Neural Machine Translation for Arabic Language

الترجمة الآلية العصبية للغة العربية

by

MANAR ALKHATIB

A thesis submitted in fulfilment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

at

The British University in Dubai

July 2019

Neural Machine Translation for Arabic Language

العربية للغة العصبية الآلية الترجمة

by

Manar Alkhatib

A thesis submitted to the Faculty of Engineering & IT in fulfilment of the requirements for the degree of Doctor of Philosophy (PhD)

at

The British University in Dubai

July 2019

Thesis Supervisor

Professor Dr. Khaled Shaalan

Approved for award:

_____	_____
Name	Name
Designation	Designation
_____	_____
Name	Name
Designation	Designation

Date: _____

DECLARATION

I warrant that the content of this research is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

Manar Alkhatib

_____.

Signature of the student

COPYRIGHT AND INFORMATION TO USERS

Abstract

Translating the Arabic Language into other languages engenders multiple linguistic problems, as no two languages can match, either in the meaning given to the conforming symbols or in the ways in which such symbols are arranged in phrases and sentences. Lexical, syntactic and semantic problems arise when translating the meaning of Arabic words into English. Machine translation (MT) into morphologically rich languages (MRL) poses many challenges, from handling a complex and rich vocabulary, to designing adequate MT metrics that take morphology into consideration.

The task of recognizing and generating paraphrases is an essential component in many Arabic natural language processing (NLP) applications. A well-established machine translation approach for automatically extracting paraphrases, leverages bilingual corpora to find the equivalent meaning of phrases in a single language, is performed by "pivoting" over a shared translation in another language. Neural machine translation has recently become a viable alternative approach to the more widely-used statistical machine translation. In this thesis, we revisit bilingual pivoting in the context of neural machine translation and present a paraphrasing model based mainly on neural networks. The thesis we present also, highlights the key challenges for Arabic language translation into English, and Arabic. Experimental results across datasets confirm that neural paraphrases significantly outperform those obtained with statistical machine translation, and indicate high similarity correlation between our model and human translation, making our model attractive for real-world deployment.

المقدمة

حققت تكنولوجيا الشبكات العصبية في الآونة الأخيرة نتائج جيدة في مجال الترجمة الآلية، والتي تعتبر واحدة من أكثر مهام الذكاء الاصطناعي تحديًا اذا ما قورنت بالترجمة اليدوية. وهذا يبدو واضحا عند ترجمة اللغة العربية إلى لغات أخرى آليا، حيث من غير الممكن أن تتطابق لغتان حرفيا، سواء كان ذلك بالمعنى أو بتركيب الجملة، اوشبه الجملة، المنتجة آليا. كما وتظهر العديد من المشكلات عادة عند ترجمة معاني الكلمات حرفياً من اللغة العربية إلى اللغة الإنجليزية كالمشكلات المعجمية والنحوية والدلالية، إضافة الى مشكلات التصريف اللغوي عند التعامل مع المفردات المعقدة والغنية في اللغة العربية. وتعد مهمة صياغة الجمل من المكونات الأساسية في العديد من تطبيقات معالجة اللغة العربية، حيث يتم استخدام تقنيات خاصة لاستخراج المصطلحات بطريقة تلقائيّة، وذلك من خلال الاستفادة من ثنائيّة اللغة لإيجاد المعنى المكافئ للجمل في لغة واحدة بواسطة "التمحور" على ترجمة مشتركة بلغة أخرى. لذا أصبحت الترجمة الآلية العصبية مؤخراً طريقة بديلة قابلة للتطبيق في الترجمة الآلية، واصبحت أكثر استخداما وعلى نطاق واسع.

تهدف هذه الرسالة الى دراسة إمكانية استخدام الشبكات العصبية لتحسين جودة الترجمة الآلية. كما ونقدم نموذجا لإعادة صياغة الجمل العربية باستخدام الشبكات العصبية، وباستخدام لغة اخرى للحصول على الترجمة الصحيحة وتوليد معنى آخر ومبسط. كما ونسلط الضوء على التحديات الرئيسية لترجمة اللغة العربية إلى اللغة الإنجليزية، وتؤكد النتائج أن العبارات العصبية تتفوق بشكل كبير على تلك التي تم الحصول عليها من خلال الترجمة الآلية الإحصائيّة، والتي تشير إلى وجود علاقة تشابه عالية بين نموذجنا والترجمة البشرية، مما يجعل نموذجنا جذابًا للنشر في العالم الحقيقي .

Acknowledgements

I am thanking my GOD *Allah* for giving me health, patience and strength to write this thesis and all the graces he has granted to me.

I would like to thank my supervisor Prof. Khaled Shaalan for supervising me during these four years. Thank you very much for your patience, guidance and encouragement. I learnt from how to be a real researcher, how to think differently and how to understand life better.

To my best friend Dr. May ElBarachi, thank you very much for being my real friend whom I trust. Your wise advice, encouragement and unending generosity made my research life easy and enjoyable. Thank you for the discussions, sharing ideas and plans for future research. I am looking forward to producing lots of publications from our great ideas.

I dedicate this thesis to my family who have always supported me in my studies and life. Without your love, care and patience, I would not have achieved this. I would like to thank my husband Khalil for being there during the good times and the hard times. Thank you for your patience, care and everything you have done to keep me and our kids; Karim and Tala, gathered in peace and happiness. Thank you for giving us the love we need to survive in this life.

Last but not least, I would like to thank my family. Words cannot express how grateful I am to my mother, father, brothers and sister. I would also like to thank all of my friends in our who supported me and helped me with my research.

Table of Contents

List of Tables

List of Figures

Part I: Introduction and Background Review

Chapter 1 Introduction

Automatic machine translation is one of the major applications in natural language processing. It has proved both the most enticing task and the least approachable. Since its introduction in the field, many approaches have been applied, from traditional rule-based methods to the more recent statistical methods (Greenstein and Penner, 2015). Still, as anyone who has spent a few minutes on Google Translate®, an online free translator that uses statistical approach, will testify, there is still a long way to go before this problem can be considered solved in any useful fashion.

However, the efficiency of a machine translation system is deeply dependent on the language pair under consideration. While there are still certain grammatical structures to be considered, for example, metaphors that are often not translated appropriately, statistical machine translation between language pairs such as French and English is considered to have obtained acceptable accuracy to be somewhat useful in practice (Mallinson, Sennrich, and Lapata, 2017).

Neural machine translation in English (Kalchbrenner and Blunsom, 2013) has become a significant alternative to the widely used statistical machine translation system (Koehn, Och, and Marcu, 2003), evidenced by the successful findings that appeared in the WMT15 and WMT16 conferences. For Arabic language translation, however, neural machine translation (NMT) is a new machine translation approach that has not led to remarkable improvements, particularly concerning human evaluation, compared to rule-based and statistical machine translation (SMT) systems (Bentivogli and Federico, 2015).

