DIALECTAL ARABIC PROCESSING USING DEEP LEARNING

Inaugural-Dissertation
zur Erlangung des Doktorgrades der Philosophie (Dr. phil.)
durch die Philosophische Fakultät der
Heinrich-Heine-Universität Düsseldorf

vorgelegt von
Younes Samih

aus
ESSEN

Betreuerin:
Prof. Dr. Laura Kallmeyer

Düsseldorf, Oktober 2017

ABSTRACT

In the last few years, the advent of the phenomena of social media and the ubiquity of the internet access have created an unprecedented deluge of information and textual data on the world wide web. This data brings in its wake new opportunities and poses many challenges for machine learning and Natural Language Processing (NLP) in particular. The sheer size, non-standard spelling, the poor quality, the informality, and the noise of this data, presents new challenges to standard NLP tools developed for traditional data. To make sense out of such data and exploit its value, novel NLP methods, resources, and efficient algorithms beyond rule-based deductive reasoning and "traditional" system engineering need to be created. Given the ground breaking results of Deep Learning (DL) models in solving hard natural language processing tasks, I argue in this dissertation that these models are well suited for processing social Media textual data. A case in point is Dialectal Arabic (DA) which is emerging as the language of informal communication on the web, in emails, Social Media platforms, blogs, etc. To systematically investigate the ability of DL models to process the less controlled and more speech-like nature of DA in Social Media, I choose to address two concrete, challenging tasks, namely linguistic Code-Switching (CS) identification and DA morphological segmentation.

ZUSAMMENFASSUNG

In den letzten Jahren ist durch die sozialen Medien und allgegenwärtigen Zugang zum Internet eine noch nie dagewesene Flut von Information und Textdaten im Internet entstanden. Diese Daten bringen neue Chancen und vielfältige Herausforderungen im Bereich des machinellen Lernens, und insbesondere in der maschinellen Sprachverarbeitung NLP. Die schiere Menge, nicht den gewohnten Normen entsprechende Rechtschreibung, die schlechte Qualität und Informalität, und das Rauschen in den Daten stellen die NLP-Standardwerkzeuge, die auf traditionellen Daten entwickelt worden sind, vor neue Probleme.

Um solche Daten interpretieren und ihren Wert nutzen zu können, müssen neue NLP-Methoden, Resourcen und effiziente Algorithmen jenseits von regelbasiertem dekutiven Herangehensweisen und "traditionellen"Methoden der Systementwicklung geschaffen werden. Im Licht der bahnbrechenden Ergebnisse von Modellen des Deep Learning (DL) argumentiere ich in dieser Dissertation, dass solche Modelle gut zur Verarbeitung von Text aus sozialen Medien geeignet sind. Als Beispielfall dient das Dialectal Arabic (DA), das sich zur Sprache der informellen Kommunikation im Web, in E-Mails, auf sozialen Medien, und in Blogs entwickelt hat. Um die Möglichkeiten von DL-Modellen zu zeigen, das weniger kontrollierte und der gesprochenen Sprache ähnliche DA in sozialen Medien zu verarbeiten, bearbeite ich zwei konkrete, herausfordernde Probleme, und zwar die Identifikation von linguistischem Code-Switching (CS) und die morphologische Segmentierung von DA.

AUTHOR STATEMENT

I would like here to acknowledge that Chapter 4 "Data and Resource Creation" includes work that was published as Samih and Maier (2016a) and Samih and Maier (2016b). I was primarily responsible for the design and creation of the corpus described in these two papers. I was also an active collaborator throughout the creation of the multi-dialectal segmentation corpus presented in the following papers, Samih et al. (2017a), Samih et al. (2017b) Samih, Maier, and Kallmeyer (2016), and Eldesouki et al. (2017).

Chapter 5 "Code-switching identification" also includes work that was published as Samih and Maier (2016a,b) and Samih, Maier, and Kallmeyer (2016). I was primarily responsible all the experiments and the design and creation of the corpus used in this chapter. The other authors contributed to these papers primarily in an advisory role. In addition, Section 5.5 in this chapter presents work that is excerpted from Samih et al. (2016), which also forms the basis for much of this chapter. I was responsible for the design of the neural models presented in that paper.

Chapter 6 "Dialectal Arabic Segmentation" also includes work that was published as Eldesouki et al. (2017) and Samih et al. (2017a,b). I was responsible for the design of the neural models presented in these three papers. I substantially contributed to the implementation and testing of these models and in the development of the segmentation algorithm. Eldesouki Mohamed was responsible for the implementation of the Support Vector Machine (SVM) models. The other authors contributed to these papers primarily in an advisory role.

Appendix A "SAWT" is an updated version of my paper titled "SAWT: Sequence Annotation Web Tool" presented in the following paper Samih, Maier, and Kallmeyer (2016).

ACKNOWLEDGMENTS

I owe an enormous debt of gratitude to my supervisor, Laura Kallmeyer, for many reasons. First she put me in the right track since Day One saving me a lot of time and effort exploring uncountable paths. Second she kept motivating and supporting me along the way. She encouraged me to write papers, and attend meetings and conferences. She also provided the financial funds needed for travelling to these research events. Third, at times when I was about to go off rail or lose confidence, her advice helped put me back in the right track.

My deepest gratitude goes to my affectionate mother and my dearest father, for everything they have provided, and to my family (my wife Ikram and my son Adam), to whom this dissertation is dedicated, for their love, patience and unfailing support.

Mohammed Attia has been my post-doc mentor even before my dissertation. I am so thankful for all the detailed comments he has made during all these years. He has always been willing to discuss various aspect of my dissertation or read a chapter. I am really grateful for his support.

I am grateful that I could finish my work at the University of Düsseldorf among nice and supportive colleagues. In particular I would like to thank Wolfgang Maier and Miriam Kaeshammer.

I would also like to thank James Kilbury, Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, Mohamed Eldesouki, Timm Lichte, Christian Wurm, Simon Petitjean, and Andreas van Cranenburgh for discussing various as-

pects of my dissertation with me and for giving me insightful comments. All remaining errors are my own!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CS	Code-Switching
CRF	Conditional Random Field
SVM	Support Vector Machine
RNN	Recurrent Neural Network
DNN	Deep Neural Network
NLP	Natural Language Processing
SGD	Stochatic Gradient Descent
PCA	Principle Component Analysis
IRQ	Iraqi Arabic
EGY	Egyptian Arabic
LEV	Levantine Arabic
GLF	Gulf Arabic
MGR	Maghrebi Arabic
DA	Dialectal Arabic
CA	Classical Arabic
MSA	Modern Standard Arabic
NLP	Natural Language Processing

DL	Deep Learning
ML	Machine Learning
ANN	Artificial Neural Network
DBN	Deep Belief Network
RBM	Restricted Boltzmann Machine
BPA	Backpropagation Algorithm
HMM	Hidden Markov Models
NN	Neural Network
MT	Machine Translation
LSTM	Long short-term memory
BiLSTM	Bidirectional Long Short-Term Memory
CODA	Conventional Orthography for Dialectal Arabic
L-BFGS	Limited-memory BFGS
LM	Language Model
OOV	Out-Of-Vocabulary

Part I

INTRODUCTION

1

INTRODUCTION

1.1 OVERVIEW AND MOTIVATION

Arabic is a Semitic[1] language. The term *Arabic* generally refers to a group of language varieties that are historically, genetically, and linguistically related. Arabic has multiple varieties. MSA is the formal standard language across the Arab world. It is used in the written media, news, and education. In contrast, DA is restricted to informal daily communication. However, this is changing with the advent of social media and the ubiquity of internet access. DA is increasingly emerging as the language of informal communication on the web, in emails, micro-blogs, blogs, forums, Social Media networks, etc. This new situation has created an unprecedented deluge of textual data on the world wide web. This data brings in its wake new opportunities, but also poses many challenges to Machine Learning (ML) and NLP in particular. The sheer size, non-standard spelling, the poor quality, the informality, and the noise of this data present additional challenges to standard NLP tools developed for traditional data (Farzindar and Inkpen, 2015).

For example, applying standard NLP systems and methods tailored for MSA directly to DA texts yields significant degradation in their performance. In order to process and extract meaningful patterns from this massive input DA data, novel NLP methods, resources, and efficient algorithms beyond rule-

1 The Semitic language family includes Akkadian, **Amharic**, Amorite, **Arabic**, Aramaic, Ge'ez, Gurage, **Hebrew**, Maltese, Moabite, Nabatean, Phonoecian, Punic, **Syrica**, **Tigrinya**, Tigre, and Ugaritic. The five most widely spoken ones are written in bold https://en.wikipedia.org/wiki/Semitic_languages Web. 20 Oct. 2017.

based deductive reasoning and "traditional" system engineering need to be created.

My main research goal in this dissertation is to devise learning models that can automatically process user-generated Social Media textual data in particular, which contains DA. The main task, therefore, consists in the development and evaluation of Neural Network (NN) techniques for the natural language processing of the noisy and strongly contextualised nature of DA, and to use these results both to improve our understanding of existing models and as a basis on which to develop better models.

Over the last decade, Deep Neural Network (DNN) models have repeatedly proven to be effective in solving a wide range of NLP related problems including Machine Translation (MT) (Bahdanau, Cho, and Bengio, 2014), text classification (Kim, 2014), language models (Kim et al., 2016; Mikolov et al., 2010), multilingual processing (Gillick et al., 2015), parsing (Chen and Manning, 2014), Multi-task NLP (Collobert et al., 2011), named entity recognition (Lample et al., 2016), and sentiment analysis (Socher et al., 2013). Much of the success of these models can be attributed to their excellent capability for extracting features from raw data without any prior knowledge or assumptions about the statistical distribution of this data.

In order to systematically investigate the ability of DNN models to process the less controlled and more speech-like nature of DA in Social Media, we choose to address two concrete, challenging tasks, namely linguistic CS identification and DA morphological segmentation.[2] Since we regard these two tasks as sequence labelling problems, they can easily be solved with standard machine learning methods, such as SVM (Cortes and Vapnik, 1995) or CRF (Lafferty, McCallum, and Pereira, 2001), to name but a few. Yet, these

2 By segmentation we designate the separation of linear transcriptions of DA words into their constituent morphological parts. This is important for a variety of applications such as machine translation, parsing and information retrieval.

models can hardly succeed on the two tasks without reliance on external resources or carefully hand-crafted features, which are time-consuming to construct and are often both over-specified and incomplete. There are several reasons that motivate me to pursue the development of DNN models for processing DA:

- The knowledge learnt from DNN models is still largely untapped in the context of DA processing.
- Given the pervasive nature of DA, DNNs generalize well on unseen data, and they are also suitable to deal with outlying, missing, unstructured, and noisy data.
- DNNs are very flexible and can be paired with other techniques or models to harness the strengths and advantages of both techniques.
- DNNs are also inherently non-linear. This makes them more practical and accurate in modelling complex data patterns as opposed to many traditional methods which are linear. In numerous real world problems including those in the fields of NLP, DNNs have been shown to outperform linear models in classification and data analysis tasks.
- DNNs can assist with specific problems related to learning from large amounts of unsupervised DA data. They have the ability to learn data representations[3] (features) in a greedy layer-wise fashion (Bengio et al., 2007).

1.2 CONTRIBUTIONS OF THIS DISSERTATION

This dissertation makes the following concrete contributions:

3 Note that these representations need not correspond in any interpretable way to linguistically motivated representations typically used in theoretical linguistics.

- DA represents the general realm of problems for processing massive volumes of raw data. In Chapter 1, Chapter 3, and throughout the dissertation I introduce and motivate the use of DNNs as highly efective techniques for NLP of DA in Social Media.

- In Chapter 4 I contribute two dialectal corpora, the *Moroccan Arabic Darija code-switching corpus* and the *Twitter multi-dialectal Arabic segmentation corpus*[4]), with token-level annotations. Both corpora have been collected from internet discussion forums and blogs and from social media platforms. They will be of use for supporting research in the linguistics and NLP and will constitute an ideal data source for DA processing.

- In Chapter 5 I demonstrate that traditional machine learning and feature engineering methods are not efficient enough to extract the complex and non-linear patterns generally observed in the *Moroccan Arabic Darija code-switching corpus*. Conversely, the application of DNNs greatly improves CS identification results in DA.

- In Chapter 6 I demonstrate that by using DNNs one can easily build a unified segmentation model where the training data for different dialects is combined and a single model is trained. The DNN model yields higher accuracies than dialect-specific models, eliminating the need for dialect identification before segmentation. I also measure the degree of relatedness between four major Arabic dialects by testing how a segmentation model trained on one dialect performs on the other dialects. I found that linguistic relatedness is contingent on geographical proximity.

4 This dataset is available here: http://alt.qcri.org/resources/da_resources/releases/current/seg_data_jun122017.tgz

1.3 DISSERTATION OVERVIEW

The natural language processing of DA is the main aim of this dissertation. Chapter 2 gives a detailed description of the different linguistic aspects of Arabic and its dialects which are involved in most of the NLP tasks. It also emphasizes the peculiarities of DA which constitute a challenge for most traditional NLP techniques. Next, Chapter 3 introduces general neural networks and describes the key technical machinery used in this dissertation, namely Recurrent Neural Networks (RNNs). Chapter 4 introduces two datasets, the *Moroccan Arabic Darija code-switching corpus* and the *Twitter multi-dialectal Arabic segmentation corpus*. These two corpora constitute a test bed for DNN. Next, Chapter 5 presents a system for identifying and classifying code-switched data for MSA-Moroccan Arabic. The system uses a neural network architecture that relies on word-level and character-level representations. This system is language independent in the sense that it does not use any language-specific knowledge or linguistic resources such as POS taggers, morphological analyzers, gazetteers, or word lists to achieve state-of-the-art performance. Chapter 6 shows that using DNN, one can easily build a unified segmentation model using limited DA labeled data without relying on lexicons, morphological analyzers, or linguistic knowledge. The results obtained are comparable to that of a state-of-the-art system, or even better.

Part II

BACKGROUND

2

ARABIC AND ITS DIALECTS

This chapter provides background material and a review of literature on the characteristics of the Arabic language and its Dialects. The chapter is organized as follows: In section 2.1 we provide an overview of the different varieties of Arabic. In section 2.2 we briefly describe some linguistic features that are shared by the Arabic dialects. This is followed in section 2.3 by a description of the motivation for processing Arabic Dialects. Finally, we specifically focus on the challenges inherent in processing Dialectal Arabic (DA) in section 2.4.

2.1 THE ARABIC LANGUAGE VARIETIES

Arabic is one of the oldest living Semitic languages in the world. It is spoken by a population of almost 300 million people[1] and is the official language across the 22 so-called Arab League countries[2]. The term *Arabic* loosely refers to a group of language varieties which, in spite of their substantial mutual differences, are historically, genetically, and linguistically related. To define any clear-cut distinction between them on geographical, structural, historical, social, or even ethnographic grounds is rather confusing for many non-Arabic speaking researchers who have less exposure to the Arabic language

1 https://www.ethnologue.com/language/ara Web. 29 Apr. 2017

2 The members of the Arab league are Algeria, Bahrain, Comoros, Djibouti, Egypt, Iraq, Jordan, Kuwait, Lebanon, Libya, Mauritania, Morocco, Oman, Palestine, Qatar, Saudi Arabia, Somalia, Sudan, Syria, Tunisia, The United Arab Emirates, and Yemen (Habash, 2010)

and culture (Albirini, 2016). In the literature, Arabic varieties are categorized into three general types:

1. Classical Arabic (CA)
2. Modern Standard Arabic (MSA)
3. Dialectal Arabic (DA)

Classical Arabic is often identified as the language of the Qur'an[3] and the Hadith[4]. It is accorded an elevated status throughout the Muslim world as it is essential for understanding the Qur'an, the Hadith, the Islamic literary tradition, and law. It is not different from what it was 1400 years ago and has a fixed orthography and grammar.

On the other hand, MSA is an artificial language in that it is no one native language and is the lingua franca of the Arabic world. It is commonly used for formal, literary, and educational purposes across the Arabic speaking countries. It exhibits relatively minor variation, mainly in vocabulary, morphology, and phonological features (Holes, 2004). For a more detailed account of the different linguistic aspects of MSA, see Habash (2010), Holes (2004), and Ryding (2005).

Sharply contrasted with both MSA and CA are the dialectal Arabic varieties used in informal communication throughout the Arabic world. These language varieties not only show considerable variation from one country to another, but also differ from one region to another within the same county (Albirini, 2016; Hetzron, 1997). For example, speakers from the Gulf and Maghreb regions can hardly understand one another's dialects. Although Maghrebi and Gulf dialects share a plethora of linguistic features to warrant their subsumability under the Arabic language, they are not mutually intel-

3 Islam's Holy Book
4 the way of life prescribed as normative for Muslims on the basis of the teachings and practices of the prophet Muhammad, *Peace be upon him* and interpretations of the Qur'an

ligible in spoken discourse. These variations between the Arabic dialects can generally be attributed to the multiple morpho-syntactic processes of simplification and mutation, as well as coinage and borrowing of new words in addition to semantic shifts of standard lexical items. Furthermore, there is a considerable effect of cross-fertilization between the MSA varieties that spread throughout the Middle East and North Africa and the indigenous languages in different countries as well as neighboring languages. With the passage of time and the juxtaposition of cultures, dialects and variants of Arabic evolved and diverged. Figure 2.1 shows a simplified schematic map of the Arabic dialects based on geolinguistic features.

In face of the complex sociolinguistic situation in the Arab world and the diversity and the fluidity of the geolinguistic distinction between the Arabic dialects, researchers use different taxonomies to classify them on the basis of religion, geography, ethnicity, and prestige, to name but a few factors (Albirini, 2016). These taxonomies allow scholars to work within a clear frame of reference which is well adapted to the study and analysis of the dialects in hand, avoiding the dangers of both over-generalization or over-restriction to cases that cannot be objectively circumscribed. For a detailed discussion of these classifications, see Watson (2011).

For the purpose of this thesis, we follow the geolinguistic classification proposed by Habash (2010, p. 2). Arabic dialectal varieties are generally classified into seven groups:

- Egyptian Arabic (EGY) covers the dialects of the Nile valley: Egypt and Sudan.
- Levantine Arabic (LEV) includes the dialects of Lebanon, Syria, Jordan, Palestine, and Israel.

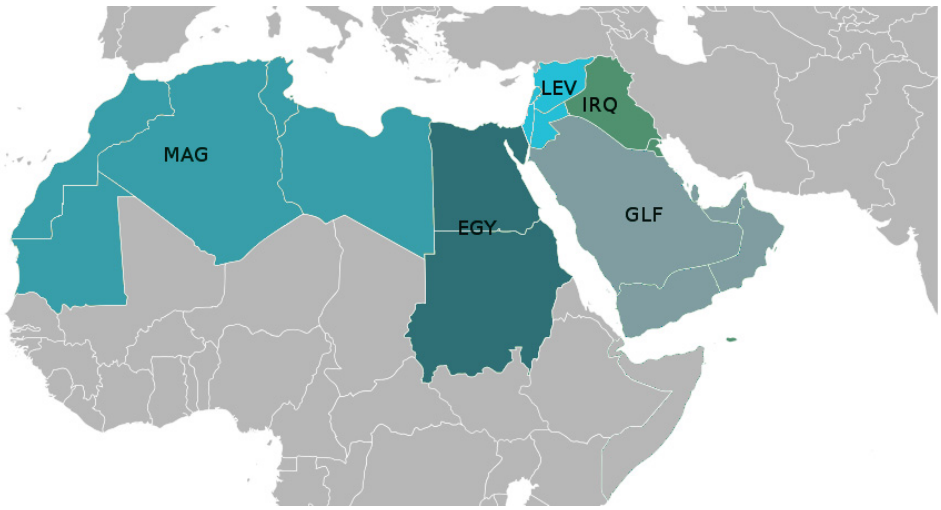

Figure 2.1: Geolinguistic classification of the Arabic dialects

- Gulf Arabic (GLF) includes the dialects of Kuwait, The United Arab Emirates, Bahrain, and Qatar. Saudi Arabia is typically included although there is a wide range of sub-dialects within it. Omani Arabic is included some times.Yemenite Arabic is often considered its own class.

- North African Maghrebi Arabic (MGR) covers the dialects of Morocco, Algeria, Tunisia and Mauritania. Libyan Arabic is sometimes included.

- Iraqi Arabic (IRQ) has elements of both Levantine and Gulf.

Note that this DA classification is rather broad and further division of the dialects groups is possible. Often Moroccan, Algerian, Tunisian, and Libyan are grouped together as Maghrebi Arabic, even though they are not all mutually intelligible.

Socially, the relation between DA and MSA is rather complex. It represents a prototypical example of diglossia (Ferguson, 1959), a situation whereby two distinct language varieties co-exist with a clear functional separation (Wardhaugh and Fuller, 2014). Arabic speakers tend to use DA and MSA for different purposes as the situation demands in their day-to-day life. While

MSA, the so-called 'high' variety (H), is an established standard among educated Arabic speakers, DA, the 'low' variety (L), is used in everyday informal communication. However, the contemporary sociolinguistic situation in the Arab world challenges the rigid separation between these two codes based on the formality-informality of the context in which they are used (Sayahi, 2014). For a more detailed discussion of diglossia, see Albirini (2016) and Bassiouney (2009).

2.2 LINGUISTIC CHARACTERISTICS OF THE ARABIC DIALECTS

Arabic dialects not only share commonalities with MSA in morphology and syntax, but they also share commonalities among themselves. It seems that dialects share common built-in functionalities for generating words, some of which may have been inherited from CA, where some of these functionalities are absent or severely diminished in MSA.

2.2.1 *Morphology*

As in all Semitic languages, the standard linguistic interpretation of word structure in DA describes words as the combination of two abstract morphemes: a root and patterns (the latter are also referred to as Binyanim). This was given the first formal generative treatment by McCarthy (1981).
A root is a sequence of three consonants and the pattern is a template of vowels with slots into which the consonants of the root are inserted. This process of insertion is called interdigitation (Beesley and Karttunen, 2003), as shown in Table 2.1. Note that the R refers to the root radical.

The resulting so called "lemmas" then pass through a series of affixations (which express morpho-syntactic features) and clitic attachments (conjunc-

Root	k-t-b كتب			
POS	V	V	N	N
Pattern	$R_1R_2eR_3$	$R_1aR_2R_2eR_3$	$mR_1R_2eR_3$	$R_1R_2AR_3$
Lemma	kteb "write"	katteb "make write"	mkteb "desk"	ktAb "book"

Table 2.1: Root and Pattern Interdigitation

tions and prepositions, for example, are mostly joined to adjacent words in writing) until they finally appear as surface forms. Even at this level, a word can undergo further form adjustments which involve a number of phonological, morphological and, orthographic rules. Morphologically, Arabic dialects share many features; to list but a few:

- Dialects have eliminated case endings.
- Dialects introduce a progressive particle, e.g. بـيقول "b+yqwl" EGY, عمـيقول "Em+yqwl" LEV, كـيقول "k+yqwl" MGR, and د يقول "d+yqwl" (Iraqi) for "he says". This does not exist in MSA.
- Some dialects use a post-negation particle, e.g. مـيحبـش "m+yHb+$" (does not like) (EGY, LEV and MGR). This does not exist in MSA or GLF.
- Dialects have future particles that are different from MSA, such as ح "H" LEV, هـ "h" EGY, and غ "g" MGR. Like that of the MSA future particle س "s", that may have resulted from shortening the particle سوف "swf" and with the shortened version as a prefix, dialectal future articles may have arisen in a similar process, where the Levantine future particle "H" is a shortened version of the word راح "rAH" (he will) (Jarad, 2014a; Persson, 2008a).
- Dialects routinely exhibit word merging, particularly when two identical letters appear consecutively. In MSA, this is mostly restricted to the case of the preposition ل "l" (to) when followed by the determiner ال "Al" (the), where the "A" in the determiner is silent. This is far more

common in dialects as in يعمل لك "yEml lk" (he does for you) ⇒ يعملك "yEmlk".

