwA* /nısmaʕo/ ‘(we) hear him’ →نسمعه *nsmςh* (/nasmaʕuhu/)

**Nouns** Groups cases in which an enclitic is attached to a noun to mark possession. Examples:

- معناتو *mςnAtw* /maʕneːto/ ‘meaning’ →معناته *mςnAth* (/maʕnaːhu/)

**Function Words** Groups cases in which an enclitic is attached to a function word. Examples:

- منو *mnw* /manno/ ‘he is not’ →منه *mnh* (no direct MSA cognate)
- فيون *fywn* /fijon/ ‘they can’ →فيهن *fyhn* (no direct MSA cognate)
- فيه *fyh* /fiː/ ‘there is’ →في *fy* (no direct MSA cognate)
- في *fy* /fiː/ ‘he can’ →فيه *fyh* (no direct MSA cognate)
- انو *Anw* /anno/ ‘that[16]’ →ان *An* /(ann)/

### Proclitic Normalization

Groups all cases in which proclitics are written in a non-standard way. There is no need to make a POS distinction here because proclitics are not an especially productive class of particles in DA in terms of grammatical features.

---

[16] Subordinating conjunction

**Regular** Groups all cases of regular proclitic standardization in which there is nothing special to note. Examples:

- حاالترك *HA Atrk* /ħaʔetrok/ ‘I will leave’ →حاترك *HAtrk* (/saʔatruku/)
- بيارضها *by ArdhA* /biʔardˤa/ ‘on the spot’ →بارضها *bArDhA* (/biʔardˤiha/)
- تايمشي *tAymšy* /tajemche/ ‘so that he can walk’ →تيمشي *tymšy* (/lijamʃija/)

### 4.2.10 Affix Normalization, Affix/clitic Normalization, Morphological Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups all cases which exhibit awkward spelling due to dialectal affixation.

#### Prefix Normalization

This category addresses prefixes. At the top level, it is divided based on the aspect of the verb, i.e., perfective, imperfective, and command. Due to to the different phonotactics of Lebanese Arabic, and the lack of any conjugation reference, some awkward spellings may arise. See Appendix D for a reference on how to deal with such cases.

**Active Perfective** If a verb form[17] contains a *connecting* ء ’ (*hamza*) همزة وصل, then it retains it in its standardized form. If it does not – and this applies to all verb forms including the quadri-literal verb forms and their augmented versions – then it does not take this ء ’ (*hamza*) since it is not pronounced in Lebanese Arabic. Examples:

- نفعل *nfςl* /nfaʕal/ ‘he got emotional’ →انفعل *Anfςl* (/ʔinfaʕala/)
- اتضرّر *AtDr∼r* /tdˤarrar/ ‘it was damaged’ →تضرّر *tDr∼r* (/tadˤarrara/)

**Passive Perfective** These verbs are written as specified by the verb form (template) used. See Appendix D.1.1 for a more detailed explanation. Examples:

- نكسر *nksr* /nkasar/ ‘it broke’ →انكسر *Anksr* (/ʔinkasara/)
- اتيأّس *AtyÂ∼s* /tjaʔʔas/ ‘he was made depressed’ →تيأّس *tyÂ∼s* (/tajaʔʔasa/)

---

[17] Almost every Arabic word – except proper nouns and foreign words – has a *root* that consists of three or four letters. Although there is a slight difference between the definition of *root* and *verb form*, they shall be used here interchangeably and unambiguously.

**Imperfective | With *b*- Prefix** Deals with cases where adding the *b*- prefix makes the spelling choice awkward. Examples:

- بقوم *bQwm* /biʔuːm/ ‘he gets up’ →بيقوم *byQwm* (/jaquːm/)
- بانضّف *bnAD~f* /bnaddˁef/ ‘I clean’ →بنضّف *bnD~f* (/ʔunaððifu/)

**Imperfective | Assimilated Prefix** If the prefix to be used with the verb is the same as the first radical of the verb, then it sometimes gets assimilated into it[18]. This sometimes leads to the omission of the prefix. It should be restored. Examples:

- نضّف *nD~f* /nnaddˁef/ ‘we clean’ →ننضّف *nnD~f* (/nunaððifu/)
- يخذلك *yxDlk* /jexedlak/ ‘he brings you’ →ياخذلك *yAxDlk* (/jaʔxuðu laka/)
- يأس *yÂs* /jiːʔas/ ‘he gets depressed’ →ييأس *yyÂs* (/jajʔasa/)
- تبّل *tb~l* /ttabbel/ ‘she seasons (sauce)’ →تتبّل *ttb~l* (/tutabbila/)
- اخذ *AxD* /eːxod/ ‘I take’ →آخذ *ĀxD* (/aːxuða/)

**Command** Deals with the processing of silent prefixes in the command aspect. Examples

- اشراب *AšrAb* /ʃraːb/ ‘drink (command)’ →شراب *šrAb* (/ʔɪʃrab/)
- نطلق *nTlQ* /ntˁəlɪʔ/ ‘go (command)’ →انطلق *AnTlQ* (/ʔɪntˁaɫɪq/)

**Suffix Normalization**

This category addresses suffixes. They can either be attached to verbs, nouns, or function words to realize some grammatical feature.

**Verbs** Groups cases in which a suffix is attached to a verb to mark number, person, or gender. This includes the null prefix (الضمير المستتر). Examples:

- غيّرناه *γy~rnAh* /ɣajjarna/ ‘we changed’ →غيّرنا *γy~rnA* (/ɣajjarna/)
- غيّرتو *γy~rtw* /ɣajjarto/ ‘you changed (plural)’ →غيّرتوا *γy~rtwA* (/ɣajjartum/)
- عطه *ςTh* /ʕətˁe/ ‘he gave’ →عطى *ςTý* (/ʕatˁa/)

---

[18] Such cases arise because Lebanese Arabic allows for two consecutive hard consonants not to be separated by a vowel, while MSA does not.

**Plural Nouns** Groups cases in which a suffix is attached to a noun to mark the plural. Many times, the ن *n* letter from the MSA plural suffix is dropped in Lebanese Arabic. This makes spelling awkward. Example:

- محاميه *mHAmyh* /muħaːmijje/ ‘lawyers’ →محاميي *mHAmyy* (/muħaːmiːn/)

**Function Words** Groups cases in which a suffix is attached to a function word. Example:

- انتوا *AntwA* /ɪnto/ ‘you (plural)’ →انتو *Antw* (/antum/)

### 4.2.11 Miscellaneous, Affix/clitic Normalization, Morphological Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups all cases which pertain to affix/clitic normalization but that do not fit in the other categories.

**Al-taarif Normalization** Due to the mixed pronunciation of the determiner article ال Al ‘the’ depending on context, it is sometimes ommited or misspelled. Examples:

- هلعلبة *hlςlbħ* /halʕɪlbe/ ‘this box’ →هالعلبة *hAlςlbħ* (/haːðihi lʕulbatu/)
- تعامل *tςAml* /ttaʕaːmol/ ‘the treatment’ →التعامل *AltςAml* (/ʔattaʕaːmul/)
- هشجرة *hšjrħ* /haʃʃaʒra/ ‘this tree’ →هالشجرة *hAlšjrħ* (/haːðihi ʃʃaʒaratu/)

#### Gemination

Groups all cases which have awkward spelling due to the interference of gemination in the phonology. Gemination usually stems either from clitic attachment or from templatic processes, i.e., the template of the word contains a *shadda* (doubling diacritic).

**Templatic** Groups cases in which gemination stems from the template of the word in question. In our case, we only focus on verbs. Example:

- سكر *skr* /sakkar/ ‘he closed’ →سكّر *sk~r* (/sakkara/)

**Clitic Boundary** Groups cases in which the spelling is made awkward due to a consonant doubling at a clitic-word boundary. Examples:

- سكّرتلله *sk~rtllh* /sakkartɪllo/ ‘I declined’ →سكّرتله *sk~rtlh* (/sakkartu lahu/)

- قلك *Qlk* /ʔallak/ ‘he told you’ →قللك *Qllk* (/qaːla laka)
- عنا *ςnA* /ʕınna/ ‘we have’ →عننا *ςnnA* (/ʕındanaː/)

### 4.2.12 Inflectional Inconsistencies, Morphological Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups inconsistencies where the inflection of a word into some specific grammatical category brings about awkward spellings due to the resulting mismatch between the inflected form’s phonology and the spelling of the uninflected form.