The main goal of this thesis is to explore neural architectures which can be either used independently for machine translation (MT) purposes, as end-to-end purely neural translation engines, or embedded as complementary modules into existing translation models in order to boost their performance. Therefore, this thesis is mainly about the application of neural networks (NNs) in MT of a low-resource language. Along with the main direction of the thesis, we also focus on challenges related to the translation such as metaphors, named entities, and ambiguous word senses. We are also interested in investigating the impact of morphological information on the performance of neural machine translation

models.

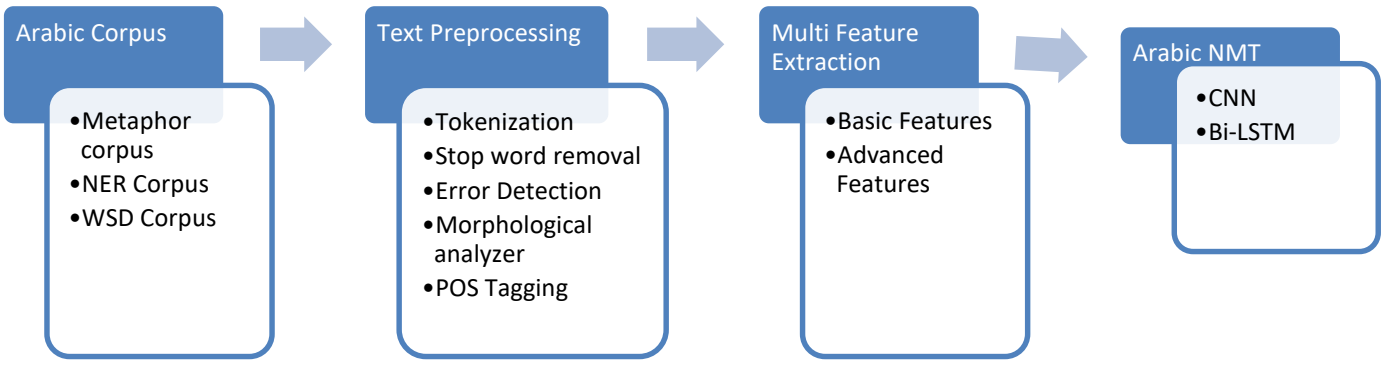


Figure 1: NMT Architecture

We propose a novel design framework for NMT using multi-feature extraction and deep neural network (DNN). We exploited a hybrid algorithm in metaphor, NER, and WSD, which aims to improve the overall of our NMT system performance and overcome all the critical aspects of the combined techniques when they are being processed individually. We invoke several processes for NMT that are as follows: text preprocessing, multi-feature extraction, feature selection, and classification. Firstly, we perform text preprocessing for the given text corpus. Here, we carried out four preprocessing steps: Tokenization, Stop Words Removal, Morphological Analysis, and Part-of-Speech Tagging. The Arabic text preprocessing is implemented using MADAMIRA v(1.0). Secondly, we perform multi-feature extraction based on the semantic relations of each word using Vector Space Model. TF-IDF is computed for each pair of input texts, and then optimum set of features is selected using Spider Monkey Optimization algorithm, which reduces computational burden in DNN. Thirdly, we fed the optimum set of features into DNN automatic recognition where we output a translated text that solves the three key challenges: named entities in three forms including person, organization, and location, metaphor, and word sense disambiguation.

We have used Al hadeeth WordNet to handle the unknown ambiguous words that can improve the translation quality for our model. Our system first tries to translate the words from training data corpus, if any of these words does not exist, we semantically find the closest synonym words from the Arabic Al Hadeeth WordNet for the unknown ambiguous words.

1.1. Research Question

In this research, we follow specific goals which define our research questions. As previously mentioned, we work with NNs in which both the input and output should be numerical vectors. Accordingly, before designing any MT model, characters, morphemes, and words as inputs and outputs in our case should be efficiently encoded into a numerical vector space. This process is called "*embedding learning*" (Mikolov et al., 2013a). Word embeddings preserve syntactic and semantic relations as well as contextual information. In addition to these types of information, we wish to highlight morphological dependencies in our embeddings, which is the focus of our first research question (**RQ1**). Clearly, in **RQ1**, we try to answer the question: *What is the best representation for Arabic morphological structure?* The model proposed for this research question is expected to provide a flexible framework to take morphologically complex structures with several subunits as its input and provide the surface-form embedding as well as subunit embeddings for the input and its internal constituents.

At the next step, we look beyond word-level modeling and focus on sequence modeling. The main challenge here is to model morphologically complex constituents at the sequence level. Language modeling by nature is a hard problem. It becomes more severe when the vocabulary is diverse and the out-of-vocabulary (OOV) word rate is high, a phenomenon frequently encountered in machine learning. We specifically try to solve problems related to rare and unknown words in language modeling, which are covered by the second research question (**RQ2**). In **RQ2**, our goal is to answer this question: *What is the most effective neural language model (NLM) for machine translation?* The model proposed for this research question is expected to receive a sequence of sub-word units as its input and model the sequence better than other word-, morpheme-, and character-level counterparts.

To answer the third research question (**RQ3**), we study methods by which we could incorporate NN-generated information into the conventional NMT pipeline. In **RQ3**, we try to enhance the quality of NMT models using results from the previous research questions. Similarly, in this part, we focus on NMT. **RQ3** mainly answers this question: *How do/can deep neural networks (DNNs) improve MT?* The framework proposed in this research question is expected to take an existing SMT engine and compare it without NMT model.

The fourth and last research question (**RQ4**) targets NMT models for MRLs. We try to perform an end-

to-end translation in purely neural settings. Existing neural architectures are not suitable for MT, as they do not consider the key challenges of MT. Accordingly, we propose several compatible neural architectures, and the main goal is to answer this question: *How can we find NMT models that are capable of translating metaphors, named entities, and ambiguous word senses?* Neural architectures proposed for this part of the research should be able to accept different types of inputs, provide high-quality representations for them, and generate the final translation better than other models. They should also be able to solve the challenges of MT with neural network architecture.

1.2. Thesis Structure

The thesis is divided into three main parts. The first part, including Chapter 1 through Chapter 4, explains the structure of the thesis along with fundamental concepts which we require to explain and expand our ideas. The second part covers the core research and answers our research questions in Chapter 5 through Chapter 10. The last part explains how this research contributes to our field and concludes the thesis with some avenues for future work in Chapter 11. More detailed information about each chapter is as follows:

- Chapter 1 explains the skeleton of the thesis along with the achievements and contributions.
- Chapter 2 provides basic concepts which we need to express the core ideas of the thesis.

2 Since the thesis is about Arabic NMT, using DNNs, first we have to introduce the problems related to Arabic NLP. Afterwards, Chapter 2 explains the fundamentals of MT in the Arabic language and highlights the key challenges of MT. In Chapter 3, we discuss different neural network algorithms. Apart from these introductory topics, Chapter 4 reviews the related literature. For the purpose of clarity, modularity, and consistency, we only review NN models in this chapter. All other chapters start with an introductory section followed by a background section including the literature review and continue with other subjects.