- Dialects often change short vowels to long vowels or vice versa. This phenomenon infrequently appears in poetry, particularly classical Arabic poetry, but is quite common in dialects, as in the change of له "lh" (to him) to ليه "lyh".

- Most dialects have eliminated dual forms, except in cases such as عيني "Eyny" (my two eyes) and قرشين "qr$yn" (two piasters). Consequently, dual agreement markers on adjectives, relative pronouns, demonstrative adjectives, and verbs have largely disappeared. Likewise, the masculine nominative plural noun and verb suffix ون "wn" has been largely replaced with the accusative/genitive forms ين "yn" and وا "wA" respectively.

- Most dialects have a two-way gender distinction (masculine and feminine) as well as a two-way number distinction (singular and plural). Nouns have gender and adjectives inflect for gender (Watson, 2011).

2.2.2 *Syntax*

Arabic dialects collectively share many syntactic features with both MSA and CA that can be summarized as follows:

- For word order, both SVO and VSO are present in the dialects, the SVO order is however the most prevalent one (Brustad, 2000).

- The time and aspect system is maintained in most dialects.

- In most of the dialects, nominal sentences (equational sentences), are constructed without copula (Watson, 2011).

- Some of MSA's complex syntactic phenomena such as irrational plural[5] agreement are maintained in the dialects (Zitouni, 2014, p. 19).

For a detailed account of dialectal Arabic syntax, refer to Brustad (2000).

2.3 PROCESSING OF DIALECTAL ARABIC

Work on DA is fairly new when compared to that on MSA. Over the last decade, only very few efforts have targeted DA processing. Many Arabic dialects are still under-resourced language varieties, as the written production remains relatively very low in comparison to MSA.
Increasingly, however, DA is emerging as the language of informal communication on the web, in emails, micro-blogs, blogs, forums, chat rooms, and social media. This new situation has amplified an urgent need for consistent language resources and Natural language processing tools for DA. While certain dialects, particularly EGY and LEV, have received their share of attention by the NLP research community, most of the dialects remain under-resourced. This scarcity of dialect-specific linguistic resources and the prevalence of MSA content have motivated many researchers to explore the possibility of adapting MSA resources and tools to dialects.

MORPHOLOGY For morpholgy, Bakr, Shaalan, and Ziedan (2008) and, Salloum and Habash (2011) modified the Buckwalter Morphological Analyzer (Buckwalter, 2004) to accept DA pefixes and suffixes. Subsequently, many DA morphological analyzers were introduced that also rely on MSA resources (Almeman and Lee, 2012; Salloum and Habash, 2012, 2014; Zribi, Khemakhem, and Belguith, 2013). Most of these approaches are aimed at converting DA text to some MSA-like form; as such they do not model DA

5 It refers to non human being. The plural of irrational nouns is treated as feminine singular.

linguistic phenomena.

On the other hand, many researchers are interested in modeling DA. They introduced many dialectal corpora such as the CALLHOME Egyptian Arabic Transcripts (LDC97T19), which was made available for research already in 1997. Newly developed resources include the corpus developed by Bouamor, Habash, and Oflazer (2014), which contains 2,000 parallel sentences in multiple dialects and MSA as well as English translations. Although this approach is more linguistically informed, building resources from scratch is very expensive in terms of time and effort. Furthermore, these attempts are lacking in coverage in one dimension or another (Habash, Eskander, and Hawwari, 2012).

LANGUAGE ID Language identification is the task of identifying the language a given document is written in. The ubiquity of the Internet access has created an unprecedented deluge of information and textual data on the world wide web. Given this information, the need for automatic means of determining the language web documents has become very crucial.

Several projects have concentrated on the identification of DA. For example, Elfardy, Al-Badrashiny, and Diab (2013) present a system for the identification of Egyptian Arabic which selects a tag based on the sequence with a maximum marginal probability, considering 5-grams. A later version of the system is named AIDA2 (Al-Badrashiny, Elfardy, and Diab, 2015) and it is a more complex hybrid system that incorporates different classifiers and components such as language models, a named entity recognizer, and a morphological analyzer. The classification strategy is built as a cascade voting system, whereby a Conditional Random Field (CRF) classifier tags each word based on the decisions from four other underlying classifiers.Zaidan and Callison-Burch (2014) introduced a system that discriminates between

Arabic dialects which can be MSA, Maghrebi, Egyptian, Levantine, Iraqi, or Gulf.

The majority of the systems dealing with word-level language identification rely on linguistic resources (such as named entity gazetteers and word lists) and linguistic information (such as POS tags and morphological analysis), and they use Machine Learning (ML) methods that have been typically used with sequence-labeling problems, such as Support Vector Machine (SVM), CRF and n-gram language models. A few, however, have recently turned to recurrent neural networks (RNN) and word embedding with remarkable success (Samih et al., 2016).

2.4 CHALLENGES OF PROCESSING DIALECTAL ARABIC

DA shares many challenges with MSA, as DA inherits the same characteristics of being a Semitic language with a complex templatic derivational morphology. As in MSA, most nouns and verbs in Arabic dialects are typically derived from a determined set of roots by applying templates to the roots to generate stems. Such templates may carry information that indicates morphological features of words such as POS tag, gender, and number. Furthermore, stems may accept prefixes and/or suffixes to form words as in a highly inflected language. Prefixes include coordinating conjunctions, determiners, particles, and prepositions, and suffixes include attached pronouns and gender and number markers. This results in a large number of words (or surface forms) and in turn a high-level of sparseness and increased number of new words during processing.

In addition to the shared challenges, DA has its own peculiarities, which can be summarized as follows:

- Lack of standard orthography. Many of the words in DA do not follow a standard orthographic system (Habash, Diab, and Rambow, 2012).

- Many words do not overlap with MSA as a result of language borrowing from other languages (Ibrahim, 2006), such as كافيه kAfiyh "cafe" and تاتو tAtuw "tattoo", or coinage, such as the negative particles مش mi$ "not" and بلاش balA$ "do not".

- Codeswitching is also very common in Arabic dialects (Samih et al., 2016).

- Multiple words are merged by concatenating and dropping letters as the word مبيجلهاش mbyjlhA$ (he did not go to her), which is a concatenation of "mA byjy lhA$".

- Some affixes are altered in form in comparision with their MSA counterparts, such as the feminine second person pronoun ك k → كي ky and the second person plural pronoun تم tm → تو tw.

- DA exhibits some morphological patterns that do not exist in MSA, such as the passive pattern AitofaEal, for instance اتكسر Aitokasar "it broke".

- In DA, new particles are introduced, such as the progressive ب b meaning 'is doing' and the post negative suffix ش $, which behaves like the French "ne-pas" negation construct.

- Letter substitution and consonant mutation are also common in DA. For example, in dialectal Egyptian, the interdental sound of the letter ث v is often substituted by either ت t or س s as in كثير kvyr "much" → كتير ktyr and the glottal stop is reduced to a glide, such as جائز jA}iz "possible" → جايز jAyiz. Such features have been extensively studied in phonology as *lenition*, *softening* of a consonant, or *fortition*, hardening of a consonant.

- Vowel elongation is also present in DA, such as راجل rAjil "man" from رجل rajul. likewise vowel shortening is also very common, such as ديما dayomA "always" from دايما dAyomA.

- DA uses masculine plural or singular noun forms instead of dual and feminine plural by dropping some articles and prepositions in some syntactic constructs. For example, DA uses only one form of noun and verb suffixes such as ين yn instead of ون wn and وا wA instead of ون wn respectively.

- In addition, there are the regular discourse features in informal texts, such as the use of emoticons and character repetition for emphasis, e.g. ادعووووووولى AdEwwwwwwwliy "pray for me".

2.5 CONCLUDING REMARKS

This chapter has given a brief overview of some important linguistic characteristics of the Arabic language and its dialects. It aims at providing sufficient detail that would highlight the challenges involved in computational processing of DA.

3

DEEP LEARNING BACKGROUND

DL is now increasingly emerging as a new research area in ML and NLP. With the advent of social media and big data, the need for efficient learning methods to process these huge amounts of textual information has drastically increased. In recent years advances in DL techniques have been shown to hold great promise as the answer to many challenges in NLP. This chapter aims to provide a brief overview of existing deep learning algorithms and will attempt to relate them to the context of NLP, in particular dialectal Arabic processing. The first section discusses the basic idea of the shallow structures and deep neural networks in brief. In the second section the most prominent deep learning primitives will be described in some detail in their simplest form. These primitives or building blocks are at the foundation of many deep learning methods and understanding their basic form will allow the reader to quickly understand more complex models relying on these building blocks. The last section provides a brief description of related work.

3.1 INTRODUCTION

Over the last several years, most natural language processing techniques have extensively relied on linear models, examplified by a number of linear classifiers such as SVM (Cortes and Vapnik, 1995), CRF (Lafferty, McCallum, and Pereira, 2001), Logistic Regression, Hidden Markov Models (HMM), and Perceptron (Rosenblatt, 1958), to name but a few. These models have

been effective in solving many supervised classification problems and have demonstrated great successes in a broad range of NLP tasks, including language identification, language modelling, chunking, Part of Speech Tagging, morphological analysis, and many more. Yet despite many successful applications, these methods have proved unfit for the more challenging tasks of artificial intelligence (AI) that require a higher-level of abstraction for a number of reasons:

- Their simple shallow structures restrict their modeling and representational power since they cannot take advantage of some valuable information for real world problems with rich structure.
- They often fail to learn or discover patterns such as non-linear relations among input values and features (Bengio, Delalleau, and Roux, 2006; Bengio et al., 2009).
- They usually rely on hand-crafted features which are task specific, incomplete, and time-consuming to construct.
- Another challenge in classification is the data non-linearity that characterizes the feature overlap of different classes, making the task of separating the classes more difficult.
- In these models, feature extraction and classifier training are kept separated and are not jointly optimized to maximize the overall performance of the system.

These limitations are partially responsible for steering away most of machine learning and natural language processing research from linear models in favor of non-linear models and much deeper structures such as DNNs with more representational power.

DL is a branch of machine learning and and it is just a re-branded name to

refer to DNNs (Collobert et al., 2011; Goldberg and Hirst, 2017), which are extensions to the traditional shallow Artificial Neural Networks (ANNs). The latter are loosely based on the concept of mimicking the activities in the human brain. They consist of many layers of information processing units, arranged in a hierarchical architecture to perform pattern classification and automatic feature learning (Deng, 2014).

DNNs have been around for a long time. As universal non-linear function approximators (Hornik, Stinchcombe, and White, 1989), DNNs theoretically offer a principled approach to modelling complex multi-dimensional non-linear problems, but in practice they are difficult to train in an efficient manner (Bengio et al., 2009; Glorot and Bengio, 2010). Through the 1990s and 2000s, deep architectures languished because their performance consistently lagged behind the widely used shallow architectures relying on feature engineering. However this situation has changed with the work of Hinton and Salakhutdinov (2006). They introduced the so-called Deep Belief Networks (DBNs), which are composed of simple learning units, particularly Restricted Boltzmann Machines (RBMs). These latter are pre-trained in a greedy layer-wise fashion, exploiting an unsupervised learning algorithm to initialize the weights of the DBNs. This makes them less prone to get stuck in local minima during backpropagation[1]. Since 2006, DNNs have re-emerged as an exciting research area, attracting a wide variety of scientists and engineers and have been successfully applied to a variety of NLP tasks.

1 A supervised learning algorithm that calculates the errors and then back propagates them to previous layers. These previous layers then adjust their weights and biases accordingly, and the configuration is thus changed. The entire training process is a means of finding the right combination of weights and biases for which the multi-layer neural network performs at its best (Shukla, Tiwari, and Kala, 2010).

3.2 BASICS OF NEURAL NETWORKS

3.2.1 *Mathematical Notation*

Following the notation of Goldberg (2016), throughout this dissertation, bold upper case letters are used to represent matrices ($\mathbf{X}, \mathbf{Y}, \mathbf{Z}$), and bold lower-case letters to represent row vectors ($\mathbf{b}$).

3.2.2 *The Artificial Neuron*

The Artificial Neuron refers to a class of biologically-inspired machine learning techniques evolved from the idea of mimicking the activity of the human brain. Figure 3.1 shows an example of a single neuron (Haykin, 1994), a very simple processing unit.

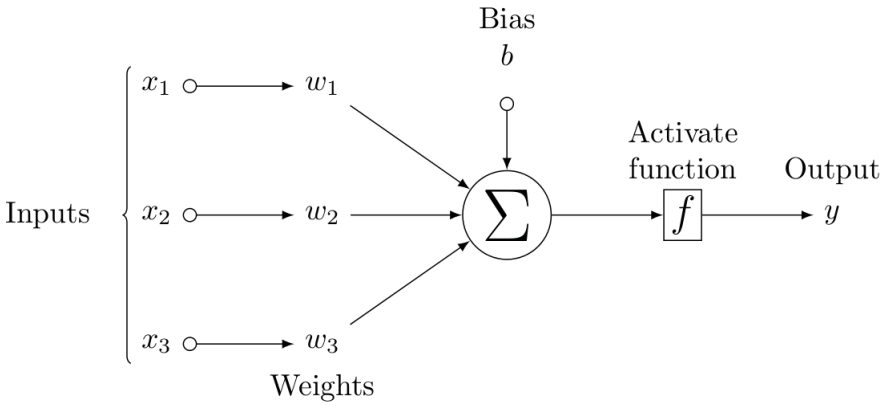

Figure 3.1: Schematic diagram of the artificial neuron

It consists of a number of inputs, synaptic weights, an activation function, and an output. Let $\mathbf{x} = (x_1, x_2, ..., x_n)$ denote an input vector, and $\boldsymbol{w} = (w_1, w_2, ..., w_n)$ be the vector of weights assigned to the links from

input to output nodes; b denotes a bias, the output y is computed by the following function:

$$y = f(\sum_{i=1}^{n} x_i w_i + b) \tag{3.1}$$

where f is a non-linear activation function that can be a simple threshold, sigmoidal, hyperbolic tangent, or radial basis function. The most commonly used activation function is the sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

or the hyperbolic tangent function:

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{3.3}$$

The *sigmoid* is a continuous squashing function which takes any real-valued number and outputs a number in the range between 0 and 1. The *tanh* produces a centered output between -1 and 1. The purpose of these activation functions is to introduce non-linearity to the network and ensure that the representation in the input space is mapped to a different space in the output. Figures 3.2 and 3.3 present a graphical plot of these two functions.

In the literature for DNNs, various other recent non-linearities have been introduced with desirable application-specific properties, for example *ReLu* (Krizhevsky, Sutskever, and Hinton, 2012) and *A leaky ReLU*(Maas, Hannun, and Ng, 2013), to name but a few.

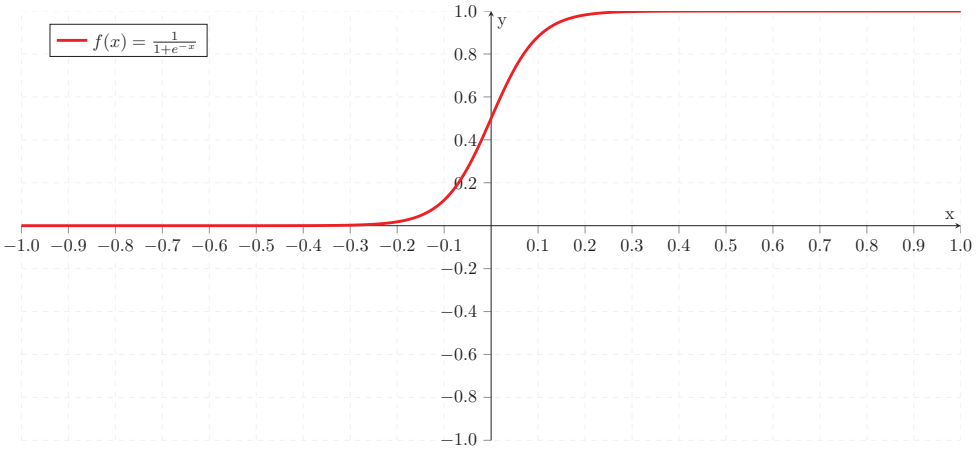

Figure 3.2: Sigmoid function

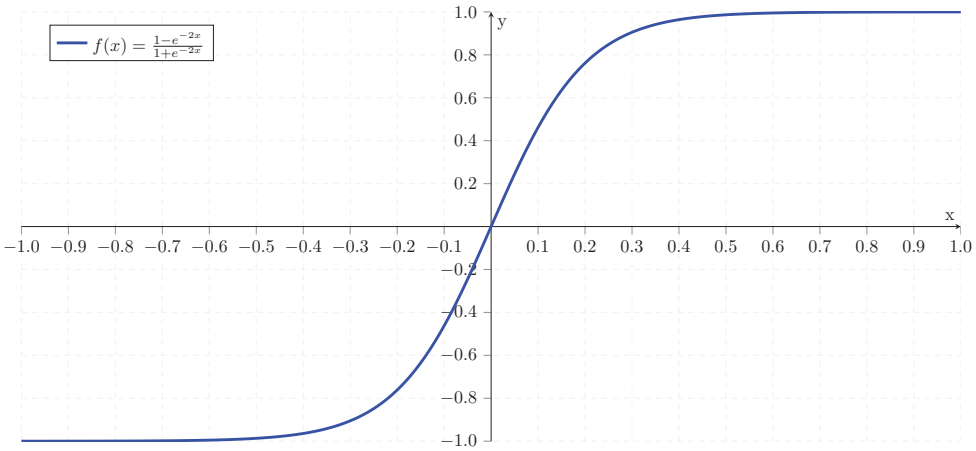

Figure 3.3: Hyperbolic tangent function

3.2.3 *Multi-Layer Neural Networks*

Neurons can be arranged in different layers, with connections feeding forward from one layer to the next to create a Multi-Layer Network (Rumelhart, Hintont, and Williams, 1986). Figure 3.4 shows the general architecture of a deep Multi-Layer Neural Network. The first and the last layers are the input and output layers respectively. On the other hand, hidden layers are layers that are neither input nor output.Regarded computationally, process-

ing information in a network with fully connected layers is simply a series of matrix-vector multiplications and element-wise activation functions. Formally, let $\mathbf{x}$ denote the input vector, let $\mathbf{W}_1$ be the matrix of weights $w_{i,j}$ connecting input neuron j with hidden neuron i, let $\mathbf{b}_1$ to be a vector with the biases of the neurons in the first hidden layer and σ_1 to be an activation function. Then the output vector $\mathbf{h}_1$ from the first hidden layer can be computed as:

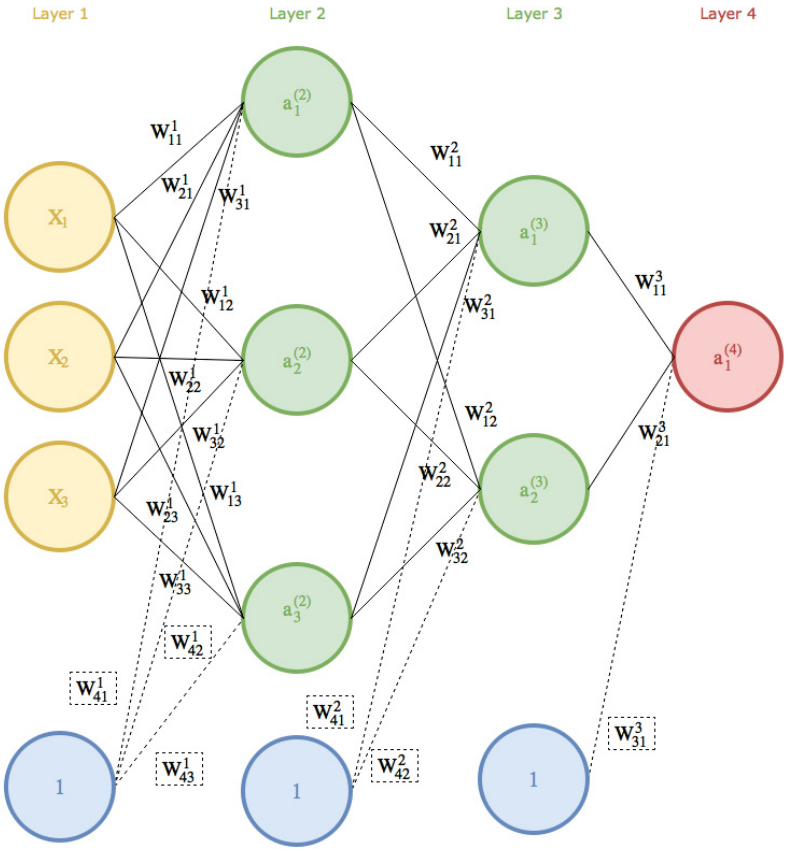

Figure 3.4: Schematic diagram of a Multi-Layer Neural Network

$$\mathbf{a}_1 = \sigma_1(\mathbf{W}_1^T\mathbf{x} + \mathbf{b}_1) \tag{3.4}$$

The output vector $\mathbf{a}_1$ is then inputted to the second hidden layer of the network. In a similar way, the output from layer n is also computed as:

$$\mathbf{a}_n = \sigma_n(\mathbf{W}_n^T \mathbf{a}_{n-1} + \mathbf{b}_n) \tag{3.5}$$

The training of a multi-layer neural network is performed by the Backpropagation Algorithm (BPA) (Rumelhart, Hintont, and Williams, 1986), which is a supervised learning algorithm that calculates the errors and then back-propagates them to previous layers. These previous layers then adjust their weights and biases accordingly, and the configuration is thus changed. The entire training process is a means of finding the right combination of weights and biases with which the multi-layer neural network performs at its best (Shukla, Tiwari, and Kala, 2010).

3.3 RECURRENT NEURAL NETWORKS

The Recurrent Neural Network (RNN) belongs to a family of neural networks suited for modeling sequential data. Figure 3.5 shows a schematic plot of an RNN. Given an input vector $\mathbf{x} = (x_1, ..., x_n)$, an RNN calculates the output vector $\mathbf{y}_t$ of each word x_t by iterating the following equations from $t = 1$ to n:

$$\mathbf{h}_t = f(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{3.6}$$

$$\mathbf{y}_t = \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y \tag{3.7}$$

where $\mathbf{h}_t$ is the hidden states vector, $\mathbf{W}$ denotes weight matrix, $\mathbf{b}$ denotes bias vector, and f is the activation function of the hidden layer.

Theoretically RNNs can learn long distance dependencies, yet in practice they fail due the vanishing or exploding gradient (Bengio, Simard, and Frasconi, 1994). The most effective solution to this problem so far is the Long short-term memory (LSTM) architecture (Hochreiter and Schmidhuber, 1997). An LSTM network introduces the memory cell, a unit of computation that

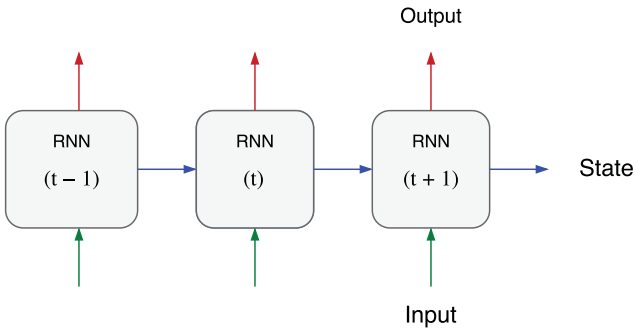

Figure 3.5: Schematic diagram of a reccurent Neural Network

replaces traditional nodes in the hidden layer of a network. With these memory cells, the networks are able to overcome difficulties with training encountered by earlier recurrent networks. The output of LSTM hidden layer $\mathbf{h}_t$ given input $\mathbf{x}_t$ is computed via the following intermediate calculations: (Graves, 2013):

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \tag{3.8}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f) \tag{3.9}$$

$$\mathbf{c}_t = \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \tag{3.10}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o) \tag{3.11}$$

$$\mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{c}_t) \tag{3.12}$$

where σ is the logistic sigmoid function, and $\mathbf{i}$, $\mathbf{f}$, $\mathbf{o}$, and $\mathbf{c}$ are respectively the input gate, forget gate, output gate, and cell activation vectors. More interpretation about this architecture can be found in (Lipton et al., 2015). Figure. 3.6 illustrates a single LSTM memory cell (Graves and Schmidhuber, 2005).