**Vowel Elongation** Groups cases in the command aspect of verbs that exhibit vowel elongation compared to their MSA form. This usually happens with verbs which contain semi-consonants. Examples:

- خذ *xD* /xoːd/ ‘take’ →خوذ *xwD* (/xuð/)
- نم *nm* /neːm/ ‘sleep’ →نام *nAm* (/nam/)
- عط *ςT* /ʕtˤiː/ ‘give’ →عطي *ςTy* (/ʔuʕtˤi/)

**Vowel Shortening** Groups cases in the command and imperfective aspect of verbs that exhibit vowel shortening. This usually happens with verbs which contain semi-consonants. This change is usually brought about by the attachment of one or more enclitics. Examples:

- خوذلك *xwDlk* /xıdlak/ ‘take yourself’ →خذلك *xDlk* (/xuð laka/)
- قولله *qwllh* /ʔillo/ ‘tell him’ →قلله *qllh* (/qul lahu/)

### 4.2.13 Phonetic Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups spellings, the intention of which is not realizable phonetically under any reasonable consideration. The only leaf of this branch is:

**ت *t* /t/ instead of ة *ħ* /∅,t/** At the end of a word, ة *ħ* is always used regardless of whether it is pronounced or not. Examples:

- عامت الشعب *ςAmt (Alšςb)* /ʕaːmmıt (ʃʃaʕb)/ ‘the common people’ →عامة الشعب *ςAmħ (Alšςb)*
- مدرست *mdrst* /madarase/ ‘school’ →مدرسة *mdrsħ*

**Phoneme-grapheme mismatch** Groups cases where there is dissonance between between the phoneme, grapheme, intended meaning, and pronunciation of a word. Examples:

- بهل الحكي *bhl AlHky* /bhal lħakı/ 'with this talk' →بهالحكي *bhAlHky* /bhal ħakı/
- مباره *mbArAh* /mubaːraː/ 'match (game)' →مباراة *mbArAħ* /mubaːraːt/

### 4.2.14 No Cognate, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

Groups those cases which are hard to spell for a writer since they do not even have an MSA cognate to refer back to.

**Proper nouns** Foreign proper nouns or those that are not typically Arab. Example:

- طوني *Twny* /tˤoːni/ 'Tony' →توني *twny*

**Etymological words** Words that are etymologically linked to other languages, e.g., French, Italian, Hebrew, Greek, Syriac, etc. and that inflect regularly like other dialectal words. Example:

- سفرن *sfrn* /sˤafran/ 'he went pale' →صفرن *Sfrn*

**Foreign words** Words that are explicitly foreign. Example:

- كاريزمة *kAryzmħ* /karizma/ 'charisma' →كاريزما *kAryzmA*

### 4.2.15 Segmentation Inconsistencies, Morphological Divergence, Cross-linguistic Cognate Interference, Phonogramical Inconsistencies, Arbitrary Inconsistencies, Non-random Errors

These are the inconsistencies first introduced in Section 3.4.2, and make up the cases in which a space character was used inconsistently. Refer to §3.4.5 for definitions of affix, clitic, and stem.

#### Split Tokens

Groups cases in which a space character was used when it should not have been. For the purposes of this section, we shall consider the definite article ال *Al* to be a prefix (as it rightly is) despite having resolved in previous sections to treat it as a clitic.

**Clitic-prefix boundary** Groups cases where the split happened at a clitic-prefix boundary. Examples:

- و تمشي *w tmšy* /wtımʃe/ ‘and she walked’ →وتمشي *wtmšy* (/watamʃi/)
- ل اللواء *l AllwA’* /lalliweːʔ/ ‘to the Major General’ →لاللواء *lAllwA’* (/lilliwaːʔ/)

**Prefix-stem boundary** Groups cases where the split happened at the prefix-stem boundary. Example:

- هل مذيعة *hl mðyςħ* /halmuziːʕa/ ‘this anchor (female)’ →هالمذيعة *hAlmðyςħ* (/haːðihi lmuðiːʕatu/)

**Clitic-stem boundary** Groups cases where the split happened at the clitic-stem boundary (no affix in between). Example:

- ع راسي *ς rAsy* /ʕaraːse/ ‘You got it!’ →عراسي *ςrAsy* (/ʕala raʔsiː/)
- ف ما *f mA* /famaː/ ‘so don’t’ →فما *fmA* (/famaː/)

**Stem-suffix boundary** Groups cases where the split happened at the stem-suffix boundary. Example:

- يطير لي *yTyr ly* /jtajjerle/ ‘he ruined (it) for me’ →يطيرلي *yTyrly* (/jutajjiru liː/)

### Merged Tokens

Groups cases in which a space character was omitted when it should have been used.

**Regular** No distinction is made between the merging cases as they usually happen at the boundary with proclitics in a way which is not very productive. Example:

- مابدّه *mAbd∼h* /mabaddo/ ‘he doesn’t want’ →ما بدّه *mA bd∼h* (/ma badduhu/)

## 4.3 Conclusion

In this section, a taxonomy of SO was put forward. It is meant as a starting point in terms of defining the boundaries of the SO standardization task. It is by no means definitive, and could very well be refined. However, due to the overall ambiguous nature of the task, the taxonomy was designed to be as flexible as possible in order to accommodate for the vast range of phenomena that take place in spontaneous writing. Furthermore, this taxonomy can be used to train a system to predict its different categories with a spelling-inconsistent token at the input, a process which might provide even more useful context for the standardization task. Since the different source-target pairs were tagged using this taxonomy as described in Section 3.4.4, a meaningful analysis of the distribution of the categories is in order, and is left for future research.

# Chapter 5

# Joint Learning Using Multiple Features

## 5.1 Multi-task Learning

In this section, various models are trained using the Annotated Shami Corpus[1] (ASC) in the goal of training a joint model for SO standardization using multi-task learning. This is done in a similar fashion to the models trained in Zalmout and Habash (2020) and Kondratyuk et al. (2018), and is especially similar to Zalmout and Habash (2020), where it has been proven that multi-task learning usually proffers improvement to some or all of the tasks involved. Here, we train for the following tasks:

- SO Standardization
- Morphological Tagging (POS and features)
- SO Tagging

The new task we introduce here is *SO Tagging*, which involves predicting what kind of taxonomy category[2] any source token falls under. Although the ASC is also tagged for lemmatization, this feature is left out for our purposes as the quality of those annotations could not be verified. All the models presented here will be accessible from `https://github.com/christios/orthonormalDA`.

## 5.2 Models

The general pipeline for Dialectal Arabic SO standardization requires first segmenting a sentence morphologically, tagging the individual segments for POS and grammatical features, and finally feeding in that context along with the BERT context to the standardizer model CL-CTX, found to be the best in Section 2. To

---

[1] See Section 3 for a thorough description of the ASC.

[2] See Section 4.

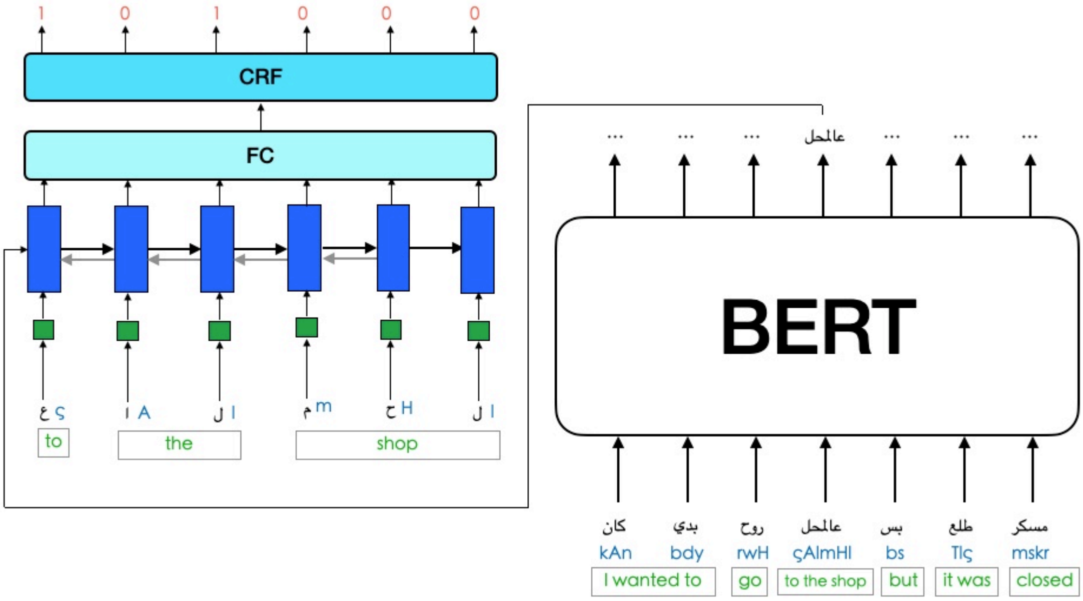


Figure 5.1: Diagram of the SEGMENT-CTX model.

do this, we build a morphological segmenter, and a morphological tagger. The SO Tagging task is independent and does not factor into the joint learning equation.

We follow the same minor pre-processing steps carried out in Section 2.2 but on the ASC rather than on the Beirut portion of MADAR CODA corpus, in order to be able to compare results. However, because the ASC already has its source-target pairs (STP) embedded[3], there is no need to perform alignment, which was previously used to account for split and merged tokens.

### 5.2.1 Morphological Segmentation

To perform morphological tagging on a morphologically rich and synthetic language, it can be very useful to segment words into their individual morphemes first. This is especially useful for Arabic and DA in particular, since many of the morphological phenomena taking place are concatenative, and hence, segmenting the data morphologically most of the time is akin to de-concatenating the tokens. The morphological segmentation model we use, which we call SEGMENT-CTX is very similar to the one used in Samih et al. (2017).

#### Data Processing

The data processing used for this task, i.e., how the segmentation boundaries were chosen, is described in detail in Section 3.4.5. Pre-processing is similar to that of CL-CTX.