- Chapter 3 explains the key challenges of machine translation for Arabic language regarding classical Arabic, modern standard Arabic, and dialect Arabic.
- Chapter 4 is about neural language modeling for MT. It discusses different models for decomposing Arabic NMT. In this chapter, we not only propose a novel NLM but also, using

our models, manipulate n-gram-based language models (LMs) to provide better translations.

- Chapter 5 presents how neural network models can be used for the task of detection and correction of Arabic grammar and spelling errors at the word level. One of the ways of improving machine translation outputs is by performing the task of error detection and correction through pre- and post-editing, which, nowadays, is becoming a common practice in machine translation. We propose a novel deep-learning framework for performing error detection in Arabic text, which achieves state-of-the-art results on many gold standard datasets that have ill-formed words annotated, validated, and manually revised by Arabic linguistic specialists.

- Chapter 6 proposes an approach for developing a WordNet linguistic resource for Al- Hadith Al-Shareef that serves its purposes for various Arabic natural language processing tasks. We use the WordNet as a bilingual corpus to translate from Arabic to English and from English to Arabic. In particular, we establish semantic connections between words in order to achieve a good understanding of the meanings of the Al-Hadith words. Our approach employs classical Arabic dictionaries and Al-Hadith ontology.

- Chapter 7 presents a paraphrasing model based mainly on neural networks. It describes paraphrases in a continuous space and generates candidate paraphrases for an Arabic source input. Experimental results across datasets confirm that neural paraphrases significantly outperform those obtained with statistical machine translation, in particular Google translator, and indicate high similarity correlation between our model and human translation, making our model attractive for real-world deployment.

- Chapter 8 develops a hybrid system and builds a huge annotated corpus that can be used to detect the Arabic NER, which can improve the neural machine translation. The system is based on Hybrid Deep Learning with Evolutionary Algorithm which is also known as convolutional neural network (CNN) with Bi-LSTM and CRF. The proposed hybrid mechanism is tested on ANERCorp. This chapter involves three stages: the first stage is preprocessing where we clean the dataset by several steps (Tokenization, Stop Words Removal, Morphological

Segmentation, and POS Tagging), the second involves multi- feature extraction and selection using Vector Space Model (VSO) and Spider Monkey Optimization (SMO), respectively, and the final stage applies the algorithm to classify the data. Chapter 9 solves the second key challenge ofmachine translation, so it helps to answer Q3 and Q4 in our thesis.

- Chapter 9 presents and studies thoroughly a series of end-to-end neural architectures directly tailored to the task, from bidirectional long short-term memory to encoder- decoder models. The extensive assessment of standard benchmarks and in multiple languages shows that sequence learning allows for more versatile all-words models, consistently leading to state-of-the-art results, even against word experts with engineered features. Chapter 10 solved the third key challenge of machine translation, and it helps to answer Q3 and Q4.
- Chapter 10 concludes the thesis and explains our plans for future work. We summarize the thesis in this chapter and provide a roadmap which declares the goals achieved so far and some questions which should be solved in the future.

1.3. Contribution

The summary of the main contributions of the thesis is as follows:

- ❖ Developing a bilingual corpus of 600K Arabic–English words, which is used to build the Arabic WordNet.
- ❖ Developing a state-of-the-art neural network structure for error detection and correction, with 9 million words for misspelling errors corpus.
- ❖ Developing a state-of-the-art neural part-of-speech (POS) tagger for Arabic.
- ❖ Incorporating morphological information into word embeddings (**RQ1**).
- ❖ Mitigating the OOV word problem in embedding learning and language modeling (**RQ1** and **RQ2**).
- ❖ Proposing compatible NMT models for solving the issue of translating metaphor and building a corpus of sentence (**RQ3** and **RQ4**).
- ❖ Developing a state-of-the-art for detecting NER for NMT (**RQ3** and **RQ4**).
- ❖ Developing a state-of-the-art WSD for NMT (**RQ3** and **RQ4**).

- ❖ Enriching SMT phrase tables using word and phrase embeddings (**RQ3** and **RQ4**).

1.4 Publication

Publications which are directly related to the research conducted in this thesis include:

1. **Chapter 2**
    i. Shaalan, Khaled, Sanjeera Siddiqui, **Manar Alkhatib**, and Azza Abdel Monem. "Challenges in Arabic Natural Language Processing." Computational Linguistics, Speech and Image Processing for Arabic Language 4 (2018): 59.
2. **Chapter 3**
    i. **Alkhatib, Manar**, and Khaled Shaalan. "The Key Challenges for Arabic Machine Translation." In Intelligent Natural Language Processing: Trends and Applications, pp. 139-156. Springer, Cham, 2018.
3. **Chapter 6**
    i. **Alkhatib, Manar**, Azza Abdel Monem, and Khaled Shaalan. "A Rich Arabic WordNet Resource for Al-Hadith Al-Shareef." Procedia Computer Science 117 (2017): 101- 110.
4. **Chapter 7 and 9**
    i. **Alkhatib, Manar**, and Khaled Shaalan. "Natural language processing for Arabic metaphors: a conceptual approach." In International Conference on Advanced Intelligent Systems and Informatics (AISI'16), pp. 170-181. Springer, Cham, 2016.
    ii. **Alkhatib, Manar**, and Khaled Shaalan. "Paraphrasing Arabic Metaphor with Neural Machine Translation." Procedia Computer Science 142 (2018): 308-314.
5. **Chapter 8**
    i. **Alkhatibl, Manar,** May El Barachi, and Khaled Shaalan. "Using Arabic Social Media Feeds for Incident and Emergency Management in Smart Cities." In

2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech), pp. 1-6. IEEE, 2018.

ii. **Alkhatib, Manar**, May El Barachi, and Khaled Shaalan. "An Arabic social media based framework for incidents and events monitoring in smart cities." Journal of Cleaner Production 220 (2019): 771-785.

Chapter Two: Arabic Natural Language Processing

2.1 Introduction

Natural language processing (NLP) is a domain of computer science that aims at facilitating communication between machines (computers that understand machine language or programming language) and human beings (who communicate and understand natural languages like English, Arabic and Chinese etc.) NLP is very important as it makes a huge impact on our daily lives. Many applications these days' use concepts from NLP. In this chapter, we discuss the challenges of Arabic language with regard to its characteristics and their related computational problems at orthographic, morphological, and syntactic levels. In automating the process of analysing Arabic sentences, there is an overlap between these levels, as they all help in making sense and meaning of words, and in disambiguating the sentence.

Arabic is the sixth most spoken language in the world. (Farghaly and Shaalan, 2009) expressed the significance of Arabic from computational linguistics perspective. It is connected with Islam and more than 200 million Muslims perform their petitions five times daily utilizing this dialect. Moreover, Arabic is the first language of the Arab world countries, which has significant importance worldwide. Arabic is related to another linguistic family, particularly Semitic vernaculars, which is different from the Indo-European lingos talked in the West. Arabic is interesting and any person with a slight knowledge of Arabic can read and understand a text written fourteen centuries ago.