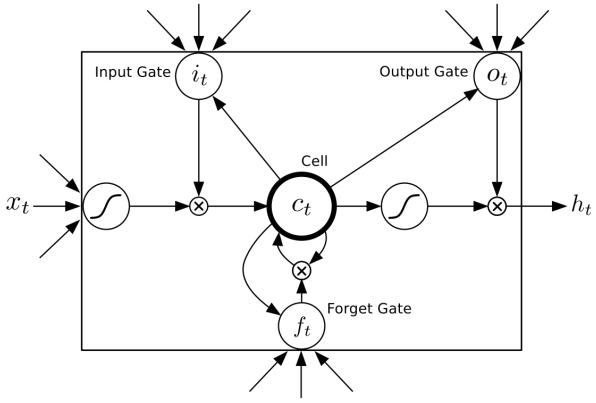

Figure 3.6: A Long Short-Term Memory Cell.

One shortcoming of conventional LSTMs is that they are only able to make use of previous context. In many NLP tasks there is no reason not to exploit future context as well. BiLSTM networks (Schuster and Paliwal, 1997) are extensions to the single LSTM networks. They are capable of learning long-term dependencies and maintain contextual features from the past and future states. As shown in Figure 3.7, they comprise two separate hidden layers that feed forward to the same output layer. A BiLSTM calculates the forward hidden sequence $\overrightarrow{\mathbf{h}}$, the backward hidden sequence $\overleftarrow{\mathbf{h}}$ and the output sequence $\mathbf{y}$ by iterating over the following equations:

$$\overrightarrow{\mathbf{h}_t} = \sigma(\mathbf{W}_{x\overrightarrow{h}}\mathbf{x}_t + \mathbf{W}_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{\mathbf{h}}_{t-1} + \mathbf{b}_{\overrightarrow{h}}) \tag{3.13}$$

$$\overleftarrow{\mathbf{h}_t} = \sigma(\mathbf{W}_{x\overleftarrow{h}}\mathbf{x}_t + \mathbf{W}_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{\mathbf{h}}_{t-1} + \mathbf{b}_{\overleftarrow{h}}) \tag{3.14}$$

$$\mathbf{y}_t = \mathbf{W}_{\overrightarrow{h}y}\overrightarrow{\mathbf{h}_t} + \mathbf{W}_{\overleftarrow{h}y}\overleftarrow{\mathbf{h}_t} + \mathbf{b}_y \tag{3.15}$$

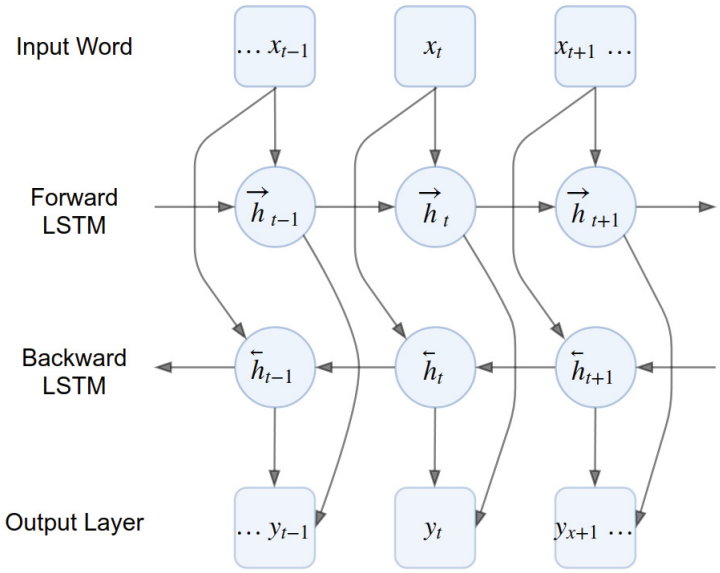

Figure 3.7: A bidirectional LSTM.

More interpretations about these formulas are found in Lipton et al. (2015).

3.4 DEEP NEURAL NETWORKS FOR NLP

In recent years, advances in deep learning have revolutionized the applicability of machine learning in a large number of natural a language problems. The advantage of deep learning with respect to the rest of machine learning methods is that practically all aspects of the model are directly learned from the data. This allows any system to learn complex functions mapping the input to the output directly from data, without depending completely on human-crafted features. (Collobert et al., 2011) employed a deep recurrent convolutional neural networks to jointly solve many NLP tasks from scratch. Their system did not rely on any task-specific feature engineering or external resources to achieve competitive results.

Another major component in deep neural networks for language is the use

of word embedding. Due to their simplicity and efficacy, pre-trained word embedding have become ubiquitous in NLP systems. Many prior studies have shown that they capture useful semantic and syntactic information (Mikolov et al., 2013) and including them in NLP systems has been shown to be enormously helpful for a variety of downstream tasks.

3.5 CONCLUDING REMARKS

This chapter has presented a brief introduction to deep learning. The literature on DL is vast, mostly coming from the ML community. Until recently, The NLP community has embraced deep learning and momentum is growing fast. DNNs provide an analytical alternative to linear models which are often limited by strict assumptions of normality, linearity, and variable independence. Since DNNs can capture many kinds of relationships they allow the user to quickly and relatively easily model phenomena which otherwise may have been very difficult or impossible to explain otherwise.

In the next chapter we will provide a description of the data we used to train our neural models.

Part III

RESOURCE CREATION FOR DIALCTAL ARABIC

4

DATA AND RESOURCE CREATION

This chapter includes work that was published as Samih and Maier (2016a) and Samih and Maier (2016b). I was primarily responsible for the design and creation of the corpus described in these two papers. I was also an active collaborator throughout the creation of the multi-dialectal segmentation corpus presented in the following three papers, Samih et al. (2017a), Samih et al. (2017b) and, Eldesouki et al. (2017).

4.1 INTRODUCTION

In recent years, the advent of the era of social media and the ubiquity of the internet access have created an unprecedented deluge of information and textual data on the world wide web. This data brings in its wake new opportunities and poses many challenges for machine learning and Natural Language Processing (NLP) in particular. The sheer size, non-standard spelling, the poor quality, the informality, and the noise of this data, presents new challenges to standard NLP tools developed for traditional data (Farzindar and Inkpen, 2015). To make sense out of such data and exploit its value, novel NLP methods, resources, and efficient algorithms beyond rule-based deductive reasoning and "traditional" system engineering need to be created.

A case in point is DA. Until recently, DA was regarded as a partially under-resourced language since the written production remains relatively low in comparison to MSA. Increasingly, however, DA is emerging as the language

of informal communication on the web, in emails, social media platforms, micro-blogs, blogs, forums, chat rooms, etc. This new situation amplifies the need for consistent language resources and NLP systems for Arabic dialects. While MSA has already received attention in NLP research, most Arabic dialects remain particularly under-resourced (Habash, 2010). They are strongly embedded in a multilingual context that entails frequent code-switching, i.e., switching between languages within the same context (Bullock and Toribio, 2009). Building linguistic resources and creating the necessary tools for processing DA is a priority, not least because its grammars, and vocabularies are particularly distant to MSA (Diab et al., 2010a).

In this chapter, we try to bridge this critical gap of resources by creating two adequate DA datasets with annotations on the token level. These two datasets will foster DA NLP research as they will provide a test bed for adaptive learning algorithms, lead to significant robustness in handling very diverse data sources, and create a framework for genuine multilingual processing. In Section 4.2 we present the *Moroccan Arabic Darija*[1] *code-switching corpus*. It is collected from Moroccan social media sources, namely blogs and internet discussion forums, and has a size of 223k tokens. To our knowledge, it is currently the largest resource of its kind. In Section 4.3 we describe our effort to compile the *Twitter multi-dialectal Arabic segmentation corpus*[2]. In this corpus each dataset consists of a set of 350 manually segmented tweets.

1 Darija is the Moroccan Dialectal Arabic.
2 This dataset is available here: http://alt.qcri.org/resources/da_resources/releases/current/seg_data_jun122017.tgz

4.2 THE MOROCCAN CODE-SWITCHING CORPUS

4.2.1 *Data Collection*

We acquire our data from internet discussion forums and blogs which are hosted in Morocco or extensively used by Moroccans[3]. The crawled output is stripped from HTML tags and other meta-data. Since sentence splitting is not a trivial task in Arabic and no such tool is available for Darija, we leave the downloaded text units ("posts") intact. Then we tokenize the text with a simple heuristic, delete all diacritic marks as usual in Arabic NLP (Habash, 2010), and transliterate the text using the Buckwalter transliteration system, as presented in Appendix C. Finally, we store the data token-wise as pairs (original and transliteration) in a MySQL database. The size of the resulting corpus is 15 million tokens in total. It covers a wide range of topics including politics, religion, sport, and economics.

Existing code-switched data sets are often highly skewed towards one language (Solorio et al., 2014a), with a high percentage of the sentences not exhibiting code-switching at all. In our corpus, MSA is more prevalent than Darija. In order to obtain a resource that concentrates on code-switching, we aim at minimizing the skewing by extracting only a subset of the data set that contains more instances of code-switching. The subset is extracted with the following iterative process. We first compile an initial seed list of 439 commonly used Darija words and phrases collected from the Internet[4]. Then we repeat the following steps. Each word and phrase in the seed list is formulated in a MySQL query as a keyword to retrieve more code-switched examples from the original data set. The retrieved examples are put into the code-switched data set. Then, the seed list is updated with all words of the

3 http://www.hibapress.com/ and https://www.goud.ma/
4 en.mo3jam.com/dialect/Moroccon

retrieved text units, and the procedure is repeated until the code-switched data set has reached a certain size. Note that the extraction procedure does not guarantee that only code-switched examples are included in the final data set. However, we do achieve a rate of 73.9% of text units with code-switching (see below), which contrasts with code-switching ratios of around 20% in data sets in previous literature (Solorio et al., 2014a). Subsection 4.2.2 shows more detailed statistics of the data.

4.2.2 *Annotation*

We adapt the annotation guidelines for the data used in the shared task on code-switching detection at EMNLP 2014 (Solorio et al., 2014a) and use all of their labels:

- `lang1` is used for MSA words,
- `lang2` for Darija words,
- `mixed` is used to mark words that mix a Darija stem with MSA morphology or vice versa,
- `ne` is used to mark named entities, including dates,
- `other` is used to mark other numbers, punctuation, and other non-language material, and
- `ambiguous` is used for material which could be interpreted as either `lang1` or `lang2`.

We additionally introduce a new label `lang3`, which accounts for the special linguistic situation in Morocco. It marks words belonging to a language which is neither MSA nor Darija, notably either French, English, Spanish, or

Berber. Note that French words may be written with Arabic script. Berber is furthermore written with its own script, called Tifinagh[5]. Since `lang3` material is scarce, we have decided against the introduction of further language labels.

Several DA annotation tools have been reported in the literature, such as COLANN_GUI and COLABA (Benajiba and Diab, 2010; Diab et al., 2010b), DATOOL (Tratz et al., 2013), DIWAN (Al-Shargi and Rambow, 2015), among others, and have been successful with different annotation tasks, yet they were either not available or did not suit our needs exactly. Therefore, we have built a custom web-based annotation tool. The annotation has been performed by three Moroccan Darija native speakers, two of them with no prior linguistic knowledge. They worked independently at different physical locations; crowd-sourcing services like Amazon Mechanical Turk[6] have not been used. In our annotation tool, a single text unit is shown at a time. The words can be annotated in random order. When the annotation of a text unit is done, it is saved by a single click back to the database, whereby the label is stored for each token together with its Arabic script version and its Buckwalter transliterated version.

To ensure agreement among the annotators, various training sessions were provided and regular inter-annotator agreement measures were performed to check the annotation quality. The final inter-annotator agreement (Cohen's κ) was computed on a selection of 50 text units between 0.82 and 0.86.

In total, 3,862 text units with around 223k tokens were annotated. See Table 4.1 the number of tokens for each label, as well as the respective ratio among all labels, and average as well as median length of tokens per label. We found the average number of switches between languages per text unit to

5 The script used to write the Berber languages in North African https://en.wikipedia.org/wiki/Tifinagh

6 https://www.mturk.com/mturk/welcome

be 2.6, and the median to be 2. Out of the 3,862 text units, 284 (7.4%) contain only MSA tokens, and 725 (18,8%) contain only Darija tokens. In other words, 2,853 (73,9%) text units are true code-switched instances. In 1,950 (50.5%) text units, more than 50% of all language tokens (i.e., not `mixed` tokens, etc.) are `lang1`; the same holds for `lang2` in 1,970 (51.0%) text units.

As mentioned above, we have not performed sentence splitting, i.e., our text units can consist of more than one sentence. Fig. 4.1 shows a histogram revealing the distribution of lengths of text units. We see that it is positively skewed; most text units have 50 or fewer tokens. Nevertheless, our data does contain very long text units with over 300 tokens.

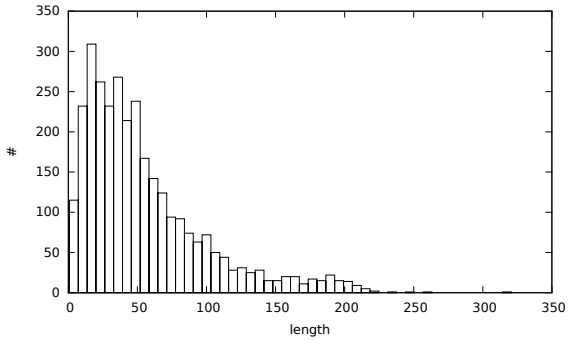

Figure 4.1: Text unit length histogram

			length	
labels	**all**	**%**	**avg.**	**med.**
`lang1`	109,025	48.82	6.13	4
`lang2`	76,732	34.36	5.22	4
`lang3`	1,383	0.62	1.88	1
`ne`	17,087	7.65	1.29	1
`mixed`	86	0.04	1.00	1
`ambiguous`	141	0.06	1.26	1
`other`	18,830	8.43	1.02	1

Table 4.1: Data statistics

As an example for the annotation, Fig. 4.2 shows a sample forum post with free translation, and the annotated version of the text. Aside from MSA words, this text unit contains Darija words and words with mixed morphology. Darija words are, e.g., رآه (demonstrative pronoun), كرف (meaning worse). Words with mixed morphology are, e.g., غاتمشي and غادجي, which both exhibit the Darija future prefix غا on words which otherwise are MSA. Last, note that the punctuation in the text unit is labeled as `other`. For the purpose of presentation, i.e., in order to not have to mix writing directions, the annotated tokens are shown only in the Buckwalter transliterated form; labels are separated from the tokens they annotate by a single forward slash.

رآه غاتمشي هاذ الحكومة و غادجي حكومة اخرى و لكن كرف من هذي لأن الحكومة الوحيدة اللي بغات تصلح لبلاد هي هذي منذ الاستقلال و ها حنا غانشوفو الشفارة اللي غاي يرجعو (الاتحاد الدستوري ، الاستقلال ، الأحرار و لقتالة)

rAh/lang2 gAtm$y/lang2 hA*/lang2 AlHkwmt/lang2 w/lang2 gAdjy/lang2 Hkwmt/lang2 AxrY/lang2 w/lang2 lkn/lang2 krf/lang2 mn/lang2 h*y/lang2 l>n/lang1 AlHkwmt/lang1 AlwHydt/lang1 Ally/lang2 bgAt/lang2 tSlH/lang2 lblAd/lang2 hy/lang2 h*y/lang2 mn*/lang1 AlAstqlAl/lang1 w/lang2 hA/lang2 HnA/lang2 gAn$wfw/lang2 Al$fArt/lang2 Ally/lang2 gAy/lang2 yrjEw/lang2 (/other AlAtHAd/ne Aldstwry/ne ,/other AlAstqlAl/ne ,/other Al>HrAr/ne w/lang2 lqtAlt/lang2)/other ./other

This government will go, and another one will come which will be worse. The current government is the only government since independence which has worked hard to develop the country; and we will see the thieves coming back (Independence Party, Constitutional Party, Liberal Party, and murderers).

Figure 4.2: Sample forum post with annotated version and free translation

4.2.3 *Related Work*

Research in processing on code-switching and of language varieties and dialects has recently attracted increased attention. This is reflected by recent workshops (Diab et al., 2014; Nakov, Osenova, and Vertan, 2014; Solorio et al., 2014a; Zampieri et al., 2014). A number of language resources has been created, such as the ones described by Tratz et al. (2013), Maharjan et al. (2015a), Dey and Fung (2014), and the data set from the Shared Task at the First Workshop on Computational Approaches to Code-Switching at EMNLP 2014 (Solorio et al., 2014a), with a particular focus on inter-operable annotation guidelines. A popular use for those resources can be found in approaches to automatic detection of code-switching points in text. This task has mostly been treated as a sequence labeling problem. Different techniques have been applied, ranging from Naive Bayes (Solorio and Liu, 2008a) over Conditional Random Fields (Elfardy, Al-Badrashiny, and Diab, 2014a; King and Abney, 2013), Support Vector Machines (Bar and Dershowitz, 2014), Markov Models (King et al., 2014) and n-gram based approaches (Bacatan et al., 2014; Shrestha, 2014) to Recurrent Neural Networks (Chang and Lin, 2014a). POS tagging of code-switched text has also been investigated (Rodrigues and Kübler, 2013; Solorio and Liu, 2008b).

Concerning the processing of Arabic in general, there is an ample body of research. For an overview, see Habash (2010). With regard to the processing of Dialectal Arabic, most of the existing work concentrates on Levantine and Egyptian Arabic (see, e.g., Elfardy and Diab (2012b) and Elfardy, Al-Badrashiny, and Diab (2014a)). An exception is Cotterell et al. (2014), who works on Algerian Arabic. In linguistics, Moroccan Arabic found attention very early (Harrell, 1962; Harrell and Sobelman, 1966). Work on the computational processing of Darija, however, remains very scarce. To our knowledge,

there is only the work of Tratz et al. (2013), who present a data collection and annotation environment for romanized Darija, and the work of Voss et al. (2014) who present an approach for finding romanized Darija in code-mixed tweets.

4.3 THE MULTI-DIALECTAL SEGMENTATION CORPUS

4.3.1 *Data Creation*

We constructed our dataset by obtaining 350 tweets that were authored for each of the following four dialects: Egyptian, Levantine, Gulf, and Maghrebi. For dialectal Egyptian tweets, we obtained the dataset described in (Darwish, Sajjad, and Mubarak, 2014), and we used the same methodology to construct the dataset for the remaining dialects. Initially, we obtained 175 million Arabic tweets by querying the Twitter API[7] using the query "lang:ar" in March 2014. Then, we identified tweets whose authors identified their location in countries where the dialects of interest are spoken (ex. Morocco, Algeria, and Tunisia for Maghrebi) using a large location gazetteer (Mubarak and Darwish, 2014). Then we filtered the tweets using a list containing 10 strong dialectal words per dialect, such as the Maghrebi word كيما "kymA" (like/as in) and the Leventine word هيك "hyk" (like this). Given the filtered tweets, we randomly selected 2,000 unique tweets for each dialect and asked a native speaker of each dialect to manually select 350 tweets that were heavily dialectal. Table 4.2 lists the number of tweets that we obtained for each dialect and the number of words they contain.

Segmentation of DA can be applied on the original raw text or on the cleaned text after correcting spelling mistakes and applying conventional or-

7 https://developer.twitter.com/en/docs/api-reference-index

Dialect	No of Tweets	No of Tokens
Egyptian	350	6,721
Levantine	350	6,648
Gulf	350	6,844
Maghrebi	350	5,495

Table 4.2: Dataset size for the different dialects

Field	Annotation
Orig. word	بيقولك "byqwlk"
Meaning	he is saying to you
In situ Segm.	ب+يقول+ك "b+yqwl+k"
CODA	بيقول لك "byqwl lk"
CODA Segm.	ب+يقول ل+ك "b+yqwl l+k"

Table 4.3: Egyptian annotation example

thography rules, such as CODA[8] (Habash, Diab, and Rambow, 2012). In this work, we decided to segment the original raw text. Though Egyptian CODA is a reasonably stable standard, Conventional Orthography for Dialectal Arabics (CODAs) for other dialects are either immature or nonexistent. Building such tools requires the establishment of clear guidelines, is laborious, and may require large annotated corpora (Eskander et al., 2013), such as the LDC Egyptian Treebank.

4.3.2 *Annotation*

To prepare the "ground truth" data for a dialect, we enlisted an annotator who was either a native speaker for the dialect or well versed in it and had background in natural language processing. The authors along with another native speaker of the dialect made multiple review rounds on the work of

8 A Conventional Orthography for Dialectal Arabic

the annotator to ensure consistency and quality. The annotation guidelines[9] were fairly straightforward. Basically, we asked annotators to:

- segment words in a way that would maintain the correct number of part of speech tags
- favor stems when repeated letters are dropped as Table 4.3
- segment multiple concatenated words with pluses as in the "merged words" example in Table 4.4.
- attach injected long vowels that trail prepositions or pronouns to the preposition or pronoun respectively (ex. ليكي "lyky" (to you – feminine) → "ly+ky")
- treat dialectal words that originated as multiple fused words as single tokens (ex. علاش "ElA$" (why) – originally على أي شيء "ElY >y $y'")
- not to segment name mentions and hashtags

In what follows, we discuss the advantages and disadvantages of segmenting raw text versus the CODA'fied text with some statistics obtained for the Egyptian tweets for which we have a CODA'fied version as exemplified in Table 4.3. The main advantage of segmenting raw text is that it does not need any preprocessing tool to generate CODA orthography, and the main advantage of CODA is that is regularizes text making it more uniform and easier to process. We manually compared the CODA version to the raw version of 2,000 words in our Egyptian dataset. We found that in 75.4% of the words, segmentation of original raw words is exactly the same as their CODA compliant equivalents (ex. ومن "w+mn" (and from) and نعملها, "nEml+hA" (we do it)). Further, if we normalize some characters, namely أ ، إ ، آ ← ا، ى ← ي، ة ← ه, and

9 Complete list of guidelines is found in at:http://alt.qcri.org/resources/da_resources/seg-guidelines.pdf

non-Arabic characters ي ← ك، ک ← گ، ف ← ڤ، ج ← چ, and remove diacritics, the percentage of matching increases to 90.3%. Table 4.4 showcases the remaining differences between raw and CODA segmentations and how often they appear. The differences are divided into two groups. In the first group (accounting for 6.8% of the cases), the number of word segments remains the same and both the raw and CODA'fied segments have the same POS tags.

Diff.	%	Examples
Same no. of segments and same POS tags		
variable spellings	2.4%	عشان ⇔ علشان "EAn, ElAn"
dropped letters	2.3%	ب+ حترم ⇔ بـ+أحترم "b+Htrm, b+AHtrm"
merged words	1.4%	يا+عم ⇔ يا عم "yA+Em, yA Em"
Shortened particles	0.4%	ع ⇔ على، ف ⇔ في "E, ElA f, fy"
elongations	0.3%	ليه ⇔ ليييه "lyyyyh,lyh"
Different no. of segments or POS tags		
spelling errors	2.2%	انا ⇔ ان ولا ⇔ وإلا "An, Ana wlA, wAlA"
fused letters	0.8%	قالهي ⇔ قال لهي "qAl+y, qAl l+y"

Table 4.4: Original vs CODA Segmentations

In this group, the "variable spelling" class contains dialectal words that may have different common spellings with one "standard" spelling in CODA. They are different than their CODA orthography, which is a common case in DA as there is no standard writing and many writings for the same word are accepted. This is similar to having "thx" and "gr8" as informal writings for the English words "thanks" and "great". The "dropped letter" and "shortened

particle" classes typically involve the omission of letters such as the first person imperfect prefix أ ">" when preceded by the present tense marker ب "b" or the future tense marker هـ "h", and the "A" in negation particle ما "mA" which is often written as م "m" and attached to the following word, and the trailing letters in prepositions. "Merged words" and "word elongations" are common in social media, where users try to keep within limit by dropping the spaces between letters that do not connect or to stress words respectively. Though some processing such as splitting of words or removing elongations is required to overcome the phenomena in this group, in situ segmentation of raw words would yield identical segments with the same POS tags as their CODA counterparts. Thus, the segmentation of raw words could be sufficient for 97% of words.