---

[3] See Section 3.4.2 for a description of how STPs are formed.

| **Model** | **Recall** | **Precision** | **F1 score** |
|---|---|---|---|
| SEGMENT | 89.0 | 91.3 | 90.1 |
| SEGMENT-CTX (BERT CLE initialization) | **94.6** | **94.4** | **94.5** |

Table 5.1: Report of the F1 scores gathered from the SEGMENT and SEGMENT-CTX models training.

### Implementation

As seen in Figure 5.1, SEGMENT-CTX draws from the CL-CTX architecture[4], as it consists of the latter's encoder and BERT context at the core. For the sake of comparison, the same model is also tested but without BERT context. This model is called SEGMENT. Both take in batches of tokens made up of space-delimited sequences of characters, and their task is to decide whether each character in the sequence is at a segmentation boundary or not. This can be viewed as binary classification at each encoder output (token character). Hence, the outputs of the encoder are directly fed into a *Conditional Random Field* (CRF) layer, which acts as a unified *softmax* over all of the encoder outputs, instead of having a softmax layer independently decoding at each output. Using the CRF layer is driven by the assumption that a token may contain multiple segmentation boundaries, the positions of which is dependent on that of the others. Training details are recorded in Appendix C.

### Results and Discussion

The training results for this model are shown in Table 5.1. The metric used is the regular F1 score. The scores gathered are satisfactory when compared to state-of-the-art results seen in Samih et al. (2017). Directly comparing the current results with theirs is not useful since they work with the Levantine regional variety as a whole and not just Lebanese Arabic, and the size of their dataset is smaller than ours. Interestingly enough, adding BERT context to the model drastically increases its accuracy (+4.4%), reducing the system's errors by almost 50%. This seems to contradict the assumption of Samih et al. (2017) of context not being useful in 99% of cases. It would be interesting to run SEGMENT-CTX on their data and see whether this also holds, but is left for future research. In any case, this is the first time BERT context is added to the segmentation task for DA, as far as the literature review is concerned.

## 5.2.2 Morphological Tagging

With segmented data in one's possession, a morphological POS and grammatical feature tagging network can be built to tag those segments individually. Here, a multi-task model learns to tag each segment for the following features[5] (9 in total): POS, ASPECT, MOOD, GENDER, NUMBER, PERSON, STATE, VOICE, and VERB FORM. Only one segment feature from the ASC is discarded and that is

---

[4] See Section 2.3.2 for a description of the CL-CTX model.

[5] These are all described in Sections 3.4.5 and 3.4.6.

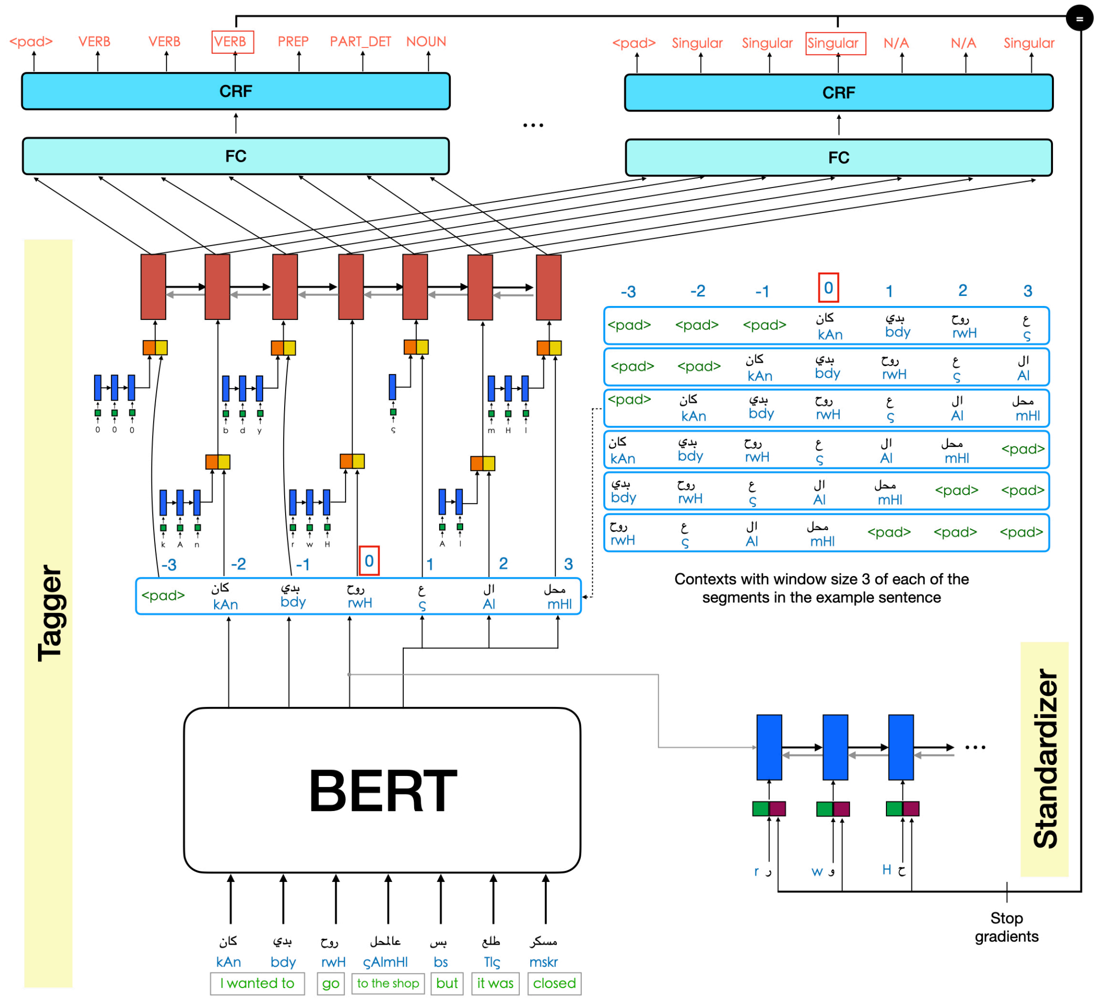

Tagger
Standardizer
BERT
CRF
FC
CRF
FC
<pad> VERB VERB VERB PREP PART_DET NOUN
<pad> Singular Singular Singular N/A N/A Singular
-3 -2 -1 0 1 2 3
<pad> كان kAn بدي bdy روح rwH ع ς ال Al محل mHl
Contexts with window size 3 of each of the segments in the example sentence
Stop gradients
كان kAn I wanted to
بدي bdy
روح rwH go
عالمحل ςAlmHl to the shop
بس bs but
طلع Tlς it was
مسكر mskr closed

Figure 5.2: Diagram of the JOINT model. It is made up of the TAGGER model, and the standardizer model (CL-CTX) on the bottom right.

*dialectalness*. Due to heavy class imbalance, this feature was not able to train properly.

### Data Processing

To process the data, the same approach as in Darwish et al. (2018) is followed. As seen in the center-right of Figure 5.2, a context window of a certain size is chosen, and each segment in the corpus gets represented by it and its context. If the window is of size $k$, then the context vector consists of the representation of the segment in question (index 0), and the representations of the preceding, and subsequent $k$ segments, so that the total representation of the segment is always of size $2k + 1$. For simplicity, *context* will refer to the segment and its context from now on. The model takes in batches of these contexts as input.

### Implementation

The model described here is called TAGGER. Once the context is fed in, each segment is passed through a bi-LSTM character encoder (blue) as seen in Figure

| **Feature** | POS | ASPECT | MOOD | STATE | GENDER | PERSON | NUMBER | VOICE | VERB FORM |
|---|---|---|---|---|---|---|---|---|---|
| **Recall** | – | 85.0 | 85.6 | 90.7 | 96.6 | 87.4 | 96.6 | 85.6 | 85.0 |
| **Precision** | – | 75.6 | 63.9 | 78.1 | 80.9 | 75.4 | 87.3 | 81.1 | 61.7 |
| **F1** | – | 80.0 | 73.2 | 83.9 | 88.1 | 80.9 | 91.7 | 83.3 | 71.5 |
| **Accuracy** | 85.8 | – | – | – | – | – | – | – | – |

Table 5.2: Report of the F1 scores and accuracy gathered from the TAGGER model training for the different features.

5.2. This encoder is different from the one used for standardization as it takes in morphological segments and not the source side of an STP. The final hidden layer of this encoder is used as a high-dimensional representation for that segment (orange). In addition, the BERT embedding (yellow) of the token which this segment belongs to is concatenated to the encoder character-level representation, forming a hybrid representation for that segment. This is done for each segment in the context. The latter is subsequently fed into the main bidirectional encoder (red) which is the backbone of the tagging architecture. The encoder outputs are then sent to 9 independent CRF layers, one for each feature, i.e., POS, ASPECT, etc. Finally, the loss is calculated for the whole context as the average of the individual loss for each of the features, while the accuracy and F1 score are calculated just for the segment with index 0 in the context. We also train TAGGER-POS, which is the same as TAGGER, but only for POS without the other features. See Appendix C for the full training details.