Arabic, as a language or dialect is exceedingly derivational and inflectional (Al-Shalabi 2008; Abd Al Salm 2009; Farra 2010), and there are no rules for emphasis (Dave et al. 2003; Wiebe and Riloff 2005; Harrag 2009; Barney 2010; Ghosh 2012). Truly, there are principles; however, there are no firm rules (Farghaly and Shaalan, 2009).

Arabic language has a rich and complex grammatical structure. For instance, a noun and its modifiers need to agree in number, gender, case, and definiteness (Shaalan et al. 2015). Moreover, in Arabic there are advancements that really mean, "mother of" or "father of" to show ownership, a trademark, or a property, and use gendered pronouns; it has no fair-minded pronouns (Izwaini 2006). Arabic sentences can be nominal (subject–verb), or verbal (verb–subject) with free order; however, English sentences are fundamentally in the (subject–verb) order. The free order property of the Arabic language presents a

crucial challenge for some Arabic NLP applications (Ray and Shaalan, 2016).

Arabic in use is characterized by three types: Classical (Traditional or Quranic) Arabic, Modern Standard Arabic and Dialect Arabic (Habash 2010; Korayem et al. 2012). Arabic language takes these forms in light of three key parameters including morphology, syntax and lexical mixes (Elgibali 2005; Abdel Monem et al. 2008; Farghaly and Shaalan 2009; Reifaee and Raiser 2014). Classical Arabic is primarily used in Arabic speaking countries, as opposed to within the diaspora. It is found in religious writings such as the Sunnah and Hadith, and numerous historical documents (Shaalan 2014). Diacritic marks (also known as "Tashkil" or short vowels) are commonly used within Classical Arabic as phonetic guides to show the correct pronunciation. On the contrary, diacritics are considered optional in most other Arabic writing.

Modern Standard Arabic (MSA) is used for TV, newspapers, poetry and in books. Arabic Courses at the Arab Academy are also taught in the Modern Standard form. The MSA can be transformed to adapt to new words that need to be created because of science or technology. However, the written Arabic script has seen no change in the alphabet, spelling or vocabulary in at least four millenniums. Hardly any living language can claim such a distinction.

Dialect Arabic or "colloquial Arabic" is casually utilized daily by Arabs. It is found in various nations and districts of a nation (Shaalan 2014). It is grouped into Mesopotamian Arabic, Arabian Peninsula Arabic, Syro Palestanian Arabic, Egyptian and Maghrebi Arabic. Arabic dialect generally used, mostly written, by Internet clients (Al-Kabi et al. 2014) and social media (Shaalan 2014); varies from locale to area is Dialect Arabic. In vernacular Arabic, portions of the words are acquired from MSA (Abo Bakr et al. 2008). Farghaly and Shaalan (2009) showed the significance of building local devices to chip away both Modern Standard and Dialect Arabic. Abo Bakr et al. (2008) presented a hybrid pre- processing approach that has the ability to convert paraphrases of Egyptian dialectal input into MSA such that the available NLP tools can be applied to the converted text. Siddiqui et al. (2016) worked on Sentiment Analysis on the data containing different Arabic Dialects.

2.2 Arabic Language

Arabic is the language of a large part of our planet. It is the main language in 22 countries, spoken by

more than 250 million people (Shaalan, 2014). It is also the second language in many Islamic countries because it is considered the spiritual language of Islam-one of the world's major religions. It is one of the official languages in the United Nations. However, separating it from another components of Arabic such as grammar, orthography, morphology, literature, writing, reading and conversation is necessary in order to facilitate a focus on its teaching and learning. The different NLP approaches for metaphor interpretation mainly depend on how the relation between the source and the target is viewed as a(n): analogy, novelty, or anomaly. Metaphorical expressions represent a great variety, ranging from conventional metaphors, which we reproduce and comprehend every day, such as "السيارة دي لهلوبة بنزين" (This car consumes a lot of petrol), to poetic, and novel, such as "وليل كموج البحر ارخى سدوله" (Like heavy waves, long nights 'pon me descend), and to Holy Qur'an (HQ) " وجعلناها سراجاً وهاجاً" (and we placed a radiant lamp). In this thesis, we present the Arabic metaphors that are in use in Holy Qur'an, Modern Standard Arabic (MSA) and Dialect Arabic (DA), which should be addressed by scholars interested in Arabic computational linguistics

2.1.1 Classical Arabic

Classical Arabic is the language of the Qur'an. The Quran is held by Muslims to be a single-authored text, the direct words of God (Allah), conveyed by the angel Gabriel to Mohammed 1355-1378 years ago, and later transcribed verbatim to be used as the sole authoritative source of knowledge, wisdom and low "إِنَّا أَنزَلْنَاهُ قُرْآنًا عَرَبِيًّا لَّعَلَّكُمْ تَعْقِلُونَ": " We have sent it down as an Arabic Qur´an, in order that you"-

2.1.2 Modern Standard Arabic

Modern Standard Arabic (MSA) is the official Arabic language nowadays. It is either written or spoken without any different in the form. MSA is the language of literature and the media; books, newspapers, magazines, official documents, private and business correspondence, street signs and shop signs – all are written in Modern Standard Arabic. MSA has been developed out of Classical Arabic, the language of the Quran. During the era of the caliphate, Classical Arabic was the language used for all religious, cultural, administrative and scholarly purposes. The linguistic features for this holy book provided unique aspects to MSA from literary, structural and stylistic points of view**.** MSA omits some classical grammatical constructs, has a stricter word order, uses a simpler numeral system, and obviously includes some more

recently coined or borrowed words (Diab, 2014).

2.2 Dialect Arabic

Tongues are the essential type of Arabic utilized as a part of all unscripted talked classifications: conversational, television shows, interviews, and so on. Dialects are progressively being used in digital media like newsgroups, weblogs, discussions and the like. Different countries such as Syria, Lebanon, Jordan, Palestine, Gulf and Egypt though use Arabic, but in reality, they are all different in dialects. Researchers need to consider this as a major fact and should not assume if a system is designed for Arabic dialect in Syria, then the same could benefit Morocco. Dissimilitude dialect is seen in terms of the variations from one another, which could be phonologically, lexically, morphologically, and linguistically; many sets of variations are commonly muddled. In unscripted circumstances where spoken MSA would typically be required (e.g. television shows on TV), the users more often than not depend on rehashed code-exchanging between their tongue and MSA, as almost all local speakers of Arabic cannot create supported unconstrained talk in MSA (Habbash and Rambow 2006), (Diab, 2014). For Example, consider the sentence “how are you ?” in different dialect: in Egypt dialect "ازايك" (azyk), in Gulf "شو اخبارج "(shw akhbarij), in Syria and Lebanon “كيفكن” (kifakun), and in Jordan and Palestine "كيفكم" (kayfakum) or "شيفكم" (shayfkum).