In the second group (accounting for 3% of the cases), both may have a different number of segments or POS tags, which would complicate downstream processing such as POS tagging. They involve spelling errors and the fusion of two identical consecutive letters (gemmination). Correcting such errors may require a spell checker. We opted to segment raw input without correction in our reference, and we kept stems, such as verbs and nouns, complete at the expense of other morphological segments such as prepositions as shown in Table 4.3.

In summary, segmenting original words is very close to segmenting CODA-fied words (at least 95%), and this ratio can be increased to 98% by considering special predefined cases. It can be applied to raw text directly without the need of tools or lexicons for each dialect. The numbers reported here are for EGY dialect, and we plan to study other dialects for different segmentation conventions.

4.4 CONCLUDING REMARKS

In this chapter, we present two dialectal corpora with annotation on token level. Both corpora were collected from internet discussion forums and blogs, social media platforms. They will be of use for supporting research in linguistics and NLP and will constitute an ideal data source for multilingual processing, an area which recently has received increased attention. Given these two new annotated datasets, in the next two chapters we will investigate two challenging problems: Dialectal Arabic segmentation and code-switching identification.

Part IV

APPLICTAIONS

5

CODE-SWITCHING IDENTIFICATION

This chapter includes work that was published as Samih and Maier (2016a,b). I was primarily responsible all the experiments and the design and creation of the corpus used in this chapter. The other authors contributed to these papers primarily in an advisory role. In addition, Section 5.5 in this chapter presents work that is excerpted from Samih et al. (2016), which also forms the basis for much of this chapter. I was responsible for the design of the neural models presented in that paper.

5.1 INTRODUCTION

CS can be defined as the alternation between elements of two or more languages or language varieties within the same utterance. The speaker's native language is sometimes referred to as the 'host language', and his second language, the embedded language, as the 'guest language' (Yeh et al., 2013). Code-switching is a wide-spread linguistic phenomenon in modern informal user-generated data, whether spoken or written. With the advent of social media such as Facebook posts, Twitter tweets, SMS messages, user comments on the articles, blogs, etc., this phenomenon is becoming more pervasive. Code-switching not only occurs across sentences (inter-sentential) but also within the same sentence (intra-sentential), adding a substantial complexity dimension to the automatic processing of natural languages (Das and Gambäck, 2014). This phenomenon is particularly frequent in multilin-

gual societies (Milroy and Muysken, 1995), migrant communities (Papalexakis, Nguyen, and Doğruöz, 2014), and in other environments that undergo social changes through education and globalization (Milroy and Muysken, 1995). There are also some social, pragmatic, and linguistic motivations for code-switching, such as the the intent to express group solidarity, establish authority (Chang and Lin, 2014b), lend credibility, or compensate for lexical gaps.

It is not necessary for code-switching to occur only between two different languages like Mandarin-Taiwanese (Yu et al., 2012), Spanish-English (Solorio and Liu, 2008b) and Turkish-German (Çetinoglu, 2016), but it can also happen between three languages, e.g. Bengali, English and Hindi (Barman et al., 2014), and in some extreme cases even between six languages: English, French, German, Italian, Romansh, and Swiss German (Volk and Clematide, 2014). Moreover, this phenomenon can occur between two different dialects of the same language such as between MSA and EGY (Elfardy and Diab, 2012a), or MSA and Moroccan Arabic (Samih and Maier, 2016a,b).

With the massive increase in code-switched writings in user-generated web content, it has become imperative to develop tools and methods to handle and process this type of data. Identification of languages used in the sentence is the first step in doing any kind of text analysis. For example, most data found in social media produced by bilingual people is a mixture of two languages. In order to process or translate this data to some other language, the first step is to detect text chunks and identify which language each chunk belongs to.

In this chapter, we present two code-switching identification systems for MSA-Moroccan Arabic. We first present the code-witching dataset used throughout the work. We then present a baseline system utilizing Conditional Random Field (CRF) and discuss its shortcomings. Later, we present

another system based on a Deep Neural Network (DNN) and demonstrate the improvements achieved in code-switching identification. This latter does not rely on any human hand-crafted features and is language agnostic. Following recent advances in artificial neural network research, it employs word embeddings and character-level embeddings to achieve state-of-the-art performance.

5.2 CODE-SWITCHING

5.2.1 *A Linguistic Analysis of Code-Switching*

Code-switching[1] is a common phenomenon in multi-lingual communities. It is a process whereby speakers switch from one language or dialect to another within the same context (Bullock and Toribio, 2009). Communities where commonly more than one language or dialect is spoken can be found around the world. Examples include India, where speakers switch between English and Hindi (among other local languages) (Dey and Fung, 2014); the United States, where migrants from Spanish-speaking countries continue to use their native language alongside English (Poplack, 1980); Spain, where people switch between regional languages such as Basque and Spanish (Barredo, 2003); Paraguay, where Spanish co-exists with Guarani (Estigarribia, 2015); and finally the Arab world, where speakers alternate between MSA and Dialectal Arabic.

Since the mid 1960s there has been a large body of linguistic studies on code-switching, the bulk of them concentrating on social and linguistic factors that constrain its occurrences (Berk-Seligson, 1986). Various models of constraints have been proposed. Poplack (1980) formulates a model in

1 Note that for the purpose of this dissertation, we do not distinguish between *code-switching* and similar concepts such as *code-mixing*.

terms of the *equivalence constraint* of the languages involved at the switch point. Namely, code-switching tends to occur at points in the sentence where the surface structure of the respective languages is the same. Myers-Scotton (1993) focuses on structural constrains in code-switching. She proposes the *matrix language-frame* (**MLF**), which is based on the assumption that one language is the matrix language (**M**) and the other language is the embedded language (**E**). While **M** provides the grammatical and functional elements as well as the structural frame of the sentence, **E** can only provide content elements (Myers-Scotton, 1997). For a further, detailed linguistic overview, consult Muysken (2000).

In the literature, three types of code-switching are distinguished. In *inter-sentential* code-switching languages are switched between sentences. An instance of this type of switching is (1) from Barredo (2003), where the speaker switches from Basque to Spanish.

(1) egia ez dala erreala? *eso es otra cosa!*
you say that the truth is not real? *that's a different thing!*

Intra-sentential code-switching consists of a language switch *within* a sentence. An example is (2) borrowed from Dey and Fung (2014). Here, the speaker switches from Hindi to English within the same sentence.

(2) Tume nahi pata, *she is the daughter of the CEO,* yaha do char din ke liye ayi hai.
Don't you know, *she is the daughter of the CEO*, she's here for a couple of days.

A third type of code-switching is *intra-word* switching, where a language switch occurs in a single word. For instance, the morphology of one lan-

guage involved can be applied to a stem of the other language. This is exemplified by (3) for Spanish/Quechua (borrowed from Muntendam, 2006). Quechua is the matrix language of the sentence. The Spanish words *farol* with its diminutive and the word *dueña* are used together with Quechua suffixes. A corresponding example of switching between Modern Standard Arabic and Moroccan dialectal Arabic within the same sentence can be found in the Section. 5.2.2.

(3) Chanta **farol-cito**-wan llojsi-mu-sqa **dueña**-n-qa
Then lantern-DIM-INSTR come out-DIR-PAST 3 SG owner-POSS-TOP
'Then her **owner** appeared with **a small lantern**'

5.2.2 *Code-Switching in Morocco*

The linguistic situation in Morocco is complex due to its diverse ethnic and linguistic make-up and the colonial history. Following Benmamoun (2001), one can distinguish different languages and dialects that occupy the linguistic space:

Darija is the native language for the majority of the population and is the language of popular culture.

Berber is the language of the original people of Morocco and is the native language of about 40% of the Moroccan population.

Modern Standard Arabic is a written language used mainly in formal education, media, administration, and religion.

French is not an official language, but is dominant in higher education, in the media, and some industries.

In recent years, the Moroccan linguistic landscape has changed dramatically due to social, political, and technological factors. Darija, the colloquial, tradi-

tionally *unwritten* variety of Arabic, is increasingly dominating the linguistic scene. It is written in a variety of ways in print media, advertising, music, fictional writing, translation, the scripts for dubbed foreign TV series, and a weekly news magazine (Elinson, 2013). It also increasingly appears on the web in blogs, emails, and social media platforms and is often code-switched with other languages and dialects, including MSA, English, and French, Spanish and, Berber (Voss et al., 2014). As an example for intra-sentential and intra-word code-switching in Morocco, consider (4), taken from our own data.

(4) فرنسا لن يبقى فيها سوى الزواق و الشيكي و الجيب خاوي و نزيدكم و الاكل بالموس و الفورشيطا.

In France, the only things that remain are pretension, empty pockets, and, to add more, also eating with knife and fork.

The speaker switches between MSA and Darija, and uses a word where French is mixed with Arabic morphology. MSA words include, e.g., فرنسا (*France*), الاكل بالموس (*eating with knife*); Darija words include الزواق و الشيكي و الجيب خاوي و نزيدكم (*pretension, empty pockets, and, to add more*); finally, الفورشيطا is the French word for *fork* (*'fourchette'*), written in Arabic script. It is prefixed by the Arabic definite article and suffixed with an Arabic case marker.

5.3 RELATED WORK

Code-Switching has attracted considerable attention in theoretical linguistics and sociolinguistics over several decades. However, there has not been much recent work on the computational processing of code-switched data. The first computational treatment of this linguistic phenomenon can be found in (Joshi, 1982). He introduces a grammar-based system for parsing and generating code-switched data. More recently, the detection of code-switching has gained attention, starting with the work of (Solorio and Liu, 2008b), and culminating in the first shared task at the "First Workshop on Computational Approaches to Code Switching" (Solorio et al., 2014b) and the 'Second Workshop on Computational Approaches to Code Switching' (Molina et al., 2016). Moreover, there have been efforts in creating and annotating code-switching resources (Çetinoglu, 2016; Elfardy and Diab, 2012a; Lignos and Marcus, 2013; Maharjan et al., 2015b; Samih and Maier, 2016b). Maharjan et al. (2015b) used a user-centric approach to collect code-switched tweets for Nepali-English and Spanish-English language pairs. They used two methods, namely a dictionary based approach and CRF and obtained an accuracy of 86% and 87% for Spanish-English and Nepali-English respectively at word level language identification task. Lignos and Marcus (2013) collected a large number of monolingual Spanish and English tweets and used a ratio list method to tag each token with its dominant language. Their system obtained an accuracy of 96.9% at word-level language identification task.

The task of detecting code-switching points is generally cast as a sequence-labeling problem. Its difficulty depends largely on the language pair being processed. Several projects have treated code-switching between MSA and Egyptian Arabic. For example, Elfardy, Al-Badrashiny, and Diab (2013)

present a system for the detection of code-switching between MSA and Egyptian Arabic which selects a tag based on the sequence with a maximum marginal probability, considering 5-grams. A later version of the system named AIDA2 (Al-Badrashiny, Elfardy, and Diab, 2015) is a more complex hybrid system that incorporates different classifiers and components such as language models, a named entity recognizer, and a morphological analyzer. The classification strategy is built as a cascade voting system, whereby a CRF classifier tags each word based on the decisions from four other underlying classifiers.

The participants of the 'First Workshop on Computational Approaches to Code Switching' applied a wide range of machine learning and sequence learning algorithms with some researchers using additional online resources like an English dictionary, a Hindi-Nepali wiki, dbpedia, online dumps, LexNorm, etc. to tackle the problem of language detection in code-switched tweets on Nepali-English, Spanish-English, Mandarin-English, and MSA Dialects (Solorio et al., 2014b). For MSA-Dialects, two CRF-based systems, a system using language-independent extended Markov models, and a system using a CRF autoencoder were presented; the latter proved to be the most successful. Similarly in the 'Second Workshop on Computational Approaches to Code Switching', the participants also applied a wide variety of system architectures ranging from simple rule based systems all the way to more complex deep learning implementations (Molina et al., 2016).

The majority of the systems dealing with word-level language identification in code-switching rely on linguistic resources (such as named entity gazetteers and word lists) and linguistic information (such as POS tags and morphological analysis), and they use ML methods that have been typically used with sequence-labeling problems, such as SVM, CRF and n-gram language models. A few investigators, however, have recently turned to RNNs

and word embedding with remarkable success. (Chang and Lin, 2014b) used an RNN architecture and combined it with pre-trained *word2vec* skip-gram word embeddings, a log bilinear model that allows words with similar contexts to have similar embeddings. The *word2vec*[2] embeddings were trained on a large Twitter corpus of random samples without filtering by language, assuming that different languages tend to share different contexts, allowing embeddings to provide good separations between languages. They showed that their system outperforms the best SVM-based systems reported in the EMNLP'14 Code-Switching Workshop. Similarly, Samih et al. (2016) make use of word embeddings and Long short-term memory (LSTM) with the added advantage of memory cells that efficiently capture long-distance dependencies. They also combine word-level with character-level representations to obtain morphology-like information on words.

Vu and Schultz (2014) proposed adapting the recurrent neural network language model to different code-switching behaviors and even use them to generate artificial code-switching text data. Adel, Vu, and Schultz (2013) investigated the application of RNN language models and factored language models to the task of identifying code-switching in speech and reported a significant improvement compared to the traditional n-gram language model.

5.4 DATASET

A detailed description of the data set we use is provided in Section 4.2. This data set contains 3,865 text units. While a large portion of them are short (cf. Figure 4.1), the average length is still 57.9, and the median 46 words. Of all text units, 287 (7.4%) contain no code-switching (i.e., no labels `lang2` or `lang3` are present).

2 https://code.google.com/archive/p/word2vec/

	all	%	av.len.	med.len.	training	dev	test
text units	3,865				3,089	388	388
tokens	223,284				178,924	21,925	22,435
lang1	109,025	48.82	6.13	4	86,464	10,988	11,573
lang2	76,732	34.36	5.22	4	62,261	7,230	7,241
lang3	1,383	0.62	1.88	1	1,138	90	155
ne	17,087	7.65	1.29	1	13,658	1,775	1,659
mixed	86	0.04	1.00	1	72	8	6
ambiguous	141	0.06	1.26	1	120	8	13
other	18,830	8.43	1.02	1	15,211	1,826	1,793

Table 5.1: Data statistics

In preparation for our experiments, we transformed Arabic scripts to SafeBuckwalter (Roth et al., 2008), a character-to-character mapping that replaces Arabic UTF-8 alphabet with Latin characters to reduce size and streamline processing. In order to reduce data sparsity, we converted all Persian numbers (e.g. ۲ ،۱) to Arabic numbers (e.g. 1, 2), Arabic punctuation (e.g. '،' and '؛') to Latin punctuation (e.g. ',' and ';'), removed kashida (elongation character) and vowel marks, and separated punctuation marks from words. We then split the data into three parts for training, development, and testing. In order to avoid imbalance, we use a round-robin[3] split. The training set receives roughly 80% of the tokens, and the test and development sets roughly 10% each. Tab. 5.1 shows the properties of the resulting sets.

5.5 CODE-SWITCHING DETECTION APPROACHES

In this section, we present two different systems for code-switching identification in social media.These two systems were designed for this dissertation, and are entirely my own. The first approach uses CRF (Lafferty, McCallum,

3 A round robin is an arrangement of choosing all elements in a group equally in some rational order, usually from the top to the bottom of a list and then starting again at the top of the list and so on.

and Pereira, 2001) to tag each word using a variety of features. This system yields state-of-the-art results on the Moroccan code-switching corpus, which is the reference dataset for comparing our proposed CS identification systems in this dissertation. This system will constitute a challenging baseline for the our proposed DNNs based model, which performs sequence-to-sequence mapping to identify CS.

5.5.1 *CRF Approch (Baseline)*

Code-switching can be treated as a sequence-labeling problem. We use a CRF (Lafferty, McCallum, and Pereira, 2001) classifier to combine knowledge from different sources. This has proven to be successful in previous work (Elfardy, Al-Badrashiny, and Diab, 2014b). For our experiments, we recur to *Wapiti* (Lavergne, Cappé, and Yvon, 2010),[4] a state-of-the-art high-speed CRF implementation with flexible configuration options. We run all experiments with standard settings, namely Limited-memory BFGS (L-BFGS)[5] optimization with elastic net penalty and no iteration limit. For evaluation we compute the accuracy for all labels, and we compute precision, recall and F_1 for each label separately.

Since performing experiments with all possible feature combinations is not feasible, we begin with a small feature set and iteratively add more features to the best combination obtained in the respective previous step. For presentation, we organize the features in four groups. The first feature group consists of the focus word and its surrounding words, as well as their prefixes and suffixes. The second group adds structural properties of the words,

4 See https://wapiti.limsi.fr/ for details.

5 It is an optimization algorithm in the family of quasi-Newton methods that approximates the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm using a limited amount of computer memory. It is a popular algorithm for parameter estimation in machine learning. https://en.wikipedia.org/wiki/Limited-memory_BFGS

such as whether they contain numbers. The third group leverages character language models, and the fourth group adds lexical knowledge from external resources, e.g., from MSA word lists. We aim at a high overall performance. Therefore, we take the feature combination with the highest overall accuracy as the best one, even though it may actually not the most fortuitous choice with respect to a certain label. Feature selection is performed on the dev set. The best combination is evaluated on the test set. For the actual experiments, we take advantage of the Wapiti template file mechanism.

TOKENS, PREFIXES, AND SUFFIXES First, as features we use the focus word itself and the surrounding words, both as unigram and bigram features (i.e., using either only the current label, or the joint current and previous labels), both alone and in sequence (i.e., we build n-grams of adjacent tokens). We also employ different window sizes from zero to three words to the left and to the right. The best result is obtained by considering three tokens on both sides of the focus words, using as features all tokens as unigrams as well as all possible pairs of adjacent tokens, and finally also the joint label consisting of the current one and the previous output tag (`b` in the Wapiti template). Particularly long combinations where not useful and it did not help to use more complex bigram features. In the following, we employ prefixes and suffixes of varying length as features, both alone and in sequence as before, in addition to the best feature combination so far. The intuition behind this is that particularly for Arabic, they should convey crucial information. The best result is obtained using as features the prefixes and suffixes of both length one and two of the focus word. Most likely due to data sparseness, neither using longer combinations nor a larger window was useful. Using prefixes of length 3 also did not help, which can be explained with the small average word length of only 4.0 characters. The inclusion of bigram features (with

the joint current and previous output label) did not help; furthermore, due to the high number of extracted features, training the respective models was also much more costly. We therefore refrained from using complex bigram features in the further experiments, and only included the feature consisting of the current and the previous label.

STRUCTURAL TOKEN PROPERTIES In our present work, we add features which model structural properties of the words to the previously winning combination. We use four Boolean features.

- NUMBER is true if a token contains a number.
- LETTER is true if a token starts with a letter
- LENGTH contains the token length. We use LENGTH as a discrete feature, the reasoning behind it being that tokens are rather short: the average token length is just 4.03.
- BUCKWALTER is true if a token contains a character which is not part of the Buckwalter transliteration set (or character which corresponds to a diacritic, see above) (Buckwalter, 2001). The intuition behind this feature is particularly to capture words in `lang3` and `ne`.

Experiments are performed with the cross-product of:

1. all 15 combinations of the four features,
2. considering zero to three words around the focus word,
3. and building n-grams of one, two, and three tokens.

The best accuracy is obtained using LETTER and BUCKWALTER together as unigrams, considering a window of one word around the focus word.

LANGUAGE MODELS We now exploit the fact that character n-gram language models have proven to be successful for code-switching detection (Al-Badrashiny, Elfardy, and Diab, 2015), and also more general text categorization tasks (Maier and Gómez-Rodríguez, 2014, e.g.). We build a character language model for each label (i.e., We end up with seven models) from the training set, treating tokens as sequences. We use the Kneser-Ney Language Model (LM) implementation in Berkeley LM (Pauls and Klein, 2011)[6], choosing $n = 5$. We then obtain an LM score for each model and token. In order to use the scores as discrete features, we bin[7] the scores with bin sizes 10, 100, 500, and 1,000, and use the respective bin index as feature (LM^i, $i \in \{10, 100, 500, 1000\}$). Furthermore, we use an additional group ARGSORT of seven features in which the ith feature has the value i if label i has the ith highest score. Note that since no morphological analyzer is available for Moroccan Arabic is available, we cannot recur to the more sophisticated approach of AIDA (Al-Badrashiny, Elfardy, and Diab, 2015), in which character LMs for prefixes and suffixes are used. In a first step, we perform five experiments. First we add all ARGSORT features to the previously winning combination (as non-combined features for the focus word); in each of the remaining four, I add all features of one of the four LM^i feature groups. The best overall result for the LM features can be obtained by including all LM^{100} features for the focus word and a window of three words around it. As a result, we obtain improvements in all labels (except `ambiguous`) and a particularly large improvement for `lang3`. This confirms our intuition from previous work that the character LMs can compensate for the fact that `lang3` is a heterogeneous label (encompassing words from different languages), even though there is

6 https://code.google.com/p/berkeleylm/

7 Data binning or bucketing is a data pre-processing technique used to reduce the effects of minor observation errors. The original data values which fall in a given small interval, a bin, are replaced by a value representative of that interval, often the central value. https://en.wikipedia.org/wiki/Data_binning

not much data to learn from. One might suppose that adding the LM features for only `lang3` and `mixed` would be enough; however, this is not the case: The corresponding experiments turned out slightly worse. Generating n-gram feature combinations and considering a window around the focus word, as before, also did not lead to better results.

LEXICAL KNOWLEDGE In a last step, we add binary features which model MSA and Darija lexical knowledge.

- The first feature MSALIST models the affinity of a certain token to MSA. It is based on a list of 9.1m MSA tokens obtained from the spell-checking model of (Attia et al., 2015), and is true for all tokens which have a match in the list.
- Similarly, the second feature DARIJALIST models the affinity to Darija. We use a manually compiled list of 703 typical Darija terms and set the feature true for all tokens which have a match in this list.
- A further feature DARIJAMORPH is set to true if the token contains components which are typical for Darija tokens. We set the feature to true for all tokens which end with *$*, or which start with any of *ha*, *ky*, *gy*, *ty*, *by*. Furthermore, it is set true for all *mA* tokens which are followed by a token ending with *$*.
- In order to improve the recognition of named entities, we recur to a gazetteer, namely ANERgazet (Benajiba and Rosso, 2008), and set the corresponding feature NER to true for all tokens that match an entry in the gazetteer. For our purpose, we do not distinguish between different types of named entities, as available in ANERgazet. In contrast to Elfardy, Al-Badrashiny, and Diab (2014b), we do not use the training set from the 2014 EMNLP code-switching workshop shared task (Solorio

et al., 2014b) due to the fact that we found a high number false positives in the annotation.

5.5.2 *A Deep Neural Network Model*

5.5.2.1 *System Architecture*

In this subsection, we provide a brief description of the different components of our deep neural code-switching detection model. The architecture of our system, shown in Figure 5.1, bears resemblance to the models introduced by Huang, Xu, and Yu (2015), Ma and Hovy (2016), and Collobert et al. (2011) among others.