### Results and Discussion

The POS accuracy obtained is consistent with, yet not as high as the state-of-the-art for Levantine Arabic found in Darwish et al. (2018). Before analyzing the reason why, we should note that comparing the accuracy for TAGGER (85.8%) with that of TAGGER-POS (84.3%) shows that joint learning with the other features fittingly boosts the POS tagging performance model seems to have learned features which it wouldn't have been able to without the other grammatical features (see Table 5.2). The metric used for the grammatical features is the multi-class F1 score as there are more than two labels for each feature. Some POS require grammatical features that others don't, so the network is also inherently tasked with recalling which POS require which feature. This is captured by recall.

Now, if we turn to the sub-optimal performance of TAGGER compared to the state-of-the-art (87.9%), the most obvious reason would be that the training did not include $k$-fold cross validation or model ensembling, and since the difference is not that great, this cannot be determined. Second, the tagging inconsistencies in the ASC could be the culprit, as the latter could not be carefully pruned due to time constraints. A third reason could be the loss that is being taken over all segments in the context. Maybe it should only be taken for the main segment in the context. This question is asked because from the results of SEGMENT-CTX, the BERT model seems to have considerably boosted performance, and we expected a similar rise here. This could not be tested also due to time constraints.

| Model | SO Standardization | | | Tagging | | | |
|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | POS | Grammatical Features | | |
| | | | | Accuracy | Precision | Recall | F1 |
| CL-CTX | 78.9 | 97.5 | 87.2 | – | – | – | – |
| JOINT | 81.9 | 97.4 | 89.0 | 85.5 | 76.6 | 90.1 | 82.7 |

Table 5.3: Report of the F1 scores and accuracy gathered from the TAGGER model training for the different features.

### 5.2.3 Joint Learning

Now, the multi-task learning setting from the last section is extended to the SO standardization task. This was already carried out by Zalmout and Habash (2020), and the performance for that task was proven to increase with the benefit of joint learning.

#### Implementation

As seen in Figure 5.2, the JOINT model joins the TAGGER architecture with the CL-CTX standardization model. The link that is made between both consists of a fully connected layer which takes in the *argmax* of the *softmax* output for the main segment in each context concatenated across the features, and feeds them as context (purple) to the standardizer model by concatenating them to the character-level embeddings (green) of the encoder. One thing to note, is that similarly to Zalmout and Habash (2020) and Kondratyuk et al. (2018), the input to this fully connected layer has its gradients deactivated, hence, preventing the gradients of SO standardization from back-propagating through the TAGGER in the computation graph. The reason for this is that multi-task learning of tasks with different granularities, i.e., tagging which predicts tags, and SO standardization which predicts characters, can cause instability in the network's gradients. However, the setting is still joint learning, since both models still share the same loss. As with the TAGGER model, the loss is the average of the individual tasks' losses.

#### Results and Discussion

Based on Zalmout and Habash (2020), the results here are quite expected. From the results in Table 5.3, we can see that multi-task learning increases the standardization task's performance by reducing the system's standardization errors by 14%. This is expected because the standardization is as much morphological in nature as it is phonetic, so providing morphological context should naturally increase performance. Note that the metrics used for SO standardization here are the ones described in Section 2.3. It is also appropriate to point out that the performance of CL-CTX on the ASC (89%) is not as good as it is on the Beirut portion of the MADAR CODA corpus (91.7%). Two reasons can be thought of, the first being that the ASC contains standardization inconsistencies as it was not thoroughly pruned. The second reason postulates that the ASC is in fact harder to train on because it incorporates more types of SO. This should be further investigated by running this model on the MADAR CODA corpus.

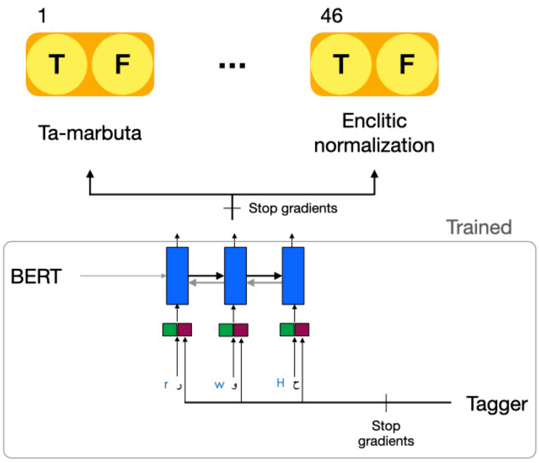


Figure 5.3: Diagram of the TAXONOMY model. It makes use of a already trained JOINT model without including it into its loss.

### 5.2.4 SO Tagging

Finally, we turn our attention to the new task of SO Tagging. As the ASC was tagged with the leaf categories of the taxonomy presented in Section 4, its aim is to infer the category of SO under which a particular STP fits. This model is called TAXONOMY.

#### Implementation

To realize the stated goal of this task, there are many ways to proceed. One should keep in mind that each STP can be tagged with at least one category. Inspecting the corpus shows that no token was tagged with more than four categories. Now, intuitively speaking, the categories can be viewed as independent from each other, in the sense that the presence of one is independent from that of another. While this statement should be validated by running tests on the data, we take it to be true for the purposes of SO Tagging. Hence, an independent binary classifier (fully connected layer of hidden size 2) is used for each of the taxonomy categories we train for, and those classifiers are ran for each STP, in an effort to predict whether it fits under that particular category or not (see Figure 5.3). Note that these classifiers are all models in their own right and do not share the same loss, i.e., they are trained independently. Also, we do not train for the 46 categories present in the taxonomy due to the very pronounced class imbalance which seemed to hinder the training process (see section below). As an input to these model, we feed in the encoder outputs from the standardizer part of the trained JOINT model. This means that the weights of JOINT are loaded while we train TAXONOMY. The layers of JOINT are frozen to exclude them from the loss and are just used for context to the SO tagging task.

A considerable amount of modeling experimentation has been carried out for this task. This includes joint training with the individual models or taking the output from the standardizer decoder instead of the encoder as input to the TAXONOMY. None of the aforementioned seemed to benefit any of the tasks.

| **Category** | **Precision** | **Recall** | **F1 score** |
|---|---|---|---|
| T-marbuta | 80.2 | 96.6 | 87.6 |
| Templatic Gemination | 38.1 | 33.3 | 35.6 |
| Root-radical Divergence | 80.0 | 42.1 | 55.2 |
| Split \| Clitic-stem | 44.4 | 25.0 | 32.0 |
| Phoneme-grapheme Mismatch | 38.5 | 26.3 | 31.2 |
| Enclitic Normalization \| Function Words | 50.0 | 81.8 | 62.1 |
| Alef Normalization | 77.8 | 100.0 | 87.5 |
| Enclitic Normalization \| Verbs | 66.7 | 66.7 | 66.7 |
| Enclitic Normalization \| Nouns | 42.9 | 75.0 | 54.5 |
| Homophone Function Words | 100.0 | 50.0 | 66.7 |
| Suffix Normalization \| Verbs | 87.5 | 77.8 | 82.4 |
| Prefix Normalization \| Imperfective | 75.0 | 75.0 | 75.0 |

Table 5.4: Report of the precision, recall, and F1 scores gathered for the taxonomy model of the 12 most frequent categories of the SO taxonomy sorted from most (top) to least (bottom) frequent.

### Results and Discussion

To evaluate this task we use precision, recall, and F1 score instead of accuracy due to the heavy class imbalance of the data. As seen in Figure 3.2 from Chapter 3, most tokens will have a negative label (binary classification) because the number of **Not Equal** STPs does not exceed 26% of all the STPs in the corpus. Adding to that the fine-grained distribution of the tags over the taxonomy, we get an overwehlmingly high number of negative labels, i.e., the STP does not contain the respective kind of SO inconsistency. Now, 33 of the 45 categories have less than 10 occurrences in the 9,600-token corpus. Hence, those were omitted, and no classifier was trained for them. The results for the classifiers that were trained can be found in Table 5.4.

In this table, the categories are sorted by order of frequency in the corpus. Before starting, note that due to class imbalance, the development set, for a big part of these categories, would not have more than 0.5% of positive examples. It is quickly apparent that the frequency of the tag in the corpus and the F1 score do not seem to be correlated. For instance, although more abundant than the subsequent categories, the *Templatic Gemination* category is a harder one to train for since it involves predicting whether there should have been the doubling diacritic ( ّ ∼) in a verb or not. This requires extensive knowledge of Arabic verb templates in their contexts, and even though the tagger context contains information about verb form, 71.5% F1 score for the verb form feature (see Table 5.2) might not have been good enough for this disambiguation. On another hand, the reason why the *Enclitic Normalization* for function words performs better than categories which are higher up the frequency chain might be because it is a much more regular phenomenon, and the examples in that category are similar. While *Clitic-stem Split* should be an easy category, the reason why it has a low score is most probably because split tokens are merged before being fed to the model. Otherwise, the medium score of *Enclictic Normalization* for function words and verbs might be due to the fact that they are separated into two different categories. Predicting at a more coarse-grained level of the taxonomy might definitely lead to better results and more insight for all categories.

Even though no clear thread could be woven from these results, the precision and recall (conservativeness) of the individual classifiers can be explained most of the time. Class imbalance can be solved by augmenting the dataset and retraining the system for each category. A context vector could be created from the concatenated outputs of these independent classifiers, and be fed to the JOINT model for additional context.