The following is only one of many that covers the main Arabic dialects (Habbash and Rambow, 2006):

- **Gulf Arabic (GLF)** includes the dialects of Bahrain, Kuwait, Qatar, United Arab Emirates, Saudi Arabia, and Oman. It is the closest of the regional dialect to MSA, perhaps because the current form of MSA evolved from an Arabic variety originating in the Gulf region
- **Iraqi Arabic (IRQ)** is the dialect of Iraq. In some dialect classifications, Iraqi Arabic is considered a sub-dialect of Gulf Arabic; though it has distinctive features of its own in terms of prepositions, verb conjugation, and pronunciation.
- **Levantine Arabic (LEV)** includes the dialects of Lebanon, Syria, Jordan, and Palestine. It differs somewhat in pronunciation and intonation, but are largely equivalent in written form; closely related to Aramaic.

- **Egyptian Arabic (EGY)** covers the dialects of the Nile valley: Egypt and Sudan. It is the most widely understood dialect, due to a thriving Egyptian television and movie industry, and Egypt's highly influential role in the region for much of 20th century.
- **Maghrebi** Arabic covers the dialects of Morocco, Algeria, Tunisia, Mauritania, and Libya. It is a large region with more variation than is seen in other regions such as the Levant and the Gulf, and could be subdivided further, even though it is heavily influenced by the French and Berber languages.

Socially, it is common to distinguish three sub dialects within each dialect region: city dwellers, peasants/farmers and Bedouins. The three degrees are often associated with a class hierarchy from rich, settled city-dwellers down to Bedouins. Different social associations exist as is common in many other languages around the world.

2.3 Challenges of Arabic Natural Language Processing

Arabic is an extremely bended tongue[1], with unique sound, especially when pronounce the letters "ض" (ḍād ("ظ" (Tha'a) ,and "غ" (Ghain). Arabic grammar has a rich morphology and intricate sentence structure and grammarians have described it as the language of ḍād ("لغة الضاد")[1] (Al-Sughaiyer and Al-Kharashi 2004; Ryding 2005). Habash (2007) states that Arabic has a greatly rich morphology depicted by a mix of templatic and affixational morphemes, complex morphological norms, and a rich part system. Arabic makes use of a lot of inflections [Bassam et al. 2002; Mostafa et al. 2006] because of the appendages which incorporate relational words and pronouns. Arabic morphology is perplexing because there are about 10,000 roots that are the basis for nouns and verbs (Darwish 2002). There are 120 patterns in Arabic morphology. Beesley (1996) highlighted the importance of 5000 roots for Arabic morphology.

he word order in Arabic is variant. We can have a free choice of the word we want to emphasize and put it at the head of sentence. Generally, the syntactic analyzer parses the input tokens produced by the lexical analyzer and tries to identify the sentence structure using Arabic grammar rules. The relatively free word order in an Arabic sentence causes syntactic ambiguities which require investigating all the possible grammar rules as well as the agreement between constituents (Siddiqui et al., 2016; Ray and Shaalan, 2016).

[1] The top alveolar ridge is located on the roof of the mouth between the upper teeth and the hard palate.

In this thesis, we discuss the challenges of Arabic language with regard to its characteristics and their related computational problems at orthographic, morphological, and syntactic levels. In automating the process of analyzing Arabic sentences, there is an overlap between these levels, as they all help in making sense and meaning of words, and in disambiguating the sentence.

2.3.1Arabic Orthography

Within the orthographic patterns of the written words, the shape of a letter can be changed depending on whether it is connected with a former and subsequent letter, or just connected with a former letter. For example, the shapes of the letter "ف" (f), i.e. "ﻓ"/"ﻔ"/"ﻒ", changes depending on whether it occurs in the beginning, middle, or end of a word, respectively. Arabic orthography includes a set of orthographic symbols, called diacritics that carry the intended pronunciation of words. This helps clarify the sense and meaning of the word.

As far as Qur'an is concerned, these vowel signs are absolutely necessary in order for children and those who are not well versed in classical Arabic language to pronounce religious text properly. It is worth noting that written copies of the Qur'ān cannot be accredited by religious institutes or authorities that review them unless the diacritics are included. The absence of short vowels (e.g. inner diacritics) prompts diverse sorts of equivocalness in Arabic writings (both basic and lexical) on the grounds that distinctive diacritics speak to distinctive implications. These ambiguities can be determined just by relevant data and satisfactory information of the language or dialect. Contextual features and an adequate knowledge of the language can only resolve these ambiguities (Ibrahim, 2013).

Arabic orthography includes 28 letters, all letters are consonants except three long vowels: "أ" (alef), "و" (Waw), and "ي" (yeh) and short vowels are represented by diacritical signs. This specificity brings into existence two forms of spelling: with or without vocalisation. The vowels added through a consonantal skeleton by means of diacritical marks produce a shallow orthography whereas vocalisation is missing. Orthography is deep and the word behaves as homograph that is semantically and phonologically ambiguous For instance, the unvoweled word "كتب" (ktb), supports several alternatives such as "كَتَبَ" (he wrote, kataba), "كُتِبَ" (it was written, kutiba), "كُتُبُ" (books, kutubun), etc. Voweled spelling is taught to novice readers, while unvoweled spelling constitutes the standard form and is gradually imposed at later

reading literacy stages. Unfortunately, MSA is devoid of diacritical markings and the restoration of these diacritics is an important task for other NLP applications such as text to speech (Said et al., 2013).

2.3.1.1 Lack of consistency in Orthography

Hamza Spelling

The most critical use of Hamza letter (“ء“ (”الهمزة”) brings in more challenges. With the very significance of Hamza being an additional letter seen at the top or bottom of the letters following the sounds of “ ,”ا و“”, or “ى”, i.e. “ؤ“ ,”أ”, or “ئ”, respectively. As these rules are confusing even for native speakers, Hamza is ignored most of the time while typing. NLP based systems should handle this assumption. There are many orthographical forms of the Hamza letter “the seat of Al-Hamza”, which is decided by the diacritics ("Tashkeel") of both the "Hamza" itself as well as the letter preceding it, i.e. either "Fatha", "Dama", "kasra" or "Sukun”. Exceptionally, when Hamza comes at the beginning of the word, we always write it over an "Alef", e.g. “أنا” (I, 'ana), or under it, e.g. “إيمان“ (faith, 'iiman).