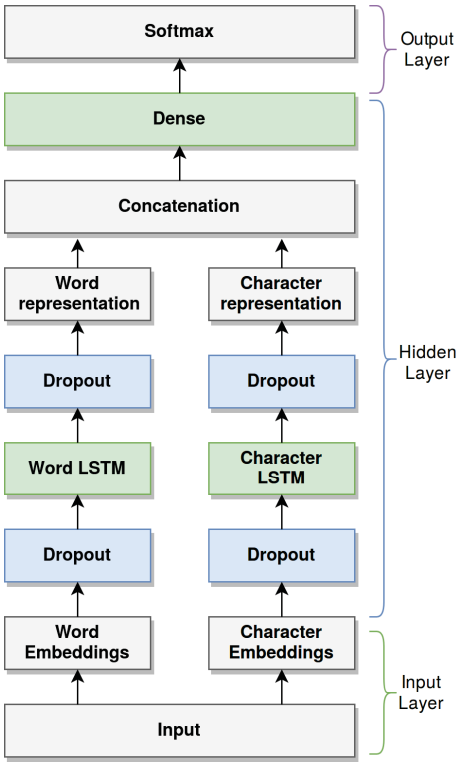

Figure 5.1: Code-switching detection architecture.

Our proposed neural network architecture consists of the following three layers:

- Input layer: comprises both character and word embeddings.
- Hidden layer: two bidirectional LSTMs map both words and character representations to hidden sequences.
- Output layer: a Softmax computes the probability distribution over all labels.

WORD EMBEDDINGS Another crucial component of our model is the use of pre-trained vectors. The basic assumption is that words from different languages (or language varieties) may appear in different contexts, so that word embeddings learned from a large multilingual corpus should provide an accurate separation between the languages or dialects at hand. Following Collobert et al. (2011), we use pre-trained word embeddings for Moroccan Arabic to initialize our look-up table. Words with no pre-trained embeddings are randomly initialized with uniformly sampled embeddings. To use these embeddings in our model, we simply replace the one hot encoding word representation with its corresponding 300-dimensional vector. For Moroccan Arabic, we trained different word embeddings using Word2Vec[8] (Mikolov et al., 2013) from a corpus of total size of 166,206,215 words, consisting of user comments on the news, and MSA texts of news articles. For more details about the data statistics, see Table 5.2.

Genre	Tokens
Moroccan News comments	20,241,480
MSA news texts	145,965,735
Total	166,206,215

Table 5.2: character embeddings training data statistics

8 https://code.google.com/archive/p/word2vec/

CHARACTER EMBEDDINGS A very important element of the recent success of many NLP applications is the use of character-level representations in deep neural networks. This has shown to be effective for numerous NLP tasks (Collobert et al., 2011; Santos et al., 2015) as it can capture word morphology and reduce out-of-vocabulary problems. This approach has also been especially useful for handling languages with rich morphology and large character sets (Kim et al., 2016). We find this important for our code-switching detection model particularly for words that are not written in the Arabic script; they have different orthographic sequences which are learned during the training phase.

5.5.2.2 *Parameterization and Training*

At the input layer a look-up table is randomly initialized mapping each word in the input to d-dimensional vectors of sequences of characters and sequences of words. At the hidden layer, the output from both character and word embeddings is used as the input to two LSTM layers to obtain fixed-dimensional representations for characters and words. At the output layer a softmax is applied over the hidden representation of the two LSTMs to obtain the probability distribution over all labels. The model is trained using Stochatic Gradient Descent (SGD) with momentum, optimizing the cross entropy objective function.

PARAMETER INITIALIZATION For the word embeddings, we use our pre-trained Word2Vec 300-dimensional embeddings. We also use randomly initialised embeddings that are uniformly sampled from range $[-\sqrt{\frac{3}{d}}, +\sqrt{\frac{3}{d}}]$ where $d = 300$ is the embedding dimension [9]. Character embeddings are also

9 We conducted side experiments without word pre-trained embeddings and the results were very low

initialized with uniformly sampled embeddings from range $[-\sqrt{\frac{3}{d}}, +\sqrt{\frac{3}{d}}]$ proven where $d = 50$.

REGULARISATION Due to the relatively small size of the training and the development data sets, overfitting poses a considerable challenge for our code-switching detection system. To make sure that our model learns significant representations, we resort to dropout (Hinton et al., 2012) to mitigate overfitting. The basic idea of dropout consists in randomly omitting a certain percentage of the neurons in each hidden layer for each presentation of the samples during training. This encourages each neuron to depend less on other neurons to detect code-switching patterns. We apply dropout masks to both embedding layers before inputting to the two BiLSTMs and to their output vectors as shown in Fig. 5.1. In our experiments we find that dropout decreases overfitting and improves the overall performance of the system. We also employ early stopping (Caruana, Lawrence, and Giles, 2000; Graves, Mohamed, and Hinton, 2013) by monitoring the model's performance on the development set.

HYPER-PARAMETERS TUNING We tuned our hyper-parameters on the development dataset by using random search. Table 5.3 provides a summary of the hyper-parameters we use in all our experiments. Even though this list of parameters is not exhaustive, it dramatically influences the network's accuracy.

5.6 RESULTS AND EXPERIMENTS

We experimented with a variety of system architectures ranging from simple linear graphical models with hand-crafted features (Section 5.5.1) all

Layer	Hyper-Parameters	Value
Characters Bi-LSTM	state size	100
	initial state	0.0
Words Bi-LSTM	state size	400
	initial state	0.0
Dropout	dropoutrate	0.5
Characters embeddings	embedding dimension	50
Words embedding	embedding dimension	300
	batch size	15
	learning rate	0.01
	decay rate	0.05

Table 5.3: Hyper-Parameters Tuning

the way to more complex deep learning models to identify the best system. We compare the performance of our proposed model with three baseline systems – CRF(feat.), a CRF with hand-crafted features; CharBLSTM, the bi-directional characters LSTM, and WordBLSTM, the bi-directional words LSTM. We run the WordBLSTM model using the same hyper-parameters as shown in Table 5.3 and the google's Word2Vec 300 dimentional words embedding. Table 5.4 shows the results of the different settings on the test dataset. The CRF(feat.) obtains better performance than both CharBLSTM and WordBLSTM. This shows that a CRF with carefully engineered features can rival neural models. However, combining the character and word representations learned by the CharWordBLSTM system with words embeddings and then applying the Softmax as a sequence classifier gives the highest overall accuracy of 92.50 and significantly outperforms the other models on all the evaluation metrics. This also shows the benefits of adding character representations for linguistic sequence labelling tasks. It significantly improves the F1-score for `ne` and `lang3` tokens. This results are in line with results reported by previous work (Al-Badrashiny and Diab, 2016; Samih et al., 2016).

Labels	CRF(feats)	wordBLSTM(emb.)	CharBLSTM	CharWordBLSTM(emb.)
ambiguous	0.00	0.00	0.00	0.00
lang1	0.92	0.92	0.81	**0.93**
lang2	0.87	0.88	0.63	**0.90**
lang3	**0.77**	0.38	0.58	0.70
mixed	0.00	0.00	0.00	0.00
ne	**0.91**	0.78	0.79	0.85
other	0.99	0.99	0.99	0.99
Accuracy	91.40	90.25	77.39	**92.5**

Table 5.4: F1 score results on MSA-Moroccan test dataset. (feats = hand-crafted features, emb. = word embeddings).

Unlike the CRF(feat.) our system, CharWordBLSTM, does not rely on any kind of hand-crafted features or external resources to achieve good performance. Table 5.4 shows the final results for the test dataset, the difference between the CharWordBLSTM system and the CRF(feat.) system is 1.1% in terms of token level accuracy. It consistently ranks first for code-switching identification for the test dataset (1% and 3% above the CRF(feat.) system for lang1 and lang2 respectively). However, it lags behind CRF(feat.), showing weaker performance in identifying ne and lang3 labels. Nonetheless, the overall results show that our system outperforms other systems with relatively high margin.

5.7 ANALYSIS

5.7.1 *What is Captured in Char-Word Representations?*

In order to investigate what the char-word LSTM model is learning, we feed the posts from the MSA-Moroccan development dataset to the system and take the vectors formed by the concatenation of both character and word representations. We then project them into 2D space by reducing the dimen-

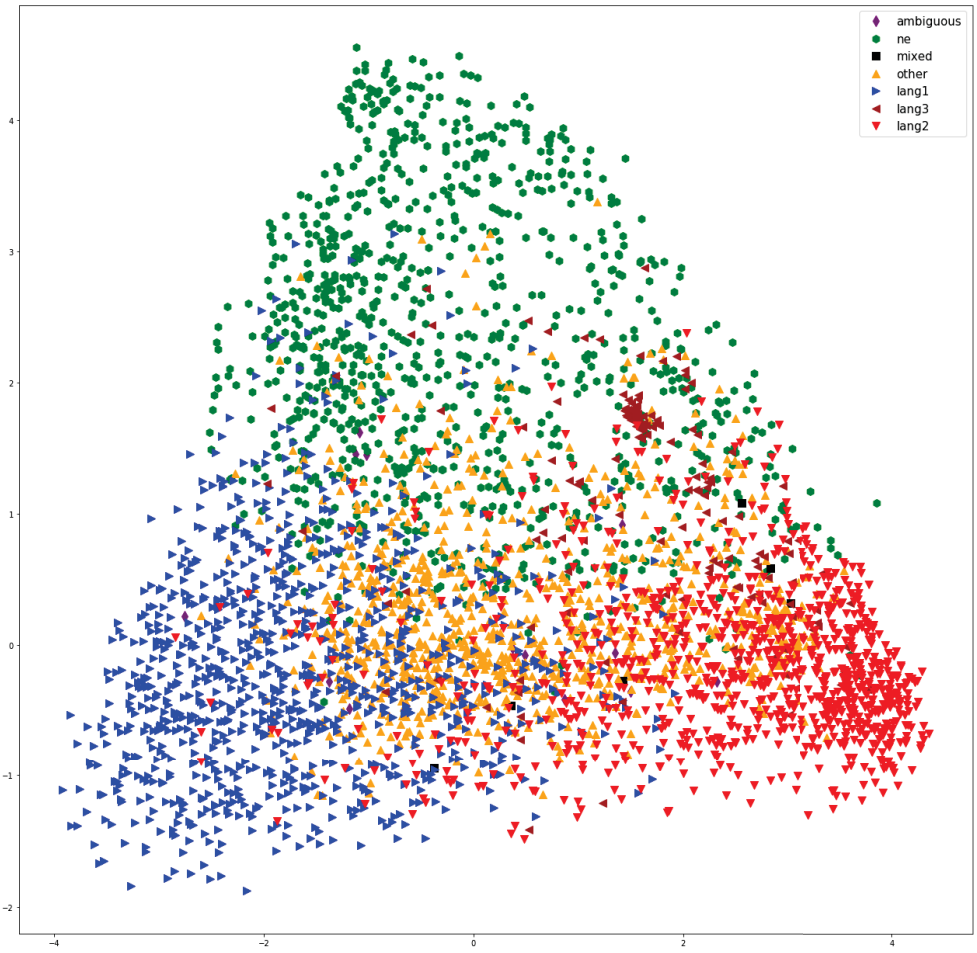

Figure 5.2: Projection of char-word LSTM representation into 2D using PCA.

sion of the vectors to 2 using Principle Component Analysis (PCA)[10]. We see in Figure 5.2, that the trained neural network has learned to cluster the tokens according to their label type. Moreover, the position of tokens in 2D space also revels that *ambiguous* and *mixed* tokens are in between `lang1` and `lang2` clusters. Figure 5.2 shows generally successful separation of tokens, with `lang1` in blue on the left, `lang2` in red on the right, and `ne` in green on the top. The *other* token occupies the space between the clusters for `lang1`, *lang2* and `ne` with more inclination toward `lang1`.

10 A linear dimensionality reduction method which uses Singular Value Decomposition (SVD) of the data in order to project it to a lower dimensional space.

5.7.2 *CRF Model*

Table 5.5 shows the most likely and unlikely transitions learned by the CRF(feat.) model for the Moroccan-MSA dataset. It is interesting to see that the transition from `lang1` to `lang2` and from `lang2` to `lang1` are much likely than `lang1` to `lang1` or `lang2` to `lang2`. This suggests that people in this informal setting switch from one language to another while communicating.

Most likely	Score	Most unlikely	Score
lang1 ⇒ lang2	1.459	lang1 ⇒ mixed	-0.154
ne ⇒ ne	1.264	mixed ⇒ lang1	-0.169
lang2 ⇒ lang1	1.193	amb ⇒ other	-0.242
lang3 ⇒ lang2	1.137	ne ⇒ mixed	-0.213
lang1 ⇒ lang3	1.099	mixed ⇒ other	-0.246
other ⇒ other	0.827	amb ⇒ lang1	-0.272
lang1 ⇒ ne	0.316	ne ⇒ lang2	-0.364
other ⇒ lang1	0.222	lang3 ⇒ ne	-0.443
lang2 ⇒ mixed	0.216	lang1 ⇒ lang1	-0.840
lang1 ⇒ other	0.191	lang2 ⇒ lang2	-0.923

Table 5.5: Most likely and unlikely transitions learned by CRF(feat.) model for the Moroccan-MSA dataset.

However, people writing in *Darija* have more tendency to use mixed tokens than people writing in MSA. We also dumped the top features for the task and found that the LM features are the top features to identify token as `lang3`. Moreover, the token structural features are top features to identify tokens as `other`. The features like *char bigram, trigram, words, suffix, and prefix* are the top features to distinguish between MSA and Moroccan Arabic tokens.

5.7.3 *Error Analysis*

When we conducted an error analysis on the output of the Arabic development set for our system, we found the following mistagging types:

- Bad segmentation in the text affects the decision, e.g. مكيناشأمغتكونش mkynA$Omgtkwn$ 'mkynA$ Omgtkwn$'.
- Abbreviations: أ 'A' "Mr." and م 'm' "eng." are not consistently treated across the dataset.
- There are cases of true ambiguity, e.g. شباط '$bAT', which can be a noun "February" or a person's name "$bAT".
- Clitic attachment can obscure tokens, e.g. وبها wbhA "and-bhA".
- Spelling errors can increase data sparsity, e.g. البيطاله AlbyTAlhp "Alb-TAlp".

Table 5.7.3 provides different examples of gold CS instances predicted by the CharWordBLSTM(emb.) model which the CRF(feat.) model failed to predict and vice versa. This shows that while the CRF(feat.) can only address instances that are manually encoded as features, our proposed neural model can efficiently cope with the noise and the diversity in the social media data. Since it is challenging and time-consuming to design features to model all the occurrences of code-switching, intrinsic flexibility of our neural model allows it to better address variations in the spelling and morphology of words. The CharWordBLSTM(emb.) is able to identify many named entities without relying on explicit gazetteers. This probably can be attributed to the model's character embeddings learned during training and which can effectively cope with Out-Of-Vocabulary (OOV) words. Interestingly enough,

CRF(feat.) is good at rarely occurring patterns or named entities that are included in the gazetteers (e.g., بولیزارو , bwlyzArw). For example, the ne token only occur in the test set, and unless the context gives a strong indication, the CharWordBLSTM(emb.) cannot detect it, whereas the CRF could, as long as it is included in the gazetteers.

Labels	CRF(feat.)	CharWordBLSTM(emb.)
ambiguous	..أما الآخرين غير **آرواس** ... ...OmA Al|xryn gyr **|rwAs** ...	**حسنتم** تعليق ولد الشعب شكرا. **Hsntm** tElyq wld Al$Eb $krA.
lang1	كنطلب من الله **عز و جل** اعاون الملك ديالنا باش اصفي هادوك الشفارة كلهم. knTlb mn Allh **Ez w jl** AEAwn Almlk dyAlnA bA$ ASfy hAdwk Al$fArp klhm.	شوهنا ازين خاصو اسجن شوهنا **امام العالم** . $whnA Azyn xASw Asjn $whnA **AmAm AlEAlm**.
lang2	..اوزين ليس طاشرون او بناي لكن فقط وزير **و باز** . ...Awzyn lys TA$rwn Aw bnAy lkn fqT wzyr **w bAz**.	أحزاب العائلات ! **ما عمرنا مانطفروه !** OHzAb AlEA}lAt **mA EmrnA mAnTfrwh**!
lang3	... و النتيجة قابلة **الباطرون** ديال الجزائر بكل حفاوة... ... w Alntyjp qAblp **Al-bATrwn** dyAl AljzA}r bkl HfAwp...	على قول المصرين: **اسمع كلامك اصدقك اشوف امورك استعجب** . ... ElY qwl AlmSryn: **AsmE klAmk ASdqk A$wf Amwrk AstEjb**.
mixed	... **فيديراليات** جهوية موسعة جهات كلشي ضربا من الخيال و الطنز. ... **fydyrAlyAt** jhwyp mwsEp jhAt kl$y DrbA mn AlxyAl w AlTnz.	يترشح حتى **يكوفري** على تروثه من الضرائب و ... ...ytr$H HtY **ykwfry** ElY trwvh mn AlDrA}
ne	هدي هيا فرصة ليعلن فيها مغرب على بوزبال **بوليزارو** جماعة إرهابية. hdy hyA frSp lyEln fyhA mgrb ElY bwzbAl **bw-lyzArw** jmAEp IrhAbyp.	متستغربش هادشي عند **بوليزبال** عادي شفرة و الارهاب و زيد. mtstgrb$ hAd$y End **bw-lyzbAl** EAdy $frp w AlArhAb w zyd.

Table 5.6: Examples of correctly detected CS instances (in bold) by the CRF(feat.) and CharWordBLSTM(emb.) models for Moroccan code-switched corpus. the examples in the CharWordBLSTM(emb.) column are only predicted by the aforementioned neural model and not predicted by the CRF(feat.) model.

5.8 CONCLUDING REMARKS

In this chapter we present our system for identifying and classifying code-switched data for MSA-Moroccan Arabic. The system uses a neural network architecture that relies on word-level and character-level representations. Our system is language independent in the sense that we have not used any language-specific knowledge or linguistic resources such as POS taggers, morphological analyzers, gazetteers, or word lists to achieve good performance.

6

DIALECTAL ARABIC SEGMENTATION

This chapter also includes work that was published as Eldesouki et al. (2017) and Samih et al. (2017a,b). I was responsible for the design of the neural models presented in these three papers. I substantially contributed to the implementation and testing of these models and in the development of the segmentation algorithm[1]. Eldesouki Mohamed was responsible for the implementation of the SVM models. The other authors contributed to these papers primarily in an advisory role.

6.1 INTRODUCTION

Segmenting Arabic words into their constituent morphological parts is important for a variety of applications such as machine translation, parsing and information retrieval. Though much work has focused on segmenting Modern Standard Arabic (MSA), recent work has begun to examine morphological segmentation in some Arabic dialects.This so-called "Dialectal segmentation" is becoming increasingly important due to the ubiquity of social media, where users typically write in their own dialects as opposed to MSA. Dialectal text poses interesting challenges such as lack of spelling standards, pervasiveness of word merging, letter substitution or deletion, and foreign word borrowing. Existing work on dialectal segmentation focused on building resources and tools for each dialect separately (Habash et al., 2013; Pasha

1 https://github.com/qcri/dialectal_arabic_tools

et al., 2014b; Samih et al., 2017a). The rationale for the separation is that different dialects have different affixes, make different lexical choices, and are influenced by different foreign languages. However, performing reliable dialect identification to properly route text to the appropriate segmenter may be problematic, because conventional dialectal identification can lead to results that are lower than 90% in accuracy (Darwish, Sajjad, and Mubarak, 2014). Thus, building a segmenter that performs reliably across multiple dialects without the need for dialect identification is desirable. In this chapter we examine the effectiveness of using a segmenter built for one dialect in segmenting other dialects. Next, we explore combining training data for different dialects in building a joint segmentation model for all dialects. We show that the joint segmentation model matches or outperforms dialect-specific segmentation models. For this work, we use training data in four different dialects, namely Egyptian Arabic (EGY), Levantine Arabic (LEV), Gulf Arabic (GLF), and Maghrebi Arabic (MGR). We utilize two methods for segmentation. The first poses segmentation as a ranking problem, where we use an Support Vector Machine (SVM) ranker which is described in Appendix B. The second poses the problem as a sequence-labeling problem, where we use a Bidirectional Long Short-Term Memory (BiLSTM) Recurrent Neural Network (RNN) that is coupled with Conditional Random Field (CRF), also introduced in Appendix B, sequence labeler.

6.2 RELATED WORK

Work on dialectal Arabic is fairly recent compared to that on MSA. A number of research projects were devoted to dialect identification (Biadsy, Hirschberg, and Habash, 2009; Eldesouki et al., 2016; Zaidan and Callison-Burch, 2014; Zbib et al., 2012). There are five major dialects including Egyptian, Gulf,

Iraqi, Levantine and Maghrebi. Only a few resources for these dialects are available such as the CALLHOME Egyptian Arabic Transcripts (LDC97T19), which was made available for research as early as 1997. Newly developed resources include the corpus developed by Bouamor, Habash, and Oflazer (2014), which contains 2,000 parallel sentences in multiple dialects and MSA as well as English translations. These sentences were translated by native speakers into the target dialects from an original dialect, Egyptian.

For segmentation, Mohamed, Mohit, and Oflazer (2012) built a segmenter based on memory-based learning. The segmenter has been trained on a small corpus of Egyptian Arabic comprising 320 comments containing 20,022 words from www.masrawy.com that were segmented and annotated by two native speakers. They reported 91.90% accuracy on the segmentation task. MADA-ARZ (Habash et al., 2013) is an Egyptian Arabic extension of the Morphological Analysis and Disambiguation of Arabic (MADA) tool. The authors trained and evaluated their system on both the Penn Arabic Treebank (PATB) (parts 1-3) and the Egyptian Arabic Treebank (parts 1-5) (Maamouri et al., 2014) and they achieved 97.5% accuracy. MADAMIRA[2] (Pasha et al., 2014b) is a new version of MADA that includes the functionality for analyzing dialectal Egyptian. Monroe, Green, and Manning (2014) used a single dialect-independent model for segmenting Egyptian dialect in addition to MSA. They argue that their segmenter is better than other segmenters that use sophisticated linguistic analysis. They evaluated their model on three corpora, namely parts 1-3 of the Penn Arabic Treebank (PATB), the Broadcast News Arabic Treebank (BN), and parts 1-8 of the BOLT Phase 1 Egyptian Arabic Treebank (ARZ) reporting an F1 score of 92.1%.

2 MADAMIRA release 20160516 2.1

6.3 SEGMENTATION DATASETS

We used datasets[3] for four dialects, namely EGY, LEV, GLF, and MGR which are described in Section 4.3. Each dataset consists of a set of 350 manually segmented tweets. Table 6.1 shows segmented examples from different dialects.

Word	Gloss	Segmentation	Dialect
بيقولك "byqwlk"	Is telling you	بـ+يقول+ك "b+yqwl+k"	EGY
و يجي "wyjy"	And he comes	و +يجـ+ي "w+yj+y"	GLF
برد "brd"	I'll return	بـ+رد "b+rd"	LEV
مغتنفاعهم "mgtnfAEhm"	It will not benefit them	مـ+بـ+تنفاع+هم "m+g+tnfAE+hm"	MGR

Table 6.1: Dialect annotation example

6.4 ARABIC DIALECTS

6.4.1 *Similarities Between Dialects*

There are some interesting observations about the similar behavior of different Arabic dialects (particularly those in our dataset) when they diverge from MSA. These observations show that Arabic dialects do not just share commonalities with MSA, but they also share commonalities among themselves. It seems that dialects share some built-in functionalities for generating words, some of which may have been inherited from classical Arabic, where some

3 Datasets are available at http://alt.qcri.org/resources/da_resources/

of these functionalities are absent or much less prominent in MSA. Some of these commonalities include:

- Dialects have eliminated case endings.
- Dialects introduce a progressive particle, e.g. بـيقول 'b+yqwl' (EGY), عمـيقول 'Em+yqwl' LEV, كـيقول 'k+yqwl' MGR, and دـيقول 'd+yqwl' IRQ for 'he says'. This does not exist in MSA.
- Some dialects use a post-negative particle, e.g. مـيحبـش "m+yHb+$" 'does not like' (EGY, LEV, and MGR). This also does not exist in either MSA or GLF.
- Dialects have future particles that are different from MSA, such as ح "H" (LEV), هـ "h" EGY, and غ "g" MGR. Similar to the MSA future particle س "s" that may have resulted from shortening the particle سوف "swf" and then using the shortened version as a prefix, dialectal future particles may have arisen using a similar process, where the Levantine future particle "H" is a shortened version of the word راح "rAH" 'he will' (Jarad, 2014b; Persson, 2008b).
- Dialects routinely employ word merging, particularly when two identical letters appear consecutively. In MSA, this is mostly restricted to the case of the preposition ل "l" (to) when followed by the determiner ال "Al" (the), where the "A" in the determiner is silent. This is far more common in dialects as in يعمل لك "yEml lk" (he does for you) ⇒ يعملك "yEmlk".
- Dialects often lengthen or shorten vowels (vowel elongation and reduction). This phenomenon infrequently appears in poetry, particularly classical Arabic poetry, but is quite common in dialects with the conversion of له "lh" (to him) to ليه "lyh".