## 5.3 Conclusion

Through all of these experiments, first and foremost, we showed that we can reach state-of-the-art results for Lebanese Arabic for the tasks of SO standardization, morphological segmentation, and morphological tagging. Factoring in a DA BERT module also increases the system's performance non-trivially for all the tasks. On another hand, we can conclude from the experimentation with the SO tagging task that the class imbalance found in the distribution of inconsistencies is an issue which should be solved. Now that the types of SO are clearly defined, we can carefully augment a dataset in a way that eases the imbalance. With this new way of looking at SO, many new possibilities arise, and are all left for future research.

# Conclusion

Now is the time to wrap everything up and make sense of the gathered results. One of the biggest achievements of this thesis in my opinion, is giving the task of spontaneous orthography standardization a clearer shape. In the past, this task tended to be overlooked because approaching the problem was not straightforward, most notably due to its daunting linguistic component. The Arabic language is complex, and the relationship between the Arabic varieties and Modern Standard Arabic (MSA) is as complex. While this thesis did not tackle the processing of these dialects jointly, it did so for Lebanese Arabic as a proof of concept which can and should readily be extended to the other varieties, and at some point, all of them combined.

As stated in the author's note in the problem statement of the thesis, the goal was not to reach peak performance for a standardizer system. Rather, a better understanding of the task was sought, and was duly acquired. Even when the same amount of data is used for MSA and Dialectal Arabic (DA), i.e., the low-resource component is taken out of the picture, many DA varieties sill struggle with reaching the performance of state-of-the-art MSA results. This begs the question of whether we are approaching the standardization task in a correct way, and the door for that was opened in the present study.

Nothing in this thesis is meant to be final, and it should be used as a blueprint for future research, be it for the taxonomy that accounts for the different types of spontaneous orthography, or the models which for example seemed to really benefit from the addition of DA BERT context. The next step should be to carry out the tasks which were left out such as data augmentation, which is expected to really benefit standardization, and training jointly for multiple varieties, or varieties from the same region. In particular, the taxonomy created should also be tested in more varied models in conjunction with data augmentation, as the initial results seem promising.

Furthermore, Lebanese Arabic was shown not be grammarless as many native lay-speakers of the language might seem to think due to the ever-present diglossic situation which haunts the Arab world. Most importantly, there is no chaos in Lebanese Arabic grammar and most probably in any of the other varieties, as it follows strict morpho-phonological rules, some of which were thoroughly described throughout the thesis.

With this said, this study can safely be concluded and paves the way for more to come.

# Bibliography

Abdul-Mageed, Muhammad, AbdelRahim Elmadany, and El Moatez Billah Nagoudi (2020). *ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic.* arXiv: `2101.01785 [cs.CL]`.

Abu Kwaik, Kathrein, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik (May 2018). "Shami: A Corpus of Levantine Arabic Dialects". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA). URL: `https://www.aclweb.org/anthology/L18-1576`.

Alabi, Jesujoba O., Kwabena Amponsah-Kaakyire, David I. Adelani, and Cristina España-Bonet (2020). *Massive vs. Curated Word Embeddings for Low-Resourced Languages. The Case of Yorùbá and Twi.* arXiv: `1912.02481 [cs.CL]`.

Attia, Mohammed, Mohamed Al-Badrashiny, and Mona Diab (Oct. 2014). "GWU-HASP: Hybrid Arabic Spelling and Punctuation Corrector". In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP).* Doha, Qatar: Association for Computational Linguistics, pp. 148–154. DOI: `10.3115/v1/W14-3620`. URL: `https://www.aclweb.org/anthology/W14-3620`.

Bassiouney, Reem (2020). "Arabic Sociolinguistics: Topics in Diglossia, Gender, Identity, and Politics, Second Edition". In:

Bouamor, Houda et al. (May 2018). "The MADAR Arabic Dialect Corpus and Lexicon". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* Ed. by Nicoletta Calzolari (Conference chair) et al. Miyazaki, Japan: European Language Resources Association (ELRA). ISBN: 979-10-95546-00-9.

Chiang, David, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef (Jan. 2006). "Parsing Arabic Dialects." In:

Conneau, Alexis, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni (July 2018). "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Melbourne, Australia: Association for Computational Linguistics, pp. 2126–2136. DOI: `10.18653/v1/P18-1198`. URL: `https://aclanthology.org/P18-1198`.

Darwish, Kareem, Hamdy Mubarak, Mohamed Eldesouki, Ahmed Abdelali, Younes Samih, Randah Alharbi, Mohammed Attia, Walid Magdy, and Laura Kallmeyer (May 2018). "Multi-Dialect Arabic POS Tagging: A CRF Approach". In:

Dasigi, Pradeep and Mona Diab (Nov. 2011). "CODACT: Towards Identifying Orthographic Variants in Dialectal Arabic". In: *Proceedings of 5th Interna-*

*tional Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, pp. 318–326. URL: https://www.aclweb.org/anthology/I11-1036.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].

Elnagar, Ashraf, Sane M. Yagi, Ali Bou Nassif, Ismail Shahin, and Said A. Salloum (2021). "Systematic Literature Review of Dialectal Arabic: Identification and Detection". In: *IEEE Access* 9, pp. 31010–31042. DOI: 10.1109/ACCESS.2021.3059504.

Elsayed Abd Elaziz, Mohamed, Mohammed A. A. Al-qaness, Ahmed Ewees, and Dahou Abdelghani (Dec. 2019). *Recent Advances in NLP: The Case of Arabic Language*. ISBN: 978-3-030-34613-3. DOI: 10.1007/978-3-030-34614-0.

Erdmann, Alexander, Nasser Zalmout, and Nizar Habash (Jan. 2018). "Addressing Noise in Multidialectal Word Embeddings". In: pp. 558–565. DOI: 10.18653/v1/P18-2089.

Eryani, Fadhl, Nizar Habash, Houda Bouamor, and Salam Khalifa (May 2020). "A Spelling Correction Corpus for Multiple Arabic Dialects". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4130–4138. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.508.

Eskander, Ramy, Nizar Habash, Owen Rambow, and Nadi Tomeh (June 2013). "Processing Spontaneous Orthography". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 585–595. URL: https://www.aclweb.org/anthology/N13-1066.

Etoori, Pravallika, Manoj Chinnakotla, and Radhika Mamidi (July 2018). "Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning". In: *Proceedings of ACL 2018, Student Research Workshop*. Melbourne, Australia: Association for Computational Linguistics, pp. 146–152. DOI: 10.18653/v1/P18-3021. URL: https://www.aclweb.org/anthology/P18-3021.

Farra, Noura, Nadi Tomeh, Alla Rozovskaya, and Nizar Habash (June 2014). "Generalized Character-Level Spelling Error Correction". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 161–167. DOI: 10.3115/v1/P14-2027. URL: https://www.aclweb.org/anthology/P14-2027.

Ferguson, C. A. (1990). ""Come forth with a Surah like it": Arabic as a measure of Arab society". In:

Habash, Nizar (2010). *Introduction to Arabic Natural Language Processing*. Synthesis digital library of engineering and computer science. Morgan & Claypool Publishers. ISBN: 9781598297959. URL: https://books.google.com.lb/books?id=kRIHCnC74BoC.

Habash, Nizar, Fadhl Eryani, et al. (May 2018). "Unified Guidelines and Resources for Arabic Dialect Orthography". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Ed. by Nicoletta Calzolari (Conference chair) et al. Miyazaki, Japan: European Language Resources Association (ELRA). ISBN: 979-10-95546-00-9.

Habash, Nizar and Owen Rambow (Jan. 2006). "MAGEAD: A Morphological Analyzer and Generator for the Arabic Dialects". In: DOI: 10.3115/1220175.1220261.

Habash, Nizar, Abdelhadi Soudi, and Timothy Buckwalter (Jan. 2007). "On Arabic Transliteration". In: pp. 15–22. ISBN: 978-1-4020-6045-8. DOI: 10.1007/978-1-4020-6046-5_2.

Hajič, Jan, Otakar Smrž, Zemánek Petr, Jan Šnaidauf, and Emanuel Beška (Jan. 2004). "Prague Arabic Dependency Treebank: development in data and tools". In: *Proc. of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools*.

Heigold, Georg, Stalin Varanasi, Günter Neumann, and Josef van Genabith (Mar. 2018). "How Robust Are Character-Based Word Embeddings in Tagging and MT Against Wrod Scramlbing or Randdm Nouse?" In: *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*. Boston, MA: Association for Machine Translation in the Americas, pp. 68–80. URL: https://www.aclweb.org/anthology/W18-1807.

Himoro, Marcelo Yuji and Antonio Pareja-Lora (May 2020). "Towards a Spell Checker for Zamboanga Chavacano Orthography". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 2685–2697. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.327.

Hládek, Daniel, Ján Staš, and Matúš Pleva (2020). "Survey of Automatic Spelling Correction". In: *Electronics* 9.10. ISSN: 2079-9292. DOI: 10.3390/electronics9101670. URL: https://www.mdpi.com/2079-9292/9/10/1670.

Hochreiter, Sepp (Apr. 1998). "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6, pp. 107–116. DOI: 10.1142/S0218488598000094.

Jin, Di, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits (2020). *Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment*. arXiv: 1907.11932 [cs.CL].

Khalifa, Salam, Nizar Habash, Fadhl Eryani, Ossama Obeid, Dana Abdulrahim, and Meera Al Kaabi (May 2018). "A Morphologically Annotated Corpus of Emirati Arabic". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L18-1607.