According to its appearance and pronunciation, there are two types of Hamza: “همزة قطع” (Hamza Al-Qata’) and "همزة وصل” (Hamza Al-Wasl). Distinguishing each type is a challenge for both text and speech processing. Hamza Al-Qata’ is the regular Hamza and is always written and pronounced, e.g. “إيمان“ and “أنا”. On the contrary, Hamza Al-Wasl is neither written nor pronounced unless it is at the start of the utterance; a bare Aleh is used instead. A simple rule to recognize Hamza Al-Wasl is to add "و " (waw, and) before it and see whether or not it is pronounced; hence, writtenFor example, the Hamza in "إقرأ الكتاب" (iq-ra' AL’Kitab, read the book), is pronounced and written. However, if we add “و“ (waw, and) at the beginning of sentence as in "واقرأ الكتاب" (waq-ra’ Al-Kitab, and read the book) the Hamza is neither pronounced nor written. A more complicated example is "أخذت ابننا" (a-khadh-tu ibnana, I grabbed our son). In the first word "أخذت" (grabbed), the Hamza is a glottal stop (pronounced strongly) and should be pronounced, but in the second word "ابننا", it is neither written nor pronounced. When the diacritic mark of the "Hamza” is either “Fatha” or “Dama”, the Hamza appears at the middle or the end of a word and is written over the letter. Table 1 presents some examples with the addition of the Hamza and the challenge it brings in causing orthographic confusion. The Hamza is following a hierarchy of vowels in the language: The Kasra has the highest priority, the Dama has the medium

priority, and the Fatha has the lowest priority. If the diacritic is Kasra/Fatha/Dama for either the Hamza itself or the letter preceding it, the Hamza takes a Kasra/Fatha/Dama diacritic, respectively. The rules for determining the diacritic of Hamza are of notorious complexity. In transcribing to Arabic, it is difficult to determine the Hamza seat as well as the short vowel that it follows. These types of the Hamza are of a complex nature and need special handling by the computational system.

Example	**Tashkeel of Hamza**	**Tashkeel of letter before Hamza**	**Pronunciation**	**Translation**
سَأَل	Fatha	Fatha	Sa-ala	Asked
سُئِل	Kasra	Kasra	So-ela	Was asked
سُؤَال	Fatha	Dama	So-aal	Question

Table 1: The Hamza diacritic is determined by its own diacritics and the preceding letter

Al-Hamza orthographic variants are non-standard ways to spell a specific variant of a name, like "الامارات"instead of "الإمارات" (Al-Emarat, Emirates), in which the Hamza is omitted and bare Alef is used instead. Though the difference between these variants cannot be strictly defined, based on "statistical and linguistic analysis" of Modern Standard Arabic orthography (Halpern 2007), they are both occurring frequently. For example, the capital of The United Arab Emirates, "أبوظبي" (Abu Dhabi) can be written in different ways. According to statistics from Google, the most frequent ones are: "أبوظبي", "ابوظبي" and "بوظبي" with 13,800,000, 9,400,000, and 1,400,000 occurrences, respectively.

Defective Verb Ambiguity

Defective (weak) verb ("الفعل المعتل") is any verb that its root has a long vowel as one of its three radicals. These long vowels will go through a change when the verb is conjugated. For example, consider the case of a negated present tense verb that is preceded by the apocopative particle Lam "حرف الجزم لم". In Arabic, this particle is used for negating a present tense verb form which is understood as a negated past form (Ryding 2005). It is one of the defining features of Modern Standard Arabic, and is not used in any

dialects. Being able to use this word properly and effectively will bring Arabic language to a higher level.

Table 2 presents examples of this verb forms. The negative past tense verb causes ambiguity by having misspelling in writing skills. When the a pocopative particle Lam precedes a past tense verb, the verb changes to the present tense form by: 1) attaching a suitable present tense letter, 2) by omitting the long vowel in the verb, and 3) adding a short vowel to the last letter. Although apocopative particle Lam is used for the past tense, it can never be used with the perfective verb itself, rather it is only used before imperfective verbs.

Verb	Transliteration	Sentence	Change applied to the present form of the verb
دعا	Da-aa	لم يدعُ	Omit the last long vowel " و " and add the present tense letter " ي"
سعى	Sa-aa	لم يسعَ	Omit the last long vowel " ى" and add the present tense letter " ي"
صلى	Sala	لم يصلِ	Omit the last long vowel " ي " and add the present tense letter " ي "
زار	Zara	لم يزرْ	Omit the middle long vowel "ا" and add the present tense letter " ي "

Table 2: Examples of negated past tense verb form Nonappearance of capital letters

Arabic has no uncommon sign rendering the recognition of a Named Entity (NE) all the more difficult [Oudah et al., 2016]. On the other hand, English, in line with numerous other Latin script-based dialects, has a particular marker in orthography, in particular upper casing of the underlying letter, and showing that a word or succession of words is a named substance. Arabic does not have capital letters; this trademark speaks to an extensive hindrance for the basic task of Named Entity Recognition in light of the fact that in different languages, capital letters speak to a vital highlight in distinguishing formal people,

places or things (Shaalan, 2014). Along these lines, the issue of distinguishing appropriate names is especially troublesome for Arabic. For instance, in English, capital letters are used, e.g. “Adam“, but no capital letter in the same name in Arabic, e.g. “آدم”.

Another reality about Arabic to consider is that the vernacular has no capital letters (e.g. for proper names: the names of people, countries, months, days of the week); therefore, cannot make usage of acronyms. This can lead to confusion, especially during Information Extraction in general and Named Entity Recognition in particular. It makes it difficult to see names of substances. For example, the NE “الامارات العربية المتحدة” has the acronyms UAE in English but not in Arabic. Therefore, it is common to resolve the nonappearance of capital letters by analyzing the context surrounding the Named Entity.

Inherent Ambiguity in Named Entities

Most Arabic proper nouns (NEs) are indistinguishable from forms that are common nouns and adjectives (non-NEs) which might cause ambiguity. For example, the noun "الجزيرة" (Aljazeera) can be recognized as an organization name or a noun corresponding to island. Nevertheless, Arabic names that are derived from adjectives are usually ambiguous, which presents a crucial challenge for some Arabic NLP applications such as Arabic Named Entity Recognition. As an example, consider the word “أمل” (Amal), which means “hope”, and can be confused with the name of a person. In the following two sentences, the word “Amal” means two different senses:

1. الشباب هم أمل البلد which means: the youth is the hope of the country.
2. أمل بنت جميلة which means: Amal is a beautiful girl.

Remedies to resolve this type of ambiguity might not necessarily fix all problems. For example, consider the sentence "رأيت أمل" (I saw hope/Amal) which have either meaning.

Vowels

In written Arabic, there are two types of vowels: diacritical symbols and long vowels. Arabic text is dominantly written without diacritics which lead to major linguistic ambiguities in most cases as an Arabic word has different meaning depending on how it is diactritized. A diacritic sign (Tashkeel or Harakat) is not an orthographic letter. It is formed as diacritical marks above or below a consonant to give it a sound. Azmi and Almajed (2013) presented a good survey of recent works in the area of

automatic diacritization. There are three groups of diacritics (Said et al., 2013; Abu-Rabia 2001). The first group consists of the short vowel diacritics such as Fatha (َ), Dhamma (ُ), and Kasra (ِ). The second group represents the doubled case ending diacritics (Nunation or tanween) such as Tanween Fatha (ً), Tanween Kasra (ٍ), and Tanween Damma (ٌ). These are vowels occurring at the end of nominal words (nouns, adjectives and adverbs) indicating nominal indefiniteness. The third group is composed of Shadda (ّ) and Sukuun (ْ) diacritics. Shadda reflects the doubling of a consonant whereas Sukuun indicates the absence of a vowel and reflects a glottal stop.