- Dialects have mostly eliminated dual forms except with nouns, e.g. عيني "Eyny" (my two eyes) and قرشين "qr$yn" (two piasters). Consequently dual agreement markers on adjectives, relative pronouns, demonstrative adjectives, and verbs have largely disappeared. Likewise, the masculine nominative plural noun and verb suffix ون "wn" has been largely replaced with the accusative/genitive forms ين "yn" and وا "wA", respectively.

Phenomena that appear in multiple dialects, but not necessarily in MSA, provide an indication that segmented training data for one dialect may be useful in segmenting other dialects. For example, fusion of particle or function words into subsequent verbs or nouns (as in "swf yEml" ⇒ "syEml") appears in very limited cases in MSA while it is common in dialects (ex. "rAH yEml" ⇒ "HyEml" or "hyEml"; "mA" "yEml" "$y" ⇒ "myEml$"; etc.). Similarly, fusion of repeated letters or repeated letters with a silent letter in the middle as in "l+Al+byt" ⇒ "llbyt" is nearly restricted in MSA to the case of the PREP l followed by the DET Al. This is far more common in dialects as "yEml lk" ⇒ "yEmlk". Yet another phenomenon is the conversion of short vowels to long vowels, which is virtually absent in MSA, but frequent in classical Arabic and dialects as in "ly" ⇒ "lyA", as shown in Example (1). So in effect, MSA has possibly lost some of the productive functionalities of Arabic, and dialects have continued to make them.

(1) تقول ابنتي لما رأت طول رحلتي ، سفارك هذا تاركي لا أبا **ليا**
*tqwl Abnty lmA rOt Twl rHlty , sfArk hA tArky lA ObA **lyA***
My daughter said when she witnessed my long travels, your travel is leaving me without a father.

6.4.2 *Differences Between Dialects*

In this section, we show some differences between dialects that cover surface lexical and morphological features in light of our datasets. Deep lexical and morphological analysis can be applied after POS-tagging of these datasets. Differences can explain why some dialects are more difficult than others, which dialects are closer to each other, and the possible effect of cross-dialect training. The differences may also aid future work on dialect identification.

We start by comparing dialects with MSA to show how close a dialect is to MSA. We randomly selected 300 words from each dialect, and we analyzed them using the Buckwalter MSA morphological analyzer (BAMA) (Buckwalter, 2004). Table 6.2 lists the percentage of words that were analyzed, analysis precision, and analysis recall, which is the percentage of actual MSA words that BAMA was able to analyze. Results show that BAMA was most successful, in terms of coverage and precision, in analyzing GLF, while it faired the worst on MGR, in terms of coverage, and the worst on LEV, in terms of precision. Some dialectal words are incorrectly recognized as MSA by BAMA, such as كده "kdh" (like this), where BAMA analyzed it as "kd+h" (his toil). It seems that GLF is the closest to MSA and MGR is the furthest away. It also did not recognize some named-entities, like برشلونة "br$lwnp" (Barcelona), so a manual revision of all analyzed and non-analyzed words was performed. Table 6.2 also shows the percentage of correctly analyzed words and unanalyzed words for each dialect. It is noteworthy that LEV has the highest percentage of analysis errors (24%) where a dialectal word like بدي "bdy" (I want) is incorrectly recognized as MSA, and MGR comes second with a 22% analysis-error rate.

Table 6.3 shows the overlap between unique words and all words for the different dialect pairs in our datasets. As the table shows, EGY, LEV, and GLF are closer together and MGR is further away from all of them.

Dialect	Percent Analyzed	Analysis Precision	Analysis Recall
EGY	83	81	94
LEV	83	76	91
GLF	**86**	**88**	94
MGR	78	78	95

Table 6.2: Buckwalter analysis

Also, LEV is closer to both EGY and GLF than the last two to each other. We can roughly say that:

- LEV is closer to both GLF and EGY than they are to each other
- MGR has the least overlap with any other dialect

We also looked at the common words between dialects to see if they had different segmentations. Aside from two words, namely ليه "lyh" (to him, why) and بيه "byh" (with it, gentleman), that both appear in EGY and LEV, all other common words have identical segmentations. This is welcome news for the lookup scheme that we employ in which we use segmentations that are seen in training directly during testing.

Dialect pairs	Unique Overlap	All Overlap
EGY-GLF	16.1%	41.6%
EGY-LEV	18.1%	43.3%
EGY-MGR	14.3%	36.7%
GLF-LEV	17.0%	41.4%
GLF-MGR	15.9%	37.8%
LEV-MGR	16.2%	38.5%

Table 6.3: Common words across dialects

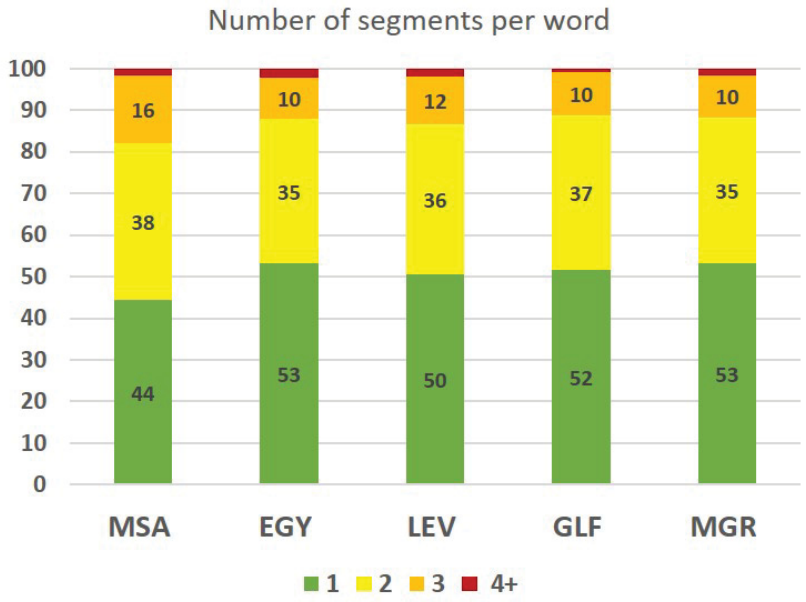

Figure 6.1: Distribution of segment count per word (percentages are overlaid on the graph)

Table 6.4 and Figure 6.1 show the distribution of segment counts per word for words in our datasets. We obtained the MSA segment counts from the Arabic Penn Treebank (parts 1-3) (Maamouri et al., 2014). The figure shows that dialectal words tend to have a similar distribution of word segment counts and they generally have fewer segments than MSA. This may indicate that dialects may have simpler segmentations than MSA, and cases where words have 4 or more segments, such as مـقلتـهاـلـهـوـش "m+qlt+hA+l+w+$" (I did not say it to him), are infrequent. It is clear from the table that words having only one segment are more frequent in DA as opposed to MSA, and this shows the general trend of simplification that DA follows. GLF has the maximum number of words with only one segment (i.e. complete words without internal segmentations), while MGR always has more words with internal segmentations than other dialects, which indicates an increasing difficulty in segmenting its words. We found that the absolute majority of words

have only one shape of segmentation regardless their contexts, and very few words have more than one segmentation depending on surrounding contexts, namely words ليه ، بيه "lyh, byh" (why, Bey), which can be also segmented as ليـه ، بيـه "ly+h, by+h" (for him, in him).

Dialect	1seg%	2seg%	3seg%	4+seg%
MSA	44.36	37.60	16.36	1.68
EGY	53.10	34.73	10.05	2.13
LEV	50.47	36.10	11.63	1.80
GLF	51.63	37.00	10.42	0.95
MGR	53.15	35.16	9.98	1.71

Table 6.4: Number of segments for each dialect

Tables 6.5 and 6.6 respectively show the number of prefixes or suffixes, the top 5 prefixes and suffixes (listed in descending order), and the unique prefixes and suffixes for each dialect in comparison to MSA. As the tables show, MGR has the greatest number of prefixes, while GLF has the most suffixes.

Dialect	No.	Top 5	Unique
MSA	8	Al,w,l,b,f	>, s
EGY	11	Al,b,w,m,h	hA, fA
LEV	11	Al,b,w,l,E	Em
GLF	14	Al,w,b,l,mA	mw,mb,$
MGR	**19**	Al,w,l,b,mA	kA,t,tA,g

Table 6.5: Prefixes statistics

Further, there are certain prefixes and suffixes that are unique to dialects. While the prefix "Al" (the) leads the list of prefixes for all dialects, the prefix ب "b" in LEV and EGY, where it is either a progressive particle or a preposition, is used more frequently than in MSA, where it is used strictly as a preposition. Similarly, the suffix كن "kn" (your) is more frequent in LEV than any other dialect. The negation suffix ش "$" (not) and feminine suffix marker

كي "ky" (your) are used in EGY, LEV, and MGR, but not in GLF or MSA. The appearance of certain affixes in some dialects and their absence in others may complicate cross dialect training, and the varying frequencies of affixes across dialects may complicate joint training.

Dialect	No.	Top 5	Unique
MSA	23	p,At,A,h,hA	hmA
EGY	24	h,p,k,$,hA	Y,kwA,nY,kY
LEV	27	p,k,y,h,w	-
GLF	**30**	h,k,y,p,t	j
MGR	24	p,w,y,k,hA	Aw

Table 6.6: Suffixes statistics

6.5 LEARNING ALGORITHMS

We present here two different systems for word segmentation. The first uses SVM-based ranking (SVM$^{\text{Rank}}$)[4] to rank different possible segmentations for a word using a variety of features. The second uses bi-LSTM-CRF, which performs character-based sequence-to-sequence mapping to predict word segmentation.

6.5.1 *SVM$^{\text{Rank}}$ Approach*

We used the SVM-based ranking approach proposed by Abdelali et al. (2016), in which they used SVM based ranking to ascertain the best segmentation for Modern Standard Arabic (MSA), which they show to be fast and of high accuracy. The approach involves generating all possible segmentations of a word

4 https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

and then ranking them. The possible segmentations are generated based on possible prefixes and suffixes that are observed during training. For example, if hypothetically we only had the prefixes و "w" (and) and ل "l" (to) and the suffix هـ "h" (his), the possible segmentations of وليده "wlydh" (his new born) would be {wlydh, w+lydh, w+l+ydh, w+l+yd+h, w+lyd+h, wlyd+h} with "wlyd+h" being the correct segmentation. SVM^{Rank} would attempt to rank the correct segmentation higher than all others. To train SVM^{Rank}, we use the following features:

- conditional probability that a leading character sequence is a prefix.
- conditional probability that a trailing character sequence is a suffix.
- probability of the prefix given the suffix.
- probability of the suffix given the prefix.
- unigram probability of the stem.
- unigram probability of the stem with first suffix.
- whether a valid stem template can be obtained from the stem, where we used Farasa (Abdelali et al., 2016) to guess the stem template.
- whether the stem that has no trailing suffixes and appears in a gazetteer of person and location names (Abdelali et al., 2016).
- whether the stem is a function word, such as على "ElY" (on) and من "mn" (from).
- whether the stem appears in the AraComLex[5] Arabic lexicon (Attia et al., 2011) or in the Buckwalter lexicon (Buckwalter, 2004). This is sensible considering the large overlap between MSA and DA.

5 http://sourceforge.net/projects/aracomlex/

- length difference from the average stem length.

The segmentations with their corresponding features are then passed to the SVM ranker (Joachims, 2006) for training. Our SVM^{Rank} uses a linear kernel and a trade-off parameter between training error and margin of 100. All segmentations are ranked out of context. Though some words may have multiple valid segmentations in different contexts, previous work on MSA has shown that it holds for 99% of the cases (Abdelali et al., 2016). This assumption allows us to improve segmentation results by looking up segmentations that were observed in the dialectal training sets (DA) or segmentations from the training sets with a back off to segmentation in a large segmented MSA corpus, namely parts 1, 2, and 3 of the Arabic Penn Treebank (Maamouri et al., 2014) (DA+MSA).

6.5.2 *Bi-LSTM-CRF Approach*

In this subsection we describe our DA segmentation model, shown in Figure 6.2. It is a simple variant of the bi-LSTM-CRF architecture first proposed by Huang, Xu, and Yu (2015), Lample et al. (2016), and Ma and Hovy (2016) among othes. The system is composed of three layers:

- Input layer: it contains character embeddings.
- Hidden layer: A BiLSTM encodes variable length input to a fixed-length vector.
- Output layer: a CRF computes the probability distribution over all labels.

CHARACTER EMBEDDINGS Character embeddings, as described in Subsection 5.5.2, have recently been proven to be an invaluable resource for many

NLP tasks. Several character aware models have been proposed to overcome some drawbacks of word embeddings (unknown words can not be properly treated) (Collobert et al., 2011; Gillick et al., 2016; Kim et al., 2016; Santos et al., 2015). In these models morphology is computed form the characters of words.

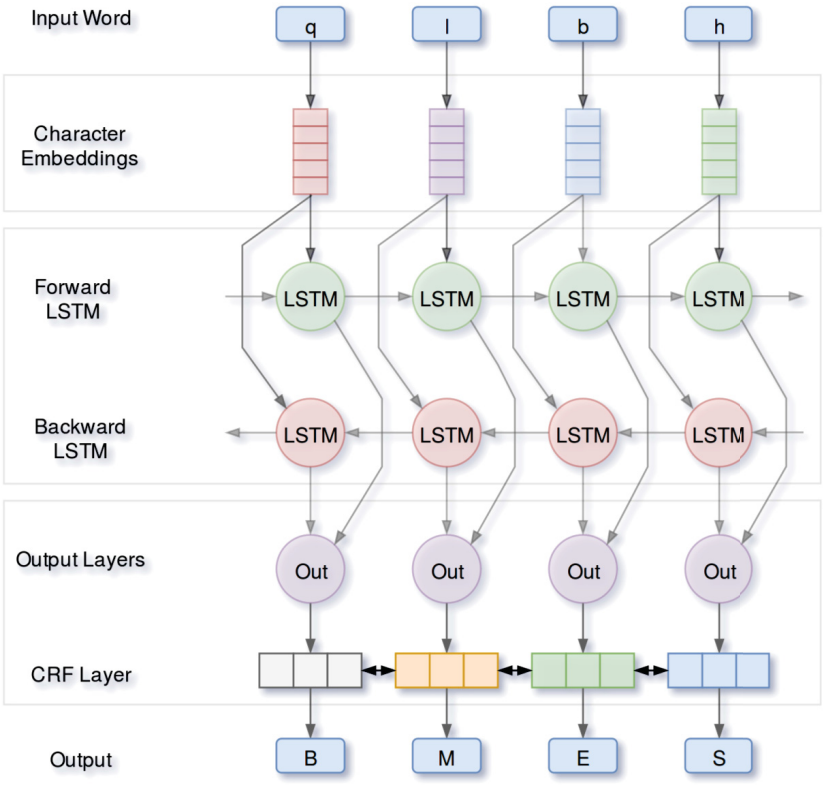

Figure 6.2: Architecture of our proposed neural network Arabic segmentation model applied to an example word. Here the model takes the word *qlbh*, "his heart" as its current input and predicts its correct segmentation. The first layer performs a look up of the characters embedding and stacks them to build a matrix. This latter is then used as the input to the BiLSTM. On the last layer, an affine transformation function followed by a CRF computes the probability distribution over all labels

CRF In many sequence-labeling tasks BiLSTMs achieve very competitive results against traditional models, however when they are used for some specific sequence classification tasks, such as segmentation and named-entity de-

tection, where there is a strict dependence between the output labels, they fail to generalize perfectly. During the training phase of the BiLSTM networks, the resulting probability distribution of each time step is independent from each other. To overcome this independence assumption imposed by the BiLSTM and to exploit this kind of labeling constraints in our Arabic segmentation system, we model label sequence logic using CRF (Lafferty, McCallum, and Pereira, 2001)

6.5.2.1 *DA segmentation Model*

The concept we followed in bi-LSTM-CRF sequence-labeling is that segmentation is a one-to-one mapping at the character level where each character is annotated as either beginning a segment (B), continues a previous segment (M), ends a segment (E), or is a segment by itself (S). After the labeling is complete we merge the characters and labels. For example, بيقولوا "byqwlwA" (they say) is labeled as "SBMMEBE", which means that the word is segmented as b+yqwl+wA. The architecture of our segmentation model, shown in Figure 6.2, is straightforward. At the input layer a look-up table is initialized with random uniformly sampled embeddings mapping each character in the input to a d-dimensional vector. At the hidden layer, the output from the character embeddings is used as the input to the BiLSTM layer to obtain fixed-dimensional representations of characters. At the output layer, a CRF is applied on the top of BiLSTM to jointly decode labels for the whole input characters. Training is performed using Stochatic Gradient Descent (SGD) descent with momentum 0.9 and batch size 50, optimizing the cross entropy objective function.

OPTIMIZATION To mitigate overfitting, given the small amount of the training data, we employ dropout (Hinton et al., 2012), which prevents co-

adaptation of hidden units by randomly setting to zero a proportion of the hidden units during training. We also employ early stopping (Caruana, Lawrence, and Giles, 2000; Graves, Mohamed, and Hinton, 2013) by monitoring the model's performance on a development set.

HYPER-PARAMETERS TUNING Table 6.7 provides a summary of the hyper-parameters we use in all our experiments. These hyper-parameters are tuned on the development dataset.

Layer	Hyper-Parameters	Value
Characters BiLSTM	state size	150
	initial state	0.0
Dropout	dropoutrate	0.5
Characters embeddings	embedding dimension	50
	batch size	50
	learning rate	0.01
	decay rate	0.05

Table 6.7: Hyper-Parameters Tuning

6.6 EXPERIMENTAL SETUP AND RESULTS

Using the approaches described earlier, we perform several experiments, serving two main objectives. First we want to see how closely related the dialects are and whether we can use one dialect for the augmentation of training data in another dialect. The second objective is to find out whether we can build a one-fits-all model that does not need to know which specific dialect it is dealing with.

In the first set of experiments shown in Table 6.8, we build segmentation models for each dialect and test them on all the other dialects. We compare this cross-dialect training and testing to training and testing on the same

	Test Set							
Farasa	85.7		82.6		82.9		82.6	
Training	EGY		LEV		GLF		MGR	
	SVM	LSTM	SVM	LSTM	SVM	LSTM	SVM	LSTM
with no lookup								
EGY	91.0	93.8	87.7	87.1	86.5	85.8	81.3	82.5
LEV	85.2	85.5	87.8	91.0	85.5	85.7	83.42	80.0
GLF	85.7	85.0	86.4	86.9	87.7	89.4	82.6	81.6
MGR	85.0	78.6	85.7	78.8	84.5	78.4	84.7	87.1
with DA lookup								
EGY	94.5	94.2	89.2	87.6	87.5	86.5	81.5	82.8
LEV	89.7	85.9	92.9	91.8	89.6	86.3	83.5	80.4
GLF	89.7	85.5	89.2	87.5	92.8	90.8	83.0	82.4
MGR	88.6	78.9	86.9	78.8	87.3	79.0	90.5	88.5
with DA+MSA lookup								
EGY	94.6	**95.0**	90.5	89.2	88.8	88.3	83.5	89.2
LEV	90.1	87.5	**93.3**	93.0	89.7	87.8	84.3	82.4
GLF	90.3	87.3	89.6	88.6	**93.1**	91.9	84.1	84.8
MGR	88.6	81.2	88.1	80.3	88.1	80.7	**91.2**	90.1

Table 6.8: Cross dialect results.

dialect, where we use 5-fold cross-validation with 70/10/20 train/dev/test splits. We also use the Farasa MSA segmenter as a baseline. We conduct the experiments at three levels: pure system output (without lookup), with DA lookup, and with DA+MSA lookup. By "lookup" we mean a post-processing add-on step where we feed segmentation solutions in the test files directly from the training data when a match is found. This is based on the assumption that segmentation is a context-free problem and therefore the utilization of observed data can be maximized. Using both algorithms (SVM and bi-LSTM-CRF) the results show a general trend where EGY segmentation yields better results from the LEV model than from the GLF's. The GLF data benefits more from the LEV model than from the EGY one. For the LEV data both GLF and EGY models are equally good. MGR seems relatively distant in that

	Test Set							
Lookup	EGY		LEV		GLF		MGR	
	SVM	LSTM	SVM	LSTM	SVM	LSTM	SVM	LSTM
No lookup	91.4	94.1	89.8	92.4	88.8	91.7	83.82	89.1
DA	94.1	94.8	92.8	93.3	91.8	92.6	89.6	90.7
DA+MSA	94.3	**95.3**	93.0	**93.9**	92.2	**93.1**	90.0	**91.4**
Joint with restricted affixes								
DA	94.5	-	92.8	-	91.9	-	89.7	-
DA+MSA	94.8	-	93.0	-	92.4	-	90.3	-

Table 6.9: Joint model results.

it does not contribute to or benefit from other dialects independently. This shows a trend where dialects favor geographical proximity. In the case with no lookup, bi-LSTM-CRF fairs better than SVM when training and testing is done on the same dialect. However, the opposite is true when we train on one dialect and test on another. This may indicate that the SVM-ranker has better cross-dialect generalization than the bi-LSTM-CRF sequence labeler. When lookup is used, SVM yields better results across the board except in three cases, namely when training and testing on Egyptian with DA+MSA lookup, when training with Egyptian and testing on MGR, and when training with GLF and testing on MGR with DA+MSA lookup. Lastly, the best SVM cross-dialect results with lookup consistently beat the Farasa MSA baseline often by several percentage points for every dialect. The same is true for bi-LSTM-CRF when training with relatively related dialects (EGY, LEV, and GLF), but the performance decreases in the case of training or testing using MGR.

In the second set of experiments, we wanted to see whether we can train a unified segmenter that would segment all the dialects in our datasets. For the results shown in Table 6.9, we also used 5-fold cross-validation (with the same splits generated earlier) where we trained on the combined training splits from all dialects and tested on all the test splits with no lookup, DA lookup, and MSA+DA lookup. We refer to these models as "joint" mod-

els. Using SVM, the accuracy of the combined model drops by 0.3% to 1.3% compared to exclusively using matching dialectal training data. We also conducted another SVM experiment in which we use the joint model in conjunction with a dialect identification oracle to restrict possible affixes only to those that are possible for that dialect (last two row in Table 6.9). The results show improvements for all dialects, but aside for EGY, the improvements do not lead to better results than those for single dialect models. Conversely, the bi-LSTM-CRF joint model with DA+MSA lookup beats every other experimental setup that we tested, leading to the best segmentation results for all dialects, without doing dialect identification. This may indicate that bi-LSTM-CRF benefited from cross-dialect data in improving segmentation for individual dialects.