Khalifa, Salam, Sara Hassan, and Nizar Habash (Apr. 2017). "A Morphological Analyzer for Gulf Arabic Verbs". In: *Proceedings of the Third Arabic Natural Language Processing Workshop*. Valencia, Spain: Association for Computational Linguistics, pp. 35–45. DOI: 10.18653/v1/W17-1305. URL: https://www.aclweb.org/anthology/W17-1305.

Kondratyuk, Daniel, Tomáš Gavenčiak, Milan Straka, and Jan Hajič (2018). "LemmaTag: Jointly Tagging and Lemmatizing for Morphologically Rich Languages with BRNNs". In: *Proceedings of the 2018 Conference on Empirical*

*Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 4921–4928. DOI: `10.18653/v1/D18-1532`. URL: `https://aclanthology.org/D18-1532`.

Luong, Minh-Thang and Christopher D. Manning (2016). "Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models". In: *CoRR* abs/1604.00788. arXiv: `1604.00788`. URL: `http://arxiv.org/abs/1604.00788`.

Maamouri, Mohamed, Sondos Krouna, Dalila Tabessi, Nadia Hamrouni, and Nizar Habash (Jan. 2012). "Egyptian Arabic Morphological Annotation Guidelines." In:

Makki, Elrabih Massoud (1983). "The Lebanese dialect of Arabic: Southern Region". PhD thesis. Georgetown University, pp. xvi+132.

Naïm, Samia (2016). "On interaction between external and internal markers in expressing aspect in Arabic dialect varieties." In: *Aspectuality and Temporality: Empirical and Theoretical Issues*. Ed. by Zlatka Guentcheva. Vol. 172. Studies in Language Companion Series. John Benjamins Publishing, pp. 325–354. DOI: `10.1075/slcs.172.10nai`. URL: `https://halshs.archives-ouvertes.fr/halshs-01802634`.

Ngo, Thi-Vinh, Thanh-Le Ha, Phuong-Thai Nguyen, and Le-Minh Nguyen (2019). "How Transformer Revitalizes Character-based Neural Machine Translation: An Investigation on Japanese-Vietnamese Translation Systems". In: *arXiv preprint arXiv:1910.02238*.

Obeid, Ossama, Salam Khalifa, Nizar Habash, Houda Bouamor, Wajdi Zaghouani, and Kemal Oflazer (May 2018). "MADARi: A Web Interface for Joint Arabic Morphological Annotation and Spelling Correction". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: `https://www.aclweb.org/anthology/L18-1415`.

Rothe, Sascha, Shashi Narayan, and Aliaksei Severyn (2020). "Leveraging Pretrained Checkpoints for Sequence Generation Tasks". In: *Transactions of the Association for Computational Linguistics* 8, pp. 264–280. DOI: `10.1162/tacl_a_00313`. URL: `https://www.aclweb.org/anthology/2020.tacl-1.18`.

Saito, Itsumi, Jun Suzuki, Kyosuke Nishida, Kugatsu Sadamitsu, Satoshi Kobashikawa, Ryo Masumura, Yuji Matsumoto, and Junji Tomita (2017). "Improving neural text normalization with data augmentation at character-and morphological levels". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 257–262.

Sajjad, Hassan, Kareem Darwish, and Yonatan Belinkov (Aug. 2013). "Translating Dialectal Arabic to English". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1–6. URL: `https://www.aclweb.org/anthology/P13-2001`.

Salloum, Wael and Nizar Habash (Dec. 2012). "Elissa: A Dialectal to Standard Arabic Machine Translation System". In: *Proceedings of COLING 2012: Demonstration Papers*. Mumbai, India: The COLING 2012 Organizing Committee, pp. 385–392. URL: `https://www.aclweb.org/anthology/C12-3048`.

Samih, Younes, Mohamed Eldesouki, Mohammed Attia, Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, and Laura Kallmeyer (Aug. 2017). “Learning from Relatives: Unified Dialectal Arabic Segmentation”. In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 432–441. DOI: `10.18653/v1/K17-1043`. URL: `https://www.aclweb.org/anthology/K17-1043`.

Sawaf, Hassan (Nov. 2010). “Arabic Dialect Handling in Hybrid Machine Translation”. In: *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

Shendy, Riham (Feb. 2019). “The Limitations of Reading to Young Children in Literary Arabic: The Unspoken Struggle with Arabic Diglossia”. In: *Theory and Practice in Language Studies* 9, p. 123. DOI: `10.17507/tpls.0902.01`.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). *Sequence to Sequence Learning with Neural Networks*. arXiv: `1409.3215 [cs.CL]`.

Taji, Dima, Nizar Habash, and Daniel Zeman (Apr. 2017a). “Universal Dependencies for Arabic”. Undefined. In: *Proceedings of the Third Arabic Natural Language Processing Workshop*. Association for Computational Linguistics (ACL), pp. 166–176. DOI: `10.18653/v1/W17-1320`.

— (Apr. 2017b). “Universal Dependencies for Arabic”. In: *Proceedings of the Third Arabic Natural Language Processing Workshop*. Valencia, Spain: Association for Computational Linguistics, pp. 166–176. DOI: `10.18653/v1/W17-1320`. URL: `https://www.aclweb.org/anthology/W17-1320`.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). *Attention Is All You Need*. arXiv: `1706.03762 [cs.CL]`.

Watson, Daniel, Nasser Zalmout, and Nizar Habash (2018). “Utilizing character and word embeddings for text normalization with sequence-to-sequence models”. In: *arXiv preprint arXiv:1809.01534*.

Zaidan, Omar F. and Chris Callison-Burch (Mar. 2014). “Arabic Dialect Identification”. In: *Computational Linguistics* 40.1, pp. 171–202. DOI: `10.1162/COLI_a_00169`. URL: `https://www.aclweb.org/anthology/J14-1006`.

Zalmout, Nasser, Alexander Erdmann, and Nizar Habash (June 2018). “Noise-Robust Morphological Disambiguation for Dialectal Arabic”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 953–964. DOI: `10.18653/v1/N18-1087`. URL: `https://www.aclweb.org/anthology/N18-1087`.

Zalmout, Nasser and Nizar Habash (July 2020). “Joint Diacritization, Lemmatization, Normalization, and Fine-Grained Morphological Tagging”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8297–8307. DOI: `10.18653/v1/2020.acl-main.736`. URL: `https://www.aclweb.org/anthology/2020.acl-main.736`.

Zbib, Rabih, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar Zaidan, and Chris Callison-Burch (2012). “Machine translation of Arabic dialects”. In: *Proceedings of the 2012* *conference of the north american chapter of the association for computational linguistics: Human language technologies*, pp. 49–59.

# Appendix A

# Transliteration Scheme

Arabic transliteration presented in the Habash-Soudi-Buckwalter scheme (Habash, Soudi, et al., 2007):

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| أ | ب | ت | ث | ج | ح | خ | د | ذ | ر | ز | س | ش | ص | ض | ط | ظ | ع | غ | ف | ق | ك | ل | م | ن | ه | و | ي |
| Â | b | t | θ | j | H | x | d | ð | r | z | s | š | S | D | T | Ď | ς | γ | f | q | k | l | m | n | h | w | y |

and the additional symbols:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ’ | Â | Ă | Ā | ŵ | ŷ | ħ | ý | ~ |
| ء | أ | إ | آ | ؤ | ئ | ة | ى | ّ |

There is a one-to-one mapping between the Arabic script letters and the letters of the transliteration scheme. This makes examples in Arabic easier to understand since the Arabic script is extremely cursive in style, and letters change forms depending on their context. This is especially useful seeing that our task deals with orthography standardization, and hence, can facilitate the appreciation of the given examples throughout this report by non-Arabic speakers or readers.

# Appendix B

# Word Alignment for Character-Level Models

To extemporaneously deal with the word-alignment problem, the first idea that comes to mind is to use a word-alignment model (e.g., any IBM model) to obtain those alignments. But after some initial testing with *fast_align*, the alignments turn out to be of very low quality. This is due to the nature of the IBM models which will work well with consistent data (i.e., with minimal amounts of noise), and because these alignment models operate at the word level without access to sub-word information.

The splitting (merging) of many source/target tokens is compounded by the fact that the split (merged) tokens also change internally. Hence great care must be taken as to the method of alignment. By virtue of the great similarity between the source and target sentences – remember that both source and target are in the same language –, a divide-and-conquer approach can be adopted to look at the alignment problem as a sequence matching problem, by describing the source-target pairs as a series of *substitutions*, *insertions*, *deletions*, and *do-nothing's*. For example, Figure B.1 shows how such a pair would be aligned by the proposed system. A sequence matcher[1] would first, based on a longest-common-sequence approach, take in the two pairs (top of Figure B.1) and output a *diff*-sequence which describes what types of operations occurred going from the source to the target side. Around 80% of the operations will be do-nothings (green boxes) so this reduces our problem to actually aligning relatively short sequences, such as the boxed sequences in the bottom of Figure B.1 (longest one recorded is of length 9).