Diacritics could also be classified into two main groups based on their functions. The first group includes the lexeme diacritics that determine the Part of Speech (POS) of a word as in كَتَبَ (wrote, kataba) and, كُتُبْ(books, kutub), and also the meaning of the word such as "مَدَرَسَة" (school, madarasa) and "مُدَرِسَة" (teacher/female, almudarisa). The second category represents the syntactic diacritics that reflect the syntactic function of the word in the sentence. For example, in the sentence "زَارَ اَلْوَلَدُ الحَدِيقَةَ" (The boy visited the garden, zar aalwalad alhadiqa), the syntactic diacritic "Fatha" of the word " الحَدِيقَةَ" (the garden, alhadiqa) reflects its "object" role in the sentence. While in sentence "تَزَيَنَتْ اَلْحَدَيقَةُ" (Spruced up the garden, tazayanat aalhadayqa) the same word occurs as a "subject" hence its syntactic diacritic is a "Damma". A text without diacritics adds layers of confusion for novice readers and for automatic computation. For example, the absence of diacritics is a serious obstacle to many of the applications such as text to speech (TTS), intent detection, and automatic understanding in general. Therefore, automatic diacritization is an essential component for many Arabic NLP applications.

The long vowels in English, which is "a","e", "I", "o" and "u", are the ones which are clearly spelled out in a text whereas in Arabic they are not. There are no exact matches between English and Arabic vowels; they may differ in quality, and they may behave differently under certain circumstances. All letters of the Arabic alphabet are consonants except three letters: "ا (Alef), "و" (Waw), and the letter "ي" (Ya'a) which are used as long vowels or diphthongs, and they also play a role as weak consonants [Abu-Rabia 2002]. The long vowel can appear at the beginning, in the middle, or at the end of a word, and it has many forms of pronunciation. Table 3 presents a homographic issue with the aid of an example:

"قالوا إنه لم يعش ، ولكن أمه لم تستسلم " (They said that he did not live, but his mother did not give up, Qalo Anaho lam ya'esh lakin Amah lam tastaslim). The acoustic and language models of the speech

processing systems should deal with long vowel issues.

Word	Transliteration	Meaning	Marks
قالوا	Qalo	Said-they	No pronunciation for the letter Alef at the end
لكن	Lakin	But	No appearance of the letter Alef in the middle but pronounced

Table 3: Homographic issue for long vowels

2.3.2 Lack of uniformity in writing styles

The high level of ambiguity of the Arabic script poses special challenges to developers of NLP areas such as Morphological Analysis, Named Entity Extraction and Machine Translation. These difficulties are exacerbated by the lack of comprehensive lexical resources, such as proper noun databases, and the multiplicity of ambiguous transcription schemes. The process of automatically transcribing a non-Arabic script into Arabic, is called Arabization, For example, transcribing an NE such as the city of Washington into Arabic NE produces variants such as such as the city of Washington into Arabic NE produces variants such as "وشنطن" "واشنغطن" ، "واشنطن" "واشنجطن" Arabizing is very difficult for many reasons; one is that Arabic has more speech sounds than Western European languages, which can ambiguously or erroneously lead to an NE having more variants. One solution is to retain all versions of the name variants with a possibility of linking them together. Another solution is to normalize each occurrence of the variant to a canonical form; this requires a mechanism (such as string distance calculation) for name variant matching between a name variant and its normalized form (Shaalan, 2014).

2.3.3 Arabic Morphology

An additional property of Arabic that should be noted is that Arabic is an exceptionally morphological rich lingo. Its vocabulary can be easily amplified using a framework that allows for a creative use of roots and morphological samples (Beesley 2001; Farghaly 1987; McCarthy 1981; Soudy et al. 2007; Abdel Monem et al. 2009; Shoukry and Rafea 2012; Farra et al 2010). According to Al-Fedaghi and Al-Anzi (1989), referred to in De Roeck and Al-Fares (2000), there are 85% of words from tri-demanding roots and there are around 10,000 free roots. Hence, Arabic is highly derivational and

inflection results in high inflections in morphology (Farghaly 1987; Ahmed 2000; Beesley 2001; Soudi 2007). Arabic is known for its templatic morphology where words involve roots and illustrations in the form of patterns, and fastened with affixes.

2.3.3.1 Morphology is Intricate

Arabic is a Semitic language that has a powerful morphology and a flexible word order. It is difficult to put a border between a word and sentence; yielding morpho-syntactic structure combinations for a word along the dimensions of parts of speech, inflection, declension, clitics, among other features (Ray and Shaalan, 2016). Arabic morphology and sentence structure give the ability to incorporate a broad number of adds to each word which makes the combinatorial expansion of possible words.

Lemma	Transliteration	=	Root	Transliteration	+ Pattern
مفتوح	Maftooh		فتح	Fath	م_؟ _ ؟ و_؟
مدروس	Madroos		درس	Daras	

Table 4: -Illustration of Arabic Language in the derivational stage

Arabic is highly derivational. All the Arabic verbs are derived from a base of three- or four-characters' root verb. Essentially, every one of the descriptors gets from a verb and every one of them are inferences too (Shaalan et al., 2015). Deductions in Arabic are quite often templatic; hence, we can say simply that: *Lemma = Root + Pattern*. Additionally, in case of a general deduction we can realize the significance of a lemma on the off chance that we know the root and Lemma, which have been utilized to determine it (Benajiba 2008). Table 4 depicts examples of the composite relation "Lemma = Root + Pattern", demonstrating a case of producing two Arabic verbs from the same classification and their inference/derivation from the same pattern. Notice that the Arabic root is consonantal whereas the pattern is the vowel(s) attached to a root.

2.3.3.2 Morphology Declension

Arabic is highly inflectional. The prefixes can be articles, relational words or conjunctions, though the suffixes are by and large protests or individual/possessive anaphora. As stated by Benajiba (2007), both prefixes and suffixes are permitted to be mixes, and along these lines a word can have zero or more

affixes, i.e. *Word = Prefix(es) + Lemma + Suffix(es)*. Arabic verb morphology is central to the construction of an Arabic sentence because of its richness of form and meaning. A more complicated example would be words that could represent an entire sentence in English such as "وسيحضرونها" (and they will bring it, wasayahdurunaha). This word can be written in this form:

وسيحضرونها = و+ س + ي + حضر + ونـ + ها (wa+sa+ya+hdr+runa+ha, and+will+bring+they +it)

In this example, the Lemma "حضر" (hadr) accepts three prefixes: "و" (wa), "س" (sa), and "ي" (ya) and two suffixes: "ون" (wa noun), and "ها" (ha). Thereby, because of the complexity of the Arabic morphology, building an Arabic NLP system is a challenging task.