6.7 CONCLUSION

This chapter presents (to the best of our knowledge) the first computational comparative study between closely related languages with regard to their automatic segmentation. Arabic dialects diverged from a single origin, yet they maintained pan-dialectal common features which allow them to cross-fertilize. Our results show that a single joint segmentation model, based on bi-LSTM-CRF, can be developed for a group of dialects and this model yields results that are comparable to, or even superior to, the performance of single dialect-specific models. Our results also show that there is a degree of closeness between dialects that is contingent with the geographical proximity. For example, we statistically show that Gulf is closer to Levantine than to Egyptian, and similarly Levantine is closer to Egyptian than to Gulf. Cross dialect segmentation experiments also show that Maghrebi is equally distant from the other three regional dialects. This sheds some light on the degree of

mutual intelligibility between the speakers of Arabic dialects, assuming that the level of success in inter-dialectal segmentation can be an indicator of how well speakers of the respective dialects can understand each other.

7

CONCLUSION AND FUTURE WORK

7.1 SUMMARY

This dissertation has presented several major contributions to the applications of neural network models to the Natural Language Processing (NLP) of Dialectal Arabic (DA) used in Social Media platforms. Chapter 1 and Chapter 3 motivate the use of DNNs as highly efective techniques for processing DA in Social Media. Chapter 4 presents two manually annotated corpora:

- The Moroccan Arabic Darija code-switching corpus
- The Twitter multi-dialectal Arabic segmentation corpus

The *Moroccan Arabic Darija code-switching corpus* is a corpus targeted at training and evaluating code-switching identification models that is by far the largest of its kind. It will be of use for supporting research in the linguistics and NLP and will constitute an ideal data source for DA processing. Similarly, the *Twitter multi-dialectal Arabic segmentation corpus* is a corpus collected from Twitter with token-level annotations. It is well suited for the training and evaluation of low-bias machine learning models like DNNs and will constitute a major benchmark for research in automatic DA morphological segmentation. Chapters 5 and 6 show that existing Deep Neural Network (DNN) architectures are capable of processing DA despite its pervasive nature. In two challenging tasks, namely morphological segmentation and CS identifi-

cation, DNNs have been shown to generalize well on unseen data and to deal with outlying, missing, unstructured, and noisy data.

7.2 FUTURE WORK

Chapters 5 and 6 have demonstrated that DNN models are effective at processing DA textual data in Social Media. They can assist with specific problems related to learning from large amounts of unsupervised DA data and they have the ability to learn data representations (features) in a greedy layer-wise fashion (Bengio et al., 2007). However there are many difficult questions that remain about the the nature of these representations and there is more work to do to better understand them:

- Do these representations correspond in any interpretable way to linguistically motivated representations typically used in theoretical linguistics?
- What are the criteria that make one representation better than another?

Unfortunately, no theory of computational linguistics or formal linguistics has given any clear insights on these longer-term questions yet. Since producing substantial human-interpretable results remains elusive, it is imperative to direct research to building tools for neural network analysis and visualisation. This will definitely enhance our understanding about the structure of these representations.

Part V

APPENDIX

A

SAWT

A.1 INTRODUCTION

Within a project concerned with the processing of code-switched data of an under-resourced Arabic dialect, Moroccan *Darija*, a large code-switched corpus had to be annotated token-wise with the extended label set from the EMNLP 2014 Shared Task on Code-Switching (Samih and Maier, 2016a; Solorio et al., 2014b). The label set contains three labels that mark MSA and DA tokens, as well as tokens in another language (English, French, Spanish, Berber). Furthermore, it contains labels for tokens which mix two languages (e.g., for French words to which Arabic morphology is applied), for ambiguous words, for Named Entities, and for remaining material (such as punctuation).

The annotation software tool had to fulfill the following requirements.

- It should excel at sequence annotation and not do anything else, i.e., "featuritis" should be avoided, furthermore it should be as simple as possible to use for the annotators, allowing for a high annotation speed;
- It should not be bound to a particular label set, since within the project, not only code-switching annotation, but also the annotation of Part-of-Speech was envisaged;
- It should allow for post-editing of tokenization during the annotation;

- It should be web-based, due to the annotators being at different physical locations;
- On client side, it should be platform-independent and run in modern browsers including browsers on mobile devices, using modern technologies such as Bootstrap[1] which provide a responsive design, without requiring a local installation of software;
- On server side, there should be safe storage; furthermore, the administration overhead should be kept minimal and there should only be minimal software requirements for the server side.

Even though several annotation interfaces for similar tasks have been presented, such as COLANN (Benajiba and Diab, 2010), COLABA (Diab et al., 2010b)DIWAN (Al-Shargi and Rambow, 2015), they were either not available or did not match our needs.

We therefore built SAWT. SAWT has been successfully used to create a code-switched corpus of 223k tokens with three annotators (Samih and Maier, 2016a). It is currently used for Part-of-Speech annotation of Moroccan Arabic dialect data. The remainder of this chapter is structured as follows. In section A.2 we present the different aspects of SWAT, namely, its data storage model, its server side structure and its client side structure. In section A.3, we review related work, and in section A.4, we conclude the chapter.

A.2 SAWT

SAWT is a web application. Its client side is machine and platform independent and runs in any modern browser. On the server side, only a PHP-

1 http://getbootstrap.com

enabled web server (ideally Apache HTTP server) and a MySQL database instance are needed.

We now describe our strategy for data storage, as well as the server side and the client side of SAWT.

A.2.1 *Data Storage*

Data storage relies on a MySQL database. One table in the database is used to store the annotator accounts. At present, there is no separate admin role, all users are annotators and cannot see or modify what the other annotators are doing.

The annotation of a text with a label set by a given user requires two MySQL tables. One table contains the actual text which is to be annotated by the user, and the other table receives the annotation; this table pair is associated with the user account which is stored in the user table mentioned above. In the first table, we store one document per row. We use the first column for a unique ID; the text is put in the second column. It is white-space tokenized at the moment it is loaded into the annotation interface (see below). In the second table, we store the annotation. Again, the document ID is put into the first column. The labels are stored in the remaining columns (one column per label).

A.2.2 *Server Side and Administration*

The complete code of SAWT will be distributed on github. The distribution will contain the complete web application code, as well as two Python scripts to be used for configuration.

The first script configures the SAWT installation and the database. It takes a configuration file as parameter (the distribution will contain a configuration file template), in which the following parameters must be specified:

- *List of tags*: A space-separated list of tags to be used in the annotation. From this list, the PHP code for the model and the view are generated which handle the final form in the interface. The generated code is then copied to the correct locations within the complete web application code.
- *Server information*: MySQL server IP, port and user account information.
- *Predictor*: The interface can show suggestions for each token, provided that a suitable sequence labeling software with a pre-trained model runs on the web server. If suggestions are desired, then in the configuration file, the corresponding path and parameters for the software must be given. If the parameter is left blank, no suggestions are shown.
- *Search box activation*: A boolean parameter indicating if a search box is desired. In the search box, the annotator can look up his previous annotations for a certain token.
- *Utility links*: The top border of the user interface consists of a link bar, the links for which can be freely configured. In our project, e.g., they are used for linking to the list of Universal POS tags (Petrov, Das, and McDonald, 2012), to a list of Arabic function words, to an Arabic Morphological Analyzer (MADAMIRA) (Pasha et al., 2014a), and to an Arabic screen keyboard, as can be seen in figures A.2 and A.3.

Once the configuration script has been run, the web application code must be copied to a suitable place within a web server installation.

In order to upload a text which is to be annotated by a certain user, the second script must be used. It takes the following command line parameters.

- *Input data*: The name of the file containing the data to be annotated. The text must be pre-tokenized (one space between each token), and there must be one document per line.
- *Server information*: MySQL server IP, port, and user account information.
- *Annotator information*: Annotator user name. If the annotator account does not exist in the respective database table, it is created, and a password must be specified.

Of course, this script can be used any number of times. At runtime, it will connect to the database and create two tables for the annotation (as mentioned above, one for the data itself and one for the annotation). It will insert the data in the first one, and insert the user account in the user account table, if necessary.

In general, for security reasons, two different servers should be used for front-end (web application) and back-end (database), but in principle, nothing stands in the way of installing everything on a single machine or even locally.

A.2.3 *Client Side and Annotator Interface*

The client side interface is written with several technologies. As a basis, we have used a MVC PHP framework, namely CodeIgniter version 3.0.[2] Furthermore, in order to achieve a responsive mobile-ready design, we have employed to the Bootstrap framework, HTML 5, and JQuery.[3]

2 http://codeigniter.net
3 http://jquery.com

When accessing the URL where SAWT is located, the annotator is queried its user name and password. After logging in, the annotation interface is shown. On top of the page, a link bar makes available several tools which are useful for the annotation, to be freely configured during installation (see above). If configured (see above), a search box is shown, in which the annotator can look up his previous annotations of a token. In a top line above the text, the ID of the document is shown, the number of tokens to be annotated, and the annotation progress, i.e., the number of tokens which have already been annotated (in previous documents). Also it is shown if the current document itself has already been annotated. Finally, there are buttons to navigate within the documents (first, previous, next, last).

For the annotation, the interface pulls the first document to be annotated from the database, applies white-space tokenization, and renders it for presentation to the user. The material to be annotated is presented with one document per page and token per line. Each line has four columns, the first one showing the absolute token ID within the complete corpus, the second one showing the token to be annotated, the third one showing a prediction of a possible tag (if configured), and the fourth one showing the possible labels. There is an edit facility, in which the annotator can correct an erroneous tokenization of the document. If an edit is performed, the modified document is white-space tokenized again and reloaded in the interface.

For label selection, we offer check-boxes. Even though radio buttons would seem to be the more natural choice, check-boxes allow us to assign several tags to a single token. This is, e.g., essential for Part-of-Speech annotation in Arabic: Due to a rich morphology, a single word can incorporate several POS functions (Habash, 2010) he user has finished the annotation of a document, a button must be clicked. This button first triggers a validation function which checks the annotation for completeness. If there are tokens which have not

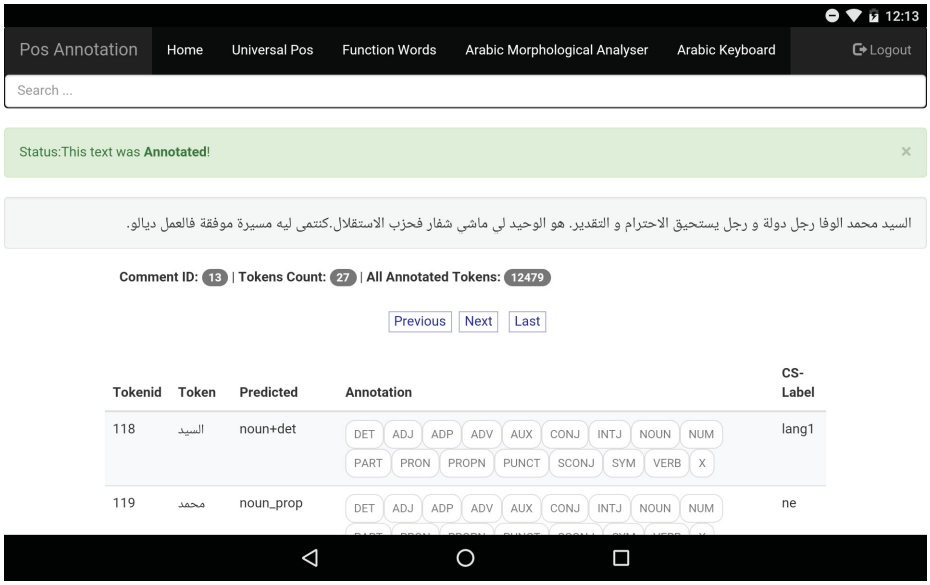

Figure A.1: Screenshot of SAWT: Annotation on Android device

been annotated, a colored alert bar is shown. Otherwise, a form is submitted which saves the annotation in the database; then the next document is loaded and rendered for annotation. We have implemented the policy that an annotator cannot change the annotation of a document once it is submitted. However, a minimal change in the code could allow a post-editing of the annotation.

We have tested the interface extensively in Google Chrome (on both PC and Android) and Mozilla Firefox.

As an example, figure A.2 shows a screenshot of the annotator interface configured for Part-of-Speech annotation with the Google Universal Part-of-Speech tag set (Petrov, Das, and McDonald, 2012). Figure A.3 shows a screenshot of code-switching annotation done in the context of our earlier work (Samih and Maier, 2016a). Finally, figure A.1 shows a screenshot of the POS annotation interface used on the Asus Nexus 7 2013 tablet running Google Chrome on Android 6.

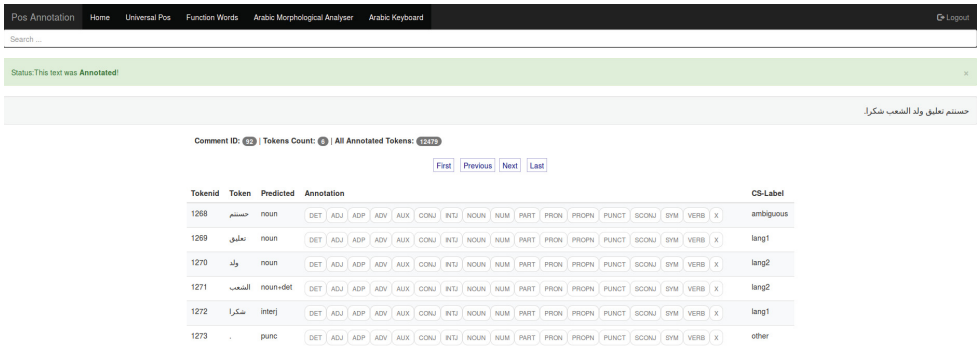

Figure A.2: Screenshot of SAWT: Annotation with Universal Part Of Speech tags

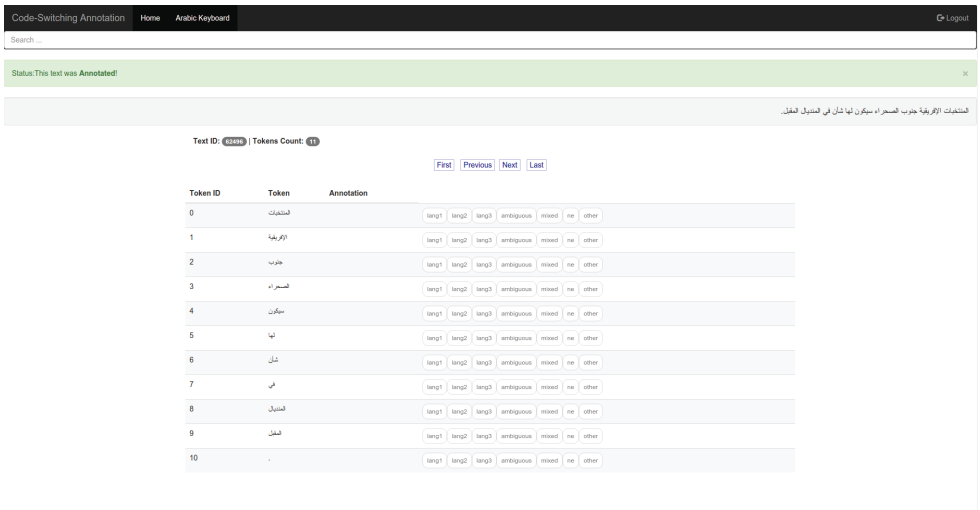

Figure A.3: Screenshot of SAWT: Annotation with code-switching labels

A.3 RELATED WORK

As mentioned above, we are not aware of a software which would have fulfilled our needs exactly. Previously released annotation software can be grouped into several categories.

Systems such as GATE Cunningham et al., 2002, CLaRK (Simov et al., 2003) and MMAX2 (Müller and Strube, 2006) are desktop-based software. They offer a large range of functions, and are in general oriented towards more complex annotation tasks, such as syntactic treebank annotation.

In the context of Arabic dialect annotation, several systems have been created. COLANN_GUI (Benajiba and Diab, 2010), which unfortunately was not available to us, is a web application that specialized on dialect annotation. DIWAN (Al-Shargi and Rambow, 2015) is a desktop application for dialect annotation which can be used online.

The systems that came closest to our needs were WebANNO (Yimam et al., 2013) and BRAT Stenetorp et al., 2012. Both are web-based and built with modern technologies. They allow for a multi-layered annotation, including a token-wise annotation. However, we decided against them due to fact that we just needed the token-wise annotation and we wanted the simplest annotator interface possible. For just sequence annotation, our annotator interface allows for a very high speed, since only one click per token is required.

A.4 CONCLUSION

We have presented SAWT, a web-based tool for sequence annotation. The main priorities of the tool are ease of use on the client side and a low requirements for the server side.

SAWT is under active development. We are currently simplifying the installation process on server side and plan to offer an admin role in the front-end. Furthermore, we want to provide a way of obtaining the annotation in a standardized format (TEI)[4] directly from the database.

4 http://www.tei-c.org/index.xml

B

CONDITIONAL RANDOM FIELDS

This chapter describes the basic framework needed for NLP tasks. It briefly introduces Conditional Random Field (CRF).

B.1 CONDITIONAL RANDOM FIELDS

CRF, a sequence labeling algorithm, predicts labels for a whole sequence rather than for the parts in isolation as shown in Equation B.1. Here, s_1 to s_m represent the labels of tokens x_1 to x_m respectively, where m is the number of tokens in a given sequence. After we have this probability value for every possible combination of labels, the actual sequence of labels for this set of tokens will be the one with the highest probability.

$$p(s_1 \dots s_m | x_1 \dots x_m) \tag{B.1}$$

$$p(\vec{s}|\vec{x};\vec{w}) = \frac{exp(\vec{w}.\vec{\Phi}(\vec{x},\vec{s}))}{\sum_{\vec{s'}\epsilon S^m} exp(\vec{w}.\vec{\Phi}(\vec{x},\vec{s'}))} \tag{B.2}$$

Equation B.2 shows the formula for calculating the probability value from Equation B.1. Here, S is the set of labels. $\vec{w}$ is the weight vector for weighting the feature vector $\vec{\Phi}$. Training and decoding are performed by the Viterbi algorithm.

C

BUCKWALTER ARABIC TRANSLITERATION CHART

C.1 CONSONANTS

Buckwalter	Glyph	Buckwalter	Glyph
’	ء	|	آ
>	أ	&	ؤ
<	إ	}	ئ
A	ا	b	ب
p	ة	t	ت
v	ث	j	ج
H	ح	x	خ
d	د	*	ذ
r	ر	z	ز
$	ش	S	ص
D	ض	T	ط
Z	ظ	E	ع
g	غ	_	ـ
f	ف	q	ق
k	ك	l	ل
m	م	n	ن
n	ن	h	ه
w	و	Y	ى
y	ي		

Table C.1: Buckwalter Transliteration System

BIBLIOGRAPHY

Abdelali, Ahmed, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak (2016). "Farasa: A fast and furious segmenter for arabic." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, San Diego, California, pp. 11–16.

Adel, Heike, Ngoc Thang Vu, and Tanja Schultz (2013). "Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling." In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, pp. 206–211.

Al-Badrashiny, Mohamed and Mona T Diab (2016). "LILI: A Simple Language Independent Approach for Language Identification." In: *COLING*, pp. 1211–1219.

Al-Badrashiny, Mohamed, Heba Elfardy, and Mona Diab (2015). "AIDA2: A Hybrid Approach for Token and Sentence Level Dialect Identification in Arabic." In: *Proc. CoNLL*.

Al-Shargi, Faisal and Owen Rambow (2015). "DIWAN: A Dialectal Word Annotation Tool for Arabic." In: *Proceedings of the Second Workshop on Arabic Natural Language Processing*. Beijing, China: Association for Computational Linguistics, pp. 49–58.

Albirini, Abdulkafi (2016). *Modern Arabic Sociolinguistics: Diglossia, variation, codeswitching, attitudes and identity*. Routledge.

Almeman, Khalid and Mark Lee (2012). "Towards developing a multi-dialect morphological analyser for arabic." In: *4th international conference on arabic language processing, rabat, morocco*.

Attia, Mohammed, Pavel Pecina, Antonio Toral, Lamia Tounsi, and Josef van Genabith (2011). "An open-source finite state morphological transducer for modern standard Arabic." In: *Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*. Association for Computational Linguistics, pp. 125–133.

Attia, Mohammed, Pavel Pecina, Younes Samih, Khaled Shaalan, and Josef van Genabith (2015). "Arabic spelling error detection and correction." In: *Natural Language Engineering*.

Bacatan, Arianna Clarisse, Bryan Loren Castillo, Marjorie Janelle Majan, Verlia Palermo, and Ria Sagum (2014). "Detection of Intra-Sentential Code-Switching Points Using Word Bigram and Unigram Frequency Count." In: *International Journal of Computer and Communication Engineering* 3.3, pp. 184–188.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate." In: *arXiv preprint arXiv:1409.0473*.

Bakr, Hitham Abo, Khaled Shaalan, and Ibrahim Ziedan (2008). "A hybrid approach for converting written Egyptian colloquial dialect into diacritized Arabic." In:

Bar, Kfir and Nachum Dershowitz (2014). "The Tel Aviv University System for the Code-Switching Workshop Shared Task." In: *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Doha, Qatar: Association for Computational Linguistics, pp. 139–143.

Barman, Utsab, Amitava Das, Joachim Wagner, and Jennifer Foster (2014). "Code Mixing: A Challenge for Language Identification in the Language

of Social Media." In: *Proceedings of The First Workshop on Computational Approaches to Code Switching*. Doha, Qatar, pp. 13–23.

Barredo, I. Muñoa (2003). "Pragmatic functions of code-switching among Basque-Spanish bilinguals." In: *Comunidades e individuos bilingües: Actas do I Simposio Internacional sobre o Bilingüismo*. Universidade de Vigo: Servizo de Publicacións.

Bassiouney, R. (2009). *Arabic Sociolinguistics*. Edinburgh University Press Series. Edinburgh University Press. ISBN: 9780748623730.

Beesley, K.R. and L. Karttunen (2003). *Finite State Morphology*. CSLI studies in computational linguistics: Center for the Study of Language and Information Bd. 1. CSLI Publications. ISBN: 9781575864341.

Benajiba, Yassine and Mona Diab (2010). "A web application for dialectal Arabic text annotation." In: *Proceedings of the LREC Workshop for Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages: Status, Updates, and Prospects*. Valletta, Malta: ELRA.

Benajiba, Yassine and Paolo Rosso (2008). "Arabic Named Entity Recognition using Conditional Random Fields." In: *Proc. Workshop on HLT & NLP within the Arabic World at LREC 2008*.

Bengio, Yoshua, Olivier Delalleau, and Nicolas L. Roux (2006). "The Curse of Highly Variable Functions for Local Kernel Machines." In: *Advances in Neural Information Processing Systems 18*. Ed. by Y. Weiss, P. B. Schölkopf, and J. C. Platt. MIT Press, pp. 107–114.

Bengio, Yoshua, Patrice Simard, and Paolo Frasconi (1994). "Learning long-term dependencies with gradient descent is difficult." In: *IEEE transactions on neural networks* 5.2, pp. 157–166.

Bengio, Yoshua, Pascal Lamblin, Dan Popovici, and Hugo Larochelle (2007). "Greedy layer-wise training of deep networks." In: *Advances in neural information processing systems*, pp. 153–160.

Bengio, Yoshua et al. (2009). "Learning deep architectures for AI." In: *Foundations and trends® in Machine Learning* 2.1, pp. 1–127.

Benmamoun, Elabbas (2001). "Language identities in Morocco: A historical overview." In: *Studies in the Linguistic Sciences* 31.1, pp. 95–106.

Berk-Seligson, S. (1986). *Linguistic Constraints on Intrasentential Code Switching: A Study of Spanish-Hebrew Bilingualism*. Cambridge University Press.

Biadsy, Fadi, Julia Hirschberg, and Nizar Habash (2009). "Spoken Arabic Dialect Identification Using Phonotactic Modeling." In: *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*. Semitic '09. Athens, Greece: Association for Computational Linguistics, pp. 53–61.