The alignment procedure within the short sequences (boxed) is also performed using edit distance, although this time, at the character level. The algorithm works well but will not capture very complex cases when multiple concurrent splits and merges happen between a source-target pair. However, these cases happen rarely, and the algorithm works more than 99% of the time. Evaluation of the alignments happens by joining the source (target) sequence (i.e., a sentence) elements

---

[1] Used the Sequence Matcher class provided in the *edit_distance* library at `https://github.com/belambert/edit-distance`

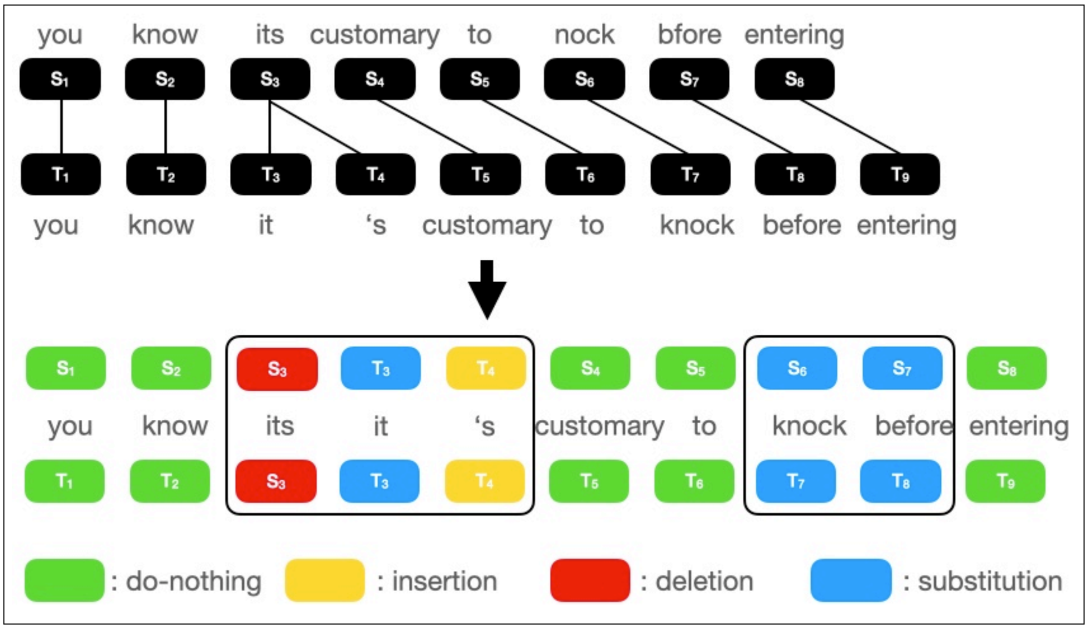


Figure B.1: Example of aligning a source-target pair at the word level.

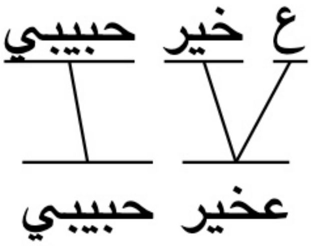


Figure B.2: Correct alignment of the source *ς␣xyr␣Hbyby* (top) - target *ςxyr␣Hbyby* (bottom) pair.

| | Source | | | Target | |
|---|---|---|---|---|---|
| **Indexes** | 1 | 0 | 0 | 1 | 0 |
| **Aligned** | حبيبي<br>*Hbyby* | خير<br>*xyr* | ع<br>*ς* | حبيبي<br>*Hbyby* | عخير<br>*ςxyr* |
| **Example 1** | ع خير<br>*ς␣xyr* | | | عخير<br>*ςxyr* | |
| **Example 2** | حبيبي<br>*Hbyby* | | | حبيبي<br>*Hbyby* | |

Table B.1: Generated training examples from the pair in Figure B.2 which will be fed in to the character-level models.

using a space delimiter and comparing to the string version of that sequence. This evaluation procedure is not foolproof as it can sometimes output false positives/negatives. This method leaves much space for improvement, however, it was settled on due to times restrictions. Figure B.2 shows how a source-target pair is aligned, and Table B.1 shows how training examples are generated from those alignments.

# Appendix C

# Training Details

All the models in Chapters 2 and 5 were built using PyTorch 1.8 and were trained using an NVIDIA V100 Tensor Core GPU. All of the models were built in-house except for the CRF module[1], as no other third-party model was used. All of the model hyper-parameters are recorded in Table C.1, and below are some comments on the choice of the latter.

- **Batch size** was an important factor for the tasks undertaken, and the lower the batch size, the better the systems' performance was. Lowering the batch size can be considered as a regularization factor, and since we are dealing with small datasets, lowering it might have helped the generalization process.
- **Dropout** As an explicit regularization factor, it was always placed between the embedding layers and the input of the RNNs.
- **Loss** Cross-entropy was used to calculate the loss for all the models. For the joint models, the average loss, i.e., the sum of the loss for each task divided by the number of tasks, was calculated.
- **Decoding** Greedy decoding was used all throughout because no suitable Beam Search decoding library could be found to work with our in-house models, and building one would have been very time-consuming.
- **Seed** A seed of 42 was used for all the experiments for reproducibility, however, no cross-validation or model ensembling was used due to lack of time.

---

[1] Used the library found at `https://pytorch-crf.readthedocs.io/en/stable/`.

| | MERGER | WL | CL-CTX | WL-HYBRID | SEGMENTER | TAGGER | JOINT |
|---|---|---|---|---|---|---|---|
| **Optimization** | Adam | | | | | | |
| **Epochs** | 32 | 25 | 23 | NA | 25 | 15 | 15 |
| **Batch Size (in Sentences)** | 64 | 32 | 32 | 32 | 16 | 4 | 4 |
| **Dropout** | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| **Learning Rate** | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 |
| **Schedule** | Reduce on plateau by a 0.5 factor with patience 4 | | | | | | |
| **RNN Cell** | LSTM | | | | | | |
| **Character Embedding Dimension** | 32 | 128 | 128 | 128 | 128 | 128 | 128 |
| **Word Embedding Dimension** | N/A | | 256 | | N/A | | |
| **Sentence Embedding Dimension** | 768 (BERT) | N/A | | | | | |
| **Character Hidden Dimension** | 64 | 256 | 256 | 256 | 256 | 256 | 256 |
| **Word Hidden Dimension** | N/A | | 512 | | N/A | | |
| **Segment Hidden Dimension** | N/A | | | | 256 | | |
| **Context Window** | N/A | | | | | 7 | 7 |
| **Number of Hidden Layers** | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| **Attention Type** | N/A | Global (Luong) | | | N/A | | Global (Luong) |
| **Attention Dimension** | | 16 | 16 | 16 | | | 16 |
| **Decoding** | | Greedy | | | | | Greedy |
| **Train/test Split** | 0.9 | | | | | | |
| **Maximum Sentence Length (in Tokens)** | 35 | | | | | | |
| **Maximum Characters per Token** | 25 | | | | | | |

Table C.1: Table showing the values of the hyper-parameters for the main models described throughout the thesis report.

# Appendix D

# Lebanese Arabic Verbs Conjugation

Table D.1 is an ad-hoc solution which serves as a reference to conjugate Lebanese Arabic verbs. It was created by phonologically and morphologically analyzing verbs that are common to MSA and Lebanese Arabic, according to the verb templates they adhere to, as MSA and Lebanese Arabic verbs follow the same set of templates (right-most column). It was not based on any scholarly reference of DA. However, in and by itself, it forms a consistent whole which serves our purposes. This table tries to unify all ways of writing different verb inflections while taking the different pronunciations encountered in Lebanese Arabic into consideration. Verbs and their spellings are especially cumbersome since their morphology is very productive. Adding to this the phonological changes of Lebanese Arabic, spelling certain verbs consistently while retaining the morpho-phonological structure of the dialectal form should be handled carefully. The orthography choices made in this table are discussed at length in Section 3.4.3.

## D.1 Tri-literal verbs

The tri-literal verb forms which are below the first row – belonging to the sound and stripped tri-literal verb form ( فعل ثلاثي مجرّد سالم) – were unrolled to point out the phonological differences springing from their conjugation[1].

### D.1.1 Stripped tri-literal verbs

In this section, we look at some patterns which are consistent across the different verb forms.