Bouamor, Houda, Nizar Habash, and Kemal Oflazer (2014). "A Multidialectal Parallel Corpus of Arabic." In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Ed. by Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Reykjavik, Iceland: European Language Resources Association (ELRA). ISBN: 978-2-9517408-8-4.

Brustad, K. (2000). *The Syntax of Spoken Arabic: A Comparative Study of Moroccan, Egyptian, Syrian, and Kuwaiti Dialects*. Georgetown University Press. ISBN: 9780878407897.

Buckwalter, Tim (2001). *Arabic Transliteration. Retrieved June 19, 2016*.

Buckwalter, Tim (2004). "Issues in Arabic orthography and morphology analysis." In: *proceedings of the workshop on computational approaches to Arabic script-based languages*. Association for Computational Linguistics, pp. 31–34.

Bullock, Barbara E. and Almeida Jacqueline Toribio (2009). *The Cambridge handbook of linguistic code-switching*. Cambridge University Press.

Caruana, Rich, Steve Lawrence, and Lee Giles (2000). "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping." In: *NIPS*, pp. 402–408.

Çetinoglu, Özlem (2016). "A Turkish-German Code-Switching Corpus." In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Portorož, Slovenia.

Chang, Joseph Chee and Chu-Cheng Lin (2014a). "Recurrent-Neural-Network for Language Detection on Twitter Code-Switching Corpus." In: *arXiv preprint arXiv:1412.4314*.

Chang, Joseph Chee and Chu-Cheng Lin (2014b). "Recurrent-Neural-Network for Language Detection on Twitter Code-Switching Corpus." In: *arXiv:1412.4314v2*.

Chen, Danqi and Christopher Manning (2014). "A fast and accurate dependency parser using neural networks." In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa (2011). "Natural Language Processing (Almost) from Scratch." In: *Journal of Machine Learning Research* 12, pp. 2493–2537.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks." In: *Machine Learning* 20.3, pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018.

Cotterell, Ryan, Adithya Renduchintala, Naomi Saphra, and Chris Callison-Burch (2014). "An Algerian Arabic-French Code-Switched Corpus." In: *LREC Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools*. Reykjavik, Iceland.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, and Valentin Tablan (2002). "GATE: an architecture for development of robust HLT applications." In: *Proceedings of the 40th annual meeting on association for com-*

putational linguistics. Association for Computational Linguistics, pp. 168–175.

Darwish, Kareem, Hassan Sajjad, and Hamdy Mubarak (2014). "Verifiably Effective Arabic Dialect Identification." In: *EMNLP*, pp. 1465–1468.

Das, Amitava and Björn Gambäck (2014). "Identifying Languages at the Word Level in Code-Mixed Indian Social Media Text." In: *In Proceedings of the 11th International Conference on Natural Language Processing*. Goa, India, pp. 169–178.

Deng, Li (2014). "A tutorial survey of architectures, algorithms, and applications for deep learning." In: *APSIPA Transactions on Signal and Information Processing* 3.

Dey, Anik and Pascale Fung (2014). "A Hindi-English Code-Switching Corpus." In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA).

Diab, Mona, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba (2010a). "COLABA: Arabic dialect annotation and processing." In: *Proceedings of the LREC Workshop for Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages: Status, Updates, and Prospects*, pp. 66–74.

Diab, Mona, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba (2010b). "COLABA: Arabic dialect annotation and processing." In: *Proceedings of the LREC Workshop for Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages: Status, Updates, and Prospects*, pp. 66–74.

Diab, Mona, Julia Hirschberg, Pascale Fung, and Thamar Solorio, eds. (2014). *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Doha, Qatar.

Eldesouki, Mohamed, Fahim Dalvi, Hassan Sajjad, and Kareem Darwish (2016). "QCRI@ DSL 2016: Spoken Arabic Dialect Identification Using Textual." In: *VarDial 3*, p. 221.

Eldesouki, Mohamed, Younes Samih, Ahmed Abdelali, Mohammed Attia, Hamdy Mubarak, Kareem Darwish, and Laura Kallmeyer (2017). "Arabic Multi-Dialect Segmentation: bi-LSTM-CRF vs. SVM." In: *CoRR* abs/1708.05891.

Elfardy, Heba, Mohamed Al-Badrashiny, and Mona Diab (2013). "Code switch point detection in Arabic." In: *Proc. NLDB*.

Elfardy, Heba, Mohamed Al-Badrashiny, and Mona Diab (2014a). "AIDA: Identifying Code Switching in Informal Arabic Text." In: *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Doha, Qatar, pp. 94–101.

Elfardy, Heba, Mohamed Al-Badrashiny, and Mona Diab (2014b). "AIDA: Identifying Code Switching in Informal Arabic Text." In: *Proc. CS Workshop at EMNLP*.

Elfardy, Heba and Mona Diab (2012a). "Simplified guidelines for the creation of Large Scale Dialectal Arabic Annotations." In: *Proc. LREC*.

Elfardy, Heba and Mona Diab (2012b). "Token Level Identification of Linguistic Code Switching." In: *Proceedings of COLING 2012: Posters*. Mumbai, India, pp. 287–296.

Elinson, Alexander E (2013). "DĀRIJA AND CHANGING WRITING PRACTICES IN MOROCCO." In: *International Journal of Middle East Studies* 45.04, pp. 715–730.

Eskander, Ramy, Nizar Habash, Owen Rambow, and Nadi Tomeh (2013). "Processing Spontaneous Orthography." In:

Estigarribia, Bruno (2015). "Guaraní-Spanish Jopara Mixing in a Paraguayan Novel – Does it Reflect a Third Language, a Language Variety, or True Codeswitching?" In: *Journal of Language Contact* 8.2, pp. 182–222.

Farzindar, Atefeh and Diana Inkpen (2015). "Natural language processing for social media." In: *Synthesis Lectures on Human Language Technologies* 8.2, pp. 1–166.

Ferguson, Charles (1959). "Diglossia." In: *Word* 15, pp. 325–340.

Gillick, Dan, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya (2015). "Multilingual language processing from bytes." In: *arXiv preprint arXiv:1512.00103*.

Gillick, Dan, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya (2016). "Multilingual Language Processing From Bytes." In: *Proceedings of NAACL-HLT*, pp. 1296–1306.

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks." In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256.

Goldberg, Y. and G. Hirst (2017). *Neural Network Methods in Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers. ISBN: 9781627052955.

Goldberg, Yoav (2016). "A Primer on Neural Network Models for Natural Language Processing." In:

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

Graves, Alex (2013). "Generating sequences with recurrent neural networks." In: *arXiv preprint arXiv:1308.0850*.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton (2013). "Speech recognition with deep recurrent neural networks." In: *Acoustics, speech and*

signal processing (icassp), 2013 ieee international conference on. IEEE, pp. 6645–6649.

Graves, Alex and Jürgen Schmidhuber (2005). "Framewise phoneme classification with bidirectional LSTM and other neural network architectures." In: *Neural Networks* 18.5, pp. 602–610.

Habash, Nizar Y. (2010). "Introduction to Arabic natural language processing." In: *Synthesis Lectures on Human Language Technologies* 3.1, pp. 1–187.

Habash, Nizar, Mona T Diab, and Owen Rambow (2012). "Conventional Orthography for Dialectal Arabic." In: *LREC*, pp. 711–718.

Habash, Nizar, Ramy Eskander, and Abdelati Hawwari (2012). "A morphological analyzer for Egyptian Arabic." In: *Proceedings of the twelfth meeting of the special interest group on computational morphology and phonology*. Association for Computational Linguistics, pp. 1–9.

Habash, Nizar, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh (2013). "Morphological Analysis and Disambiguation for Dialectal Arabic." In: *Hlt-Naacl*, pp. 426–432.

Harrell, Richard S. (1962). *A Short Reference Grammar of Moroccan Arabic*. Georgetown University Press.

Harrell, Richard and Harvey Sobelman, eds. (1966). *A dictionary of Moroccan Arabic*. Georgetown University Press.

Haykin, Simon (1994). "Neural networks, A comprehensive Foundation." In:

Hetzron, R. (1997). *The Semitic Languages*. Routledge family language descriptions. Routledge. ISBN: 9780415057677.

Hinton, Geoffrey E and Ruslan R Salakhutdinov (2006). "Reducing the dimensionality of data with neural networks." In: *science* 313.5786, pp. 504–507.

Hinton, Geoffrey E, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov (2012). "Improving neural networks by preventing co-adaptation of feature detectors." In: *arXiv preprint arXiv:1207.0580*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory." In: *Neural computation* 9.8, pp. 1735–1780.

Holes, C. (2004). *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic. Georgetown University Press. ISBN: 9781589010222.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer feedforward networks are universal approximators." In: *Neural Networks* 2.5, pp. 359 –366. ISSN: 0893-6080. DOI: http://dx.doi.org/10.1016/0893-6080(89)90020-8.

Huang, Zhiheng, Wei Xu, and Kai Yu (2015). "Bidirectional LSTM-CRF Models for Sequence Tagging." In: *CoRR* abs/1508.01991.

Ibrahim, Zeinab (2006). "Borrowing in Modern Standard Arabic." In: *Innovation and Continuity in Language and Communication of Different Language Cultures 9. Edited by Rudolf Muhr*, pp. 235–260.

Jarad, Najib Ismail (2014a). "The Grammaticalization of the Motion Verb "Ra?" as a Prospective Aspect Marker in Syrian Arabic." In: *Al-'Arabiyya* 47, pp. 101–118. ISSN: 08898731, 23754036.

Jarad, Najib Ismail (2014b). "The Grammaticalization of the Motion Verb RaH as a Prospective Aspect Marker in Syrian Arabic." In: *Al-'Arabiyya* 47, pp. 101–118. ISSN: 08898731, 23754036.

Joachims, Thorsten (2006). "Training Linear SVMs in Linear Time." In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 217–226.

Joshi, Aravind K (1982). "Processing of sentences with intra-sentential code-switching." In: *Proc. COLING*.

Kim, Yoon (2014). "Convolutional neural networks for sentence classification." In: *In EMNLP*. Citeseer.

Kim, Yoon, Yacine Jernite, David Sontag, and Alexander M Rush (2016). "Character-Aware Neural Language Models." In: *Thirtieth AAAI Conference on Artificial Intelligence*.

King, Ben and Steven Abney (2013). "Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods." 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. 1110–1119.

King, Levi, Eric Baucom, Timur Gilmanov, Sandra Kübler, Dan Whyatt, Wolfgang Maier, and Paul Rodrigues (2014). "The IUCL+ System: Word-Level Language Identification via Extended Markov Models." In: *Proc. CS Workshop at EMNLP*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems*, pp. 1097–1105.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data." In: *Proc. ICML*.

Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer (2016). "Neural architectures for named entity recognition." In: *arXiv preprint arXiv:1603.01360*.

Lavergne, Thomas, Olivier Cappé, and François Yvon (2010). "Practical Very Large Scale CRFs." In: *Proc. ACL*.

Lignos, Constantine and Mitch Marcus (2013). "Toward web-scale analysis of codeswitching." In: *87th Annual Meeting of the Linguistic Society of America*.

Lipton, Zachary C, David C Kale, Charles Elkan, and Randall Wetzell (2015). "A Critical Review of Recurrent Neural Networks for Sequence Learning." In: *CoRR* abs/1506.00019.

Ma, Xuezhe and Eduard Hovy (2016). "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF." In: *Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1064–1074.

Maamouri, Mohamed, Ann Bies, Seth Kulick, Michael Ciul, Nizar Habash, and Ramy Eskander (2014). "Developing an Egyptian Arabic Treebank: Impact of Dialectal Morphology on Annotation and Tool Development." In: *LREC*, pp. 2348–2354.

Maas, Andrew L, Awni Y Hannun, and Andrew Y Ng (2013). "Rectifier nonlinearities improve neural network acoustic models." In: *Proc. ICML*. Vol. 30. 1.

Maharjan, Suraj, Elizabeth Blair, Steven Bethard, and Thamar Solorio (2015a). "Developing Language-tagged Corpora for Code-switching Tweets." In: *Proceedings of The 9th Linguistic Annotation Workshop*. Denver, Colorado, USA, pp. 72–84.

Maharjan, Suraj, Elizabeth Blair, Steven Bethard, and Thamar Solorio (2015b). "Developing Language-tagged Corpora for Code-switching Tweets." In: *Proc. LAW IX at NAACL*.

Maier, Wolfgang and Carlos Gómez-Rodríguez (2014). "Language variety identification in Spanish tweets." In: *Proc. LT4CloseLang at EMNLP*.

McCarthy, John J (1981). "A prosodic theory of nonconcatenative morphology." In: *Linguistic inquiry*, pp. 373–418.

Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur (2010). "Recurrent neural network based language model." In: *Interspeech*. Vol. 2, p. 3.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient Estimation of Word Representations in Vector Space." In: *CoRR* abs/1301.3781.

Milroy, Lesley and Pieter Muysken (1995). *One Speaker, Two Languages: Cross-Disciplinary Perspectives on Code-Switching*. Cambridge Univ. Press.

Mohamed, Emad, Behrang Mohit, and Kemal Oflazer (2012). "Annotating and Learning Morphological Segmentation of Egyptian Colloquial Arabic." In: *LREC*, pp. 873–877.

Molina, Giovanni, Nicolas Rey-Villamizar, Thamar Solorio, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, and Mona Diab (2016). "Overview for the second shared task on language identification in code-switched data." In: *EMNLP 2016*, p. 40.

Monroe, Will, Spence Green, and Christopher D Manning (2014). "Word Segmentation of Informal Arabic with Domain Adaptation." In: *ACL (2)*, pp. 206–211.

Mubarak, Hamdy and Kareem Darwish (2014). "Using Twitter to collect a multi-dialectal corpus of Arabic." In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pp. 1–7.

Müller, Christoph and Michael Strube (2006). "Multi-level annotation of linguistic data with MMAX2." In: *Corpus technology and language pedagogy: New resources, new tools, new methods* 3, pp. 197–214.

Muntendam, Antje (2006). "Diglossia, Footing and Quechua-Spanish Code-Switching." In: *Symposium about language and society*. Austin, TX.

Muysken, Pieter (2000). *Bilingual speech: A typology of code-mixing*. Vol. 11. Cambridge Univ. Press.

Myers-Scotton, Carol (1993). "Common and uncommon ground: Social and structural factors in codeswitching." In: *Language in Society* 22, pp. 475–475.

Myers-Scotton, Carol (1997). *Duelling Languages: Grammatical Structure in Codeswitching*. Clarendon Press. ISBN: 9780198237129.

Nakov, Preslav, Petya Osenova, and Cristina Vertan, eds. (2014). *Proceedings of the EMNLP'2014 Workshop on Language Technology for Closely Related Languages and Language Variants*. Doha, Qatar.

Papalexakis, Evangelos, Dong-Phuong Nguyen, and A Seza Doğruöz (2014). "Predicting Code-Switching in Multilingual Communication for Immigrant Communities." In: *Proceedings of The First Workshop on Computational Approaches to Code Switching*. Doha, Qatar, pp. 42–50.

Pasha, Arfath, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth (2014a). "MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic." In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland, pp. 1094–1101. ISBN: 978-2-9517408-8-4.

Pasha, Arfath, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M Roth (2014b). "Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic." In: *Proc. LREC*.

Pauls, Adam and Dan Klein (2011). "Faster and Smaller N-Gram Language Models." In: *Proc. ACL:HLT*.

Persson, Maria (2008a). "The Role of the b-prefix in Gulf Arabic Dialects as a Marker of Future, Intent and/or Irrealis." In: Journal of Arabic and Islamic Studies 8, pp. 26–52. ISSN: 0806-198X.

Persson, Maria (2008b). "The Role of the b-prefix in Gulf Arabic Dialects as a Marker of Future, Intent and/or Irrealis." In: Journal of Arabic and Islamic Studies 8, pp. 26–52. ISSN: 0806-198X.

Petrov, Slav, Dipanjan Das, and Ryan McDonald (2012). "A Universal Part-of-Speech Tagset." In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*. Istanbul, Turkey.

Poplack, Shana (1980). "Sometimes I'll start a sentence in Spanish y termino en español: Toward a typology of code-switching." In: *Linguistics* 18.7-8, pp. 581–618.

Rodrigues, Paul and Sandra Kübler (2013). "Part of Speech Tagging Bilingual Speech Transcripts with Intrasentential Model Switching." In: *AAAI Spring Symposium: Analyzing Microtext*.

Rosenblatt, Frank (1958). "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6, p. 386.

Roth, Ryan, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin (2008). "Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking." In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, pp. 117–120.

Rumelhart, David E, Geoffrey E Hintont, and Ronald J Williams (1986). "Learning representations by back-propagating errors." In: *NATURE* 323, p. 9.

Ryding, K.C. (2005). *A Reference Grammar of Modern Standard Arabic*. Reference Grammars. Cambridge University Press. ISBN: 9781139443333.

Salloum, Wael and Nizar Habash (2011). "Dialectal to standard Arabic paraphrasing to improve Arabic-English statistical machine translation." In: *Proceedings of the first workshop on algorithms and resources for modelling of dialects and language varieties*. Association for Computational Linguistics, pp. 10–21.

Salloum, Wael and Nizar Habash (2012). "Elissa: A Dialectal to Standard Arabic Machine Translation System." In: *Coling (demos)*, pp. 385–392.

Salloum, Wael and Nizar Habash (2014). "Adam: Analyzer for dialectal arabic morphology." In: *Journal of King Saud University-Computer and Information Sciences* 26.4, pp. 372–378.

Samih, Younes and Wolfgang Maier (2016a). "An Arabic-Moroccan Darija Code-Switched Corpus." In: *Proc. LREC*.

Samih, Younes and Wolfgang Maier (2016b). "Detecting code-switching in Moroccan Arabic." In: *Proceedings of SocialNLP @ IJCAI-2016*. New York.

Samih, Younes, Wolfgang Maier, and Laura Kallmeyer (2016). "SAWT: Sequence Annotation Web Tool." In: *EMNLP 2016*, p. 65.

Samih, Younes, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio (2016). "Multilingual Code-switching Identification via LSTM Recurrent Neural Networks." In: *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, Austin, TX, pp. 50–59.

Samih, Younes, Mohammed Attia, Mohamed Eldesouki, Hamdy Mubarak, Ahmed Abdelali, Laura Kallmeyer, and Kareem Darwish (2017a). "A Neural Architecture for Dialectal Arabic Segmentation." In: *WANLP 2017 (co-located with EACL 2017)*, p. 46.

Samih, Younes, Mohamed Eldesouki, Mohammed Attia, Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, and Laura Kallmeyer (2017b). "Learning from Relatives: Unified Dialectal Arabic Segmentation." In: *CoNLL 2017*, p. 432.

Santos, Cıcero dos, Victor Guimaraes, RJ Niterói, and Rio de Janeiro (2015). "Boosting named entity recognition with neural character embeddings." In: *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, p. 25.

Sayahi, Lotfi (2014). *Diglossia and language contact: Language variation and change in North Africa*. Cambridge University Press.

Schuster, Mike and Kuldip K Paliwal (1997). "Bidirectional recurrent neural networks." In: *IEEE Transactions on Signal Processing* 45.11, pp. 2673–2681.

Shrestha, Prajwol (2014). "Incremental N-gram Approach for Language Identification in Code-Switched Text." In: *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Doha, Qatar: Association for Computational Linguistics, pp. 133–138.

Shukla, Anupam, Ritu Tiwari, and Rahul Kala (2010). *Real life applications of soft computing*. CRC press.

Simov, Kiril, Alexander Simov, Milen Kouylekov, Krasimira Ivanova, Ilko Grigorov, and Hristo Ganev (2003). "Development of corpora within the CLaRK system: The BulTreeBank project experience." In: *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*. Association for Computational Linguistics, pp. 243–246.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts (2013). "Recursive deep models for semantic compositionality over a sentiment treebank." In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

Solorio, Thamar and Yang Liu (2008a). "Learning to Predict Code-Switching Points." In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 973–981.

Solorio, Thamar and Yang Liu (2008b). "Learning to predict code-switching points." In: *Proc. EMNLP*, pp. 973–981.

Solorio, Thamar and Yang Liu (2008b). "Part-of-speech tagging for English-Spanish code-switched text." In: *Proceedings of the Conference on Empirical*

Methods in Natural Language Processing. Association for Computational Linguistics, pp. 1051–1060.

Solorio, Thamar et al. (2014a). "Overview for the First Shared Task on Language Identification in Code-Switched Data." In: *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Doha, Qatar, pp. 62–72.

Solorio, Thamar et al. (2014b). "Overview for the First Shared Task on Language Identification in Code-Switched Data." In: *Proc. CS Workshop at EMNLP*.

Stenetorp, Pontus, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii (2012). "brat: a Web-based Tool for NLP-Assisted Text Annotation." In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France: Association for Computational Linguistics, pp. 102–107. URL: http://www.aclweb.org/anthology/E12-2021.

Tratz, Stephen, Douglas Briesch, Jamal Laoudi, and Clare Voss (2013). "Tweet Conversation Annotation Tool with a Focus on an Arabic Dialect, Moroccan Darija." In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Sofia, Bulgaria, pp. 135–139.

Volk, Martin and Simon Clematide (2014). "Detecting Code-Switching in a Multilingual Alpine Heritage Corpus." In: *Proceedings of The First Workshop on Computational Approaches to Code Switching*. Doha, Qatar, pp. 24–33.

Voss, Clare, Stephen Tratz, Jamal Laoudi, and Douglas Briesch (2014). "Finding Romanized Arabic Dialect in Code-Mixed Tweets." In: *Proc. LREC*.

Vu, Ngoc Thang and Tanja Schultz (2014). "Exploration of the Impact of Maximum Entropy in Recurrent Neural Network Language Models for Code-Switching Speech." In: *Proceedings of The First Workshop on Computational Approaches to Code Switching*. Doha, Qatar, pp. 34–41.

Wardhaugh, R. and J.M. Fuller (2014). *An Introduction to Sociolinguistics*. Blackwell Textbooks in Linguistics. Wiley. ISBN: 9781118732298.

Watson, J (2011). "Arabic dialects (general article)." In: *The Semitic Languages: An international handbook*. Ed. by S Weninger, G Khan, M Streck, and JCE Watson. Walter de Gruyter, 851 –896 (47).

Yeh, Ching-Feng, Aaron Heidel, Hong-Yi Lee, and Lin-Shan Lee (2013). "Recognition of Highly Imbalanced Code-Mixed Bilingual Speech with Frame-Level Language Detection Based on Blurred Posteriorgram." In: *IEEE* 2013.

Yimam, Seid Muhie, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann (2013). "WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations." In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1–6. URL: http://www.aclweb.org/anthology/P13-4001.

Yu, Liang-Chih, Wei-Cheng He, Wei-Nan Chien, and Yuen-Hsien Tseng (2012). "Identification of Code-Switched Sentences and Words Using Language Modeling Approaches." In: *Mathematical Problems in Engineering* 978-1-4673-0046.

Zaidan, Omar F and Chris Callison-Burch (2014). "Arabic dialect identification." In: *Computational Linguistics* 40.1, pp. 171–202.

Zampieri, Marcos, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann, eds. (2014). *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*. Dublin, Ireland.

Zbib, Rabih, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. 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. NAACL HLT '12. Montreal, Canada: Association for Computational Linguistics, pp. 49–59.

Zitouni, Imed (2014). *Natural Language Processing of Semitic Languages*. Theory and Applications of Natural Language Processing. Springer-Verlag Berlin Heidelberg. ISBN: 978-3-642-45357-1,978-3-642-45358-8.

Zribi, Inès, Mariem Ellouze Khemakhem, and Lamia Hadrich Belguith (2013). "Morphological Analysis of Tunisian Dialect." In: *IJCNLP*, pp. 992–996.

DECLARATION

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht habe und dass die Arbeit bisher in gleicher oder ähnlicher Form keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht wurde. Bereits veröffentliche Teile sind in der Arbeit gekennzeichnet.

Düsseldorf, Oktober 2017

Younes Samih