#### Command Aspect

The first pattern we notice is that all tri-literal non-augmented verbs exhibit middle vowel elongation in the command aspect (صيغة الأمر). The only exception

---

[1] These are used in Section 3.4.3 to infer spelling standardization rules.

| Template | Perfective | | Imperfective | | | | | Command |
|---|---|---|---|---|---|---|---|---|
| | Active | Passive | Active | | | | Passive | |
| | | | b-1S | b-3MS | 1S | 3MS | | |
| **Stripped Tri-literal Verbs** | | | | | | | | |
| فعل | كسر | انفعل | بكسر | بيكسر | اكسر | يكسر | ينفعل | كسور |
| وعل | وصف | انفعل | بوصف | بيوصف | اوصف | يوصف | ينفعل | وصوف |
| يعل | يبس | تفعّل | بيبس | بييبس | ايبس | ييبس | يتفعّل | يباس |
| اعل | اخذ | افتعل | بآخذ | بيآخذ | آخذ | يآخذ | يفتعل | خوذ |
| ف{و\|ي}ل | قام | انفعل | بقوم | بيقوم | قوم | يقوم | ينفعل | قوم |
| فع{و\|ي} | رضي | انفعل | برضى | بيرضى | ارضى | يرضى | ينفعل | رضى |
| ف{و\|ي}{و\|ي} | شوى | انفعل | بشوي | بيشوي | اشوي | يشوي | ينفعل | شوي |
| فعّ | مدّ | انفعل | بمدّ | بيمدّ | مدّ | يمدّ | ينفعل | مدّ |
| {و\|ي}ع{و\|ي} | وعى | تفعّل | بوعى | بيوعى | اوعى | يوعى | يتفعّل | وعى |
| **Augmented Tri-literal Verbs** | | | | | | | | |
| فعّل | نضّف | تفعّل | بنضّف | بينضّف | نضّف | ينضّف | يتفعّل | نضّف |
| فاعل | شارط | ∅ | بشارط | بيشارط | شارط | يشارط | ∅ | شارط |
| افعل | اجرم | ∅ | باجرم | بياجرم | اجرم | ياجرم | ∅ | اجرم |
| تفاعل | تفاعل | ∅ | بتفاعل | بيتفاعل | اتفاعل | يتفاعل | ∅ | تفاعل |
| انفعل | انفعل | ∅ | بنفعل | بينفعل | انفعل | ينفعل | ∅ | انفعل |
| افتعل | افتعل | ∅ | بفتعل | بيفتعل | افتعل | يفتعل | ∅ | افتعل |
| افعلّ | انضمّ | ∅ | بنضمّ | بينضمّ | انضمّ | ينضمّ | ∅ | انضمّ |
| استفعل | استعلم | ∅ | بستعلم | بيستعلم | استعلم | يستعلم | ∅ | استعلم |
| افعوعل | ازرورق | ∅ | بزرورق | بيزرورق | ازرورق | يزرورق | ∅ | ازرورق |
| **Stripped Quadri-literal Verbs** | | | | | | | | |
| فعلل | ترشق | ∅ | بترشق | بيترشق | ترشق | يترشق | ∅ | ترشق |
| فوعل | فونص | ∅ | بفونص | بيفونص | فونص | يفونص | ∅ | فونص |
| فنعل | فنتر | ∅ | بفنتر | بيفنتر | فنتر | يفنتر | ∅ | فنتر |
| فيعل | طيلع | ∅ | بطيلع | بيطيلع | طيلع | يطيلع | ∅ | طيلع |
| فعلن | شرعن | ∅ | بشرعن | بيشرعن | شرعن | يشرعن | ∅ | شرعن |
| **Agumented Quadri-literal Verbs** | | | | | | | | |
| تفعلل | تسلسل | ∅ | بتسلسل | بيتسلسل | اتسلسل | يتسلسل | ∅ | تسلسل |
| تمفعل | تمقطع | ∅ | بتمقطع | بيتمقطع | اتمقطع | يتمقطع | ∅ | تمقطع |
| تفوعل | تمحور | ∅ | بتمحور | بيتمحور | اتمحور | يتمحور | ∅ | تمحور |
| تفعلن | تشرعن | ∅ | بتشرعن | بيتشرعن | اتشرعن | يتشرعن | ∅ | تشرعن |

Table D.1: Conjugation table of Lebanese Arabic verbs based on the different verb templates. *3* and *1* refer to the third and first person; *M* and *S* refer to masculine and singular; *b-* refers to the ب *b-* prefix marker. {و|ي} means any semi-consonant. Table should be read from right to left.

| Command + Clitics | Perfective | | | Template |
|---|---|---|---|---|
| | Active + Clitics | | Active | |
| | M | F | | |
| **Stripped Tri-literal Verbs** | | | | |
| كسرلك | كسرلك | كسرتلك | كسر | فعل |
| وصفلك | وصفلك | وصفتلك | وصف | وعل |
| يبسلك | يبسلك | يبستلك | يبس | يعل |
| خذلك | اخدلك | اخدتلك | اخذ | اعل |
| قملك | قملك | قامتلك | قام | ف{واي}ل |
| رضيلك | رضيلك | رضيتلك | رضي | فع{واي} |
| شويلك | شويلك | شويتلك | شوى | ف{واي}{واي} |
| مدّلك | مدّلك | مدّتلك | مدّ | فعّ |
| وعيلك | وعيلك | وعيتلك | وعى | {واي}ع{واي} |

Table D.2: Conjugation table of Lebanese Arabic verbs based on the different verb templates. *M* and *F* refer to masculine and feminine. {واي} means any semi-consonant. Table should be read from right to left.

is the doubled tri-literal form (افعال مضعّفة), in which no elongation takes place.

### Passive Voice

Concerning the passive voice (المجهول), Lebanese Arabic verbs seem to follow a strict logic as to how they are realized. The main difference with realizing the passive voice grammatical feature when compared to MSA is that in the latter, verbs are conjugated (مصرّف) into the passive, while in Lebanese Arabic, another verb form (template) is borrowed from the augmented tri-literal verb forms to do just that (third column from the right). The second inferred pattern is that all tri-literals borrow the انفعل template to form the passive except *yaa'-beginning* verbs (افعال يائية), verbs which have defective letters (احرف علّة) in the beginning and ending position (افعال لفيفة مفروقة), and finally *hamza-beginning* verbs (افعال مهموزة). The first two use the تفعّل template to form the passive, and the third one the افتعل template.

### Prefix Pronunciation in the Active Imperfective Aspect

Another interesting thing happens when we analyze the pronunciation in Lebanese Arabic of the active imperfective verbs (مضارع معلوم) with the *b*- prefix marker[2] potentially attached. We only analyze the third and first person, and masculine singular versions of the verbs because only those pose phonological problems. The pronunciation and spelling of the rest (feminine, plural, and second person) are inferable unambiguously from MSA grammar rules. In all stripped tri-literal verbs, the third person ي *y* prefix[3] seems to be pronounced as the consonant /j/ when in the third person when the *b*- prefix is used in conjunction, but with two

---

[2] This prefix does not exist in MSA and poses some pronunciation/spelling problems here as it interacts with the verb's phonology.

[3] Inflects based on gender, number, and person.

exceptions, namely, hollow verbs (افعال جوفاء), and doubled verbs, in which it is assimilated and realized as /i/ and /∅/ respectively. With these two verb forms, a similar phenomenon happens for the first person singular without the *b*- prefix, wherein the ا *A* /ʔ/ prefix is not pronounced. Using the imperfective aspect without the *b*- prefix happens in a few cases, such as directly after a conjunction, or when the imperfective verb is followed by a *beginning verb* (فعل شروع), or a modal verb. For example, in the sentence كنت ضلّ روح لهونيك *knt Dll rwH lhwnyk* ‘I always used to go there’, the verb روح *rwH* ‘go’ /ruːħ/ is in the imperfective but does not take the *b*- prefix. But since it is a hollow verb, the ا *A* /ʔ/ prefix is not pronounced.

### Attaching Clitics

When clitics are attached to the active perfective or command verbs, they can interfere with the phonology of the verb. Particularly, they induce spelling ambiguities at the gender level. When attached to an active perfective hollow verb, they instigate a middle vowel shortening for the masculine and a middle vowel elongation for the feminine. No other verb form shows this feature. In the command verbs, the vowel elongation that happens with no attached clitic (see Section D.1.1) is retracted except for the defective verbs (الافعال المعتلّة الآخر) which have a semi-consonant (ا و ي ى *ýy w A*) as the final radical[4] since the latter can function as vowels in the phonetic sense. Augmented tri-literal and quadri-literal verbs do not have this issue.

## D.1.2 Augmented tri-literal verbs

The rules for the augmented tri-literal verbs (افعال رباعية المزيد) are simpler than for the stripped versions.

### Command

All augmented tri-literal command verbs seem to be either pronounced the same way as the root form of a verb (active, third person, perfective) or with the next to last vowel changed from /a/ to /ɪ/.

### Passive Voice

It seems that in Lebanese Arabic, there is no way of directly forming the passive voice of an augmented tri-literal verb. However, some of these verb forms in the right context can inherently carry the passive voice feature without even being inflected like انفعل or تفعّل . One exception to this is the فعّل template, the passive of which can always be realized using the تفعّل template. One way of emulating the passive voice for some of these forms is by using the plural form in conjunction with a pronoun enclitic.

---

[4] Almost every Arabic word – except proper nouns and foreign words – has a *root* that consists of three or four letters. These are called *radicals*.

### Prefix Pronunciation in the Active Imperfective Aspect

All the augmented tri-literals in Lebanese Arabic have the property that the third person ي *y* prefix is pronounced as the consonant /j/ when the *b*- prefix is used in conjunction, but with three exceptions, namely, فعّل, فاعل, and أفعل, in which it is assimilated and realized as /i/ and /∅/ respectively. The same happens for the first-person prefix as with the stripped forms.

## D.2 Quadri-literal verbs

For quadri-literal verbs (افعال رباعية), the pattern is more obvious in all cases. For the command aspect and passive voice, the same rules are followed as for tri-literal augmented verbs. For the pronunciation of the prefixes, the third person ي *y* prefix seems to be pronounced as the consonant /j/ when the *b*- prefix is used in conjunction, only for the augmented quadri-literal verbs. For the stripped quadri-literals, it is assimilated and realized as /i/. The same goes for the first-person prefix as explained above.