The early step in analyzing an Arabic text is to identify the words in the input sentence based on its type and properties, and outputs them as tokens. There might be a problem in segmentation where some word fragments that should be parts of the lemma of a word and were mistaken to be part of the prefix or suffix of the word; thus, were separated from the rest of the word as a result of tokenization. This problem arises with Named Entities Recognition where the ending character n-grams of the Named Entity were mistaken for objects or personal/possessive anaphora, and were separated by tokenization [Shaalan, 2014]. Moreover, the POS tagger used for the training and test data may have produced some incorrect tags, incrementing the noise factor even further.

Another morphological challenge is highlighted by [Thakur 1997] with regard to relationships between words. The syntactic relationship that a word has with alternate words in the sentence shows itself in its inflectional endings and not in the spot in connection to alternate words in that sentence. For example, "المعلم المخلص يحترمه طلابه" (Al Mu'alim al-mukhlis yahtarimaho Tulabaho, the faithful teacher is respected by his students), the suffix pronoun "ـه" (Heh) in the two words "يحترمه" (yahtarima-ho, respected-him), and "طلابه" (Tulaba-ho, students-his) refers to the word "المعلم" (Al Mu'alim, teacher-the).

Generally, Arabic computational morphology is challenging because the morphological structure of Arabic also comprises a predominant system of clitics. These are morphemes that are grammatically independent, but morphologically dependent on another word or phrase [Shaalan, 2005a]. Subsequently, one can naturally conclude that this proportion is higher for Arabic information than for different languages with less perplexing morphology that the same word can be joined to various appends and

clitics and thus, the vocabulary is much greater. The following Arabic words: "مكتوب", (Maktoob, Written) "كتابات", (Kitabat, Writings), "كاتب" (Katib, Writer) "كتاب" (Kitab, Book), "كتب" (Kutob, Books) , "مكتب" (Maktab, Office) , "مكتبة" (Maktabah, Library), "كتابه" (Kitabah, Writing) are derived from the same Arabic three consonants trilateral with the origin verb "كتب" (Ktb, Wrote). They also refer to the same concept. To extract the stem from the words, there are two types of stemming. The first type is light stemming which is used to remove affixes (prefixes, infixes, and suffixes) that belong to the letters of the word "سألتمونيها" (sa'altamuniha); where they are formed by combinations of these letters. The second type is called heavy stemming (i.e. root stemming) which is used to extract the root of the words and includes implicitly light stemming (Al-Kabi 2015).

2.3.3.3 Annexation

Another morphologic challenge in Arabic language is that we can compose a word to another by a conjunction of two words. This conjunction can be with nouns, verbs, or particles. Although it is not common in traditional Arabic language, it is used in Modern Standard Arabic. Usually, the compound word is semantically transparent such that the meaning of the compound word is compositional in the sense that the meaning of the whole is equal to the meaning of parts put together. . For example, the word "رأسمالية " (capitalism, rasimalia) comes from compound of two nouns "رأس المال" (capital, ras almal); the word "مادام " (as long as, madam) comes from the compound of a particle "ما" (ma) and a verb "دام" (dam) , and the word "كيفما" (however) comes from the compound of two particles "كيف" (kayf) and "ما" (ma) (Walid, 2010). The meaning of a compound word is important for understanding the Arabic text, which is a challenge to POS tagging and applications that require semantic processing.

2.3.4Syntax is Intricate

Historically, as Islam spread, the Arab grammarians wanted to lay down the basis of grammar rules that prevents the incorrect reading of the Holy Qur'an. Arabic syntax is intricate. Automating the process that makes the computer analyze the Arabic sentences is truly a challenging problem from the computer perspective.

Arabic grammar distinguishes between two types of sentences: verbal and nominal. Verbal sentences usually begin with a verb, and they have at least a verb ("فعل", faeal) and a subject ("فاعل", faeil). The

subject as well as the object can be indicated by the conjugation of the verb, and not written separately. For example, the conjugated verb"شاهدتك" (I saw you, saw-I-you, shahidtuk) has a subject and an object suffix pronouns attached to it. Another example of a verbal sentence is"يدرس الولد" (studying the boy, "yadrus alwald"). This type of sentence is not applied in English sentences. All the English sentences begin with a subject, and followed by a verb, for example, "the boy is studying".

In Arabic, a nominal sentence begins with a noun or a pronoun. The nominal sentence has two parts: a subject or topic "مبتدأ", (mubtada) and a predicate "خبر", (khabar). The nominal sentences have two types: with or without a verb. The nominal verbless sentence is a typical noun phrase. When the nominal sentence is about being, which in some languages such as English requires the presence of the linking verb 'to be' (i.e. copula) in the sentence. This verb is not given in Arabic. Instead, it is implied and understood from the context. For example, "الطقس جميل" (alttaqs jamil) has two nouns without a verb; its English translation is "The Weather [is] wonderful". This can be confusing to second language learners who speak European languages and are used to have a verb in each sentence (Shaalan, 2005b; Hammo, 2014). Arabic grammar allows complex sentence structure formation which is discussed in the following subsections.

2.3.4.1 Multi word expressions

Multi word expressions are very important constructs because their total semantics usually cannot be determined by adding up the semantics of the parts. For example, the multi words expression " بالحديد والنار" (by force, bialhadid walnnar) consists of two words that have the literal meaning "حديد" (iron, hadid) and "نار" (fire, nar). Another example is the medical terminology "فقر الدم" (Anemia, faqar alddam) that consists of two words that has the literal meaning "فقر" (poor, faqar) and "دم" (blood, dam). These non-decomposable lexicalized phrases are syntactically-unalterable units that are unable to capture the effects of inflectional variation. Thus, they can cause problems in Machine Translation, Information Retrieval, Text Summarization, among other NLP applications. Such expression is termed as idiomatic multi word expressions. Other multi words expressions are words that co-occur together more often than not, but with transparent compositional semantics such as "رئيس الدولة" (The president of

the country, rayiys alddawla). As such, they do not pose a challenge in NLP applications. Such expressions could be of interest if we categorize them to typesas in Named Entity Recognition, i.e. contextual cues.

2.3.4.2 Anaphora Resolution

Anaphora Resolution is specifically concerned with matching up particular entities or pronouns with the nouns or names that they refer to. This is very important since without it a text would not be fully and correctly understood, and without finding the proper antecedent, the meaning and the role of the anaphor cannot be realized. Anaphora occurs very frequently in written texts and spoken dialogues. Almost all NLP applications such as Machine Translation, Information Extraction, Automatic Summarization, Question Answering, etc., require successful identification and resolution of anaphora (Hammami, 2009).

Anaphora Resolution is classically recognized as a very difficult problem in NLP. It is one of the challenging tasks that is very time consuming and requires a significant effort from the human annotator and the NLP system in order to understand and resolve references to earlier or later items in the discourse.

Ambiguous Anaphora

The pronominal anaphora is a very widely used type in Arabic language as it has empty semantic structure and does not have an independent meaning from their antecedent; the main subject. This pronoun could be a third personal pronoun, called "ضمير الغائب" (damir alghayib) in Arabic, such as "ها" /hA/ (her/hers/it/its), "ه" /h/ (him/his/it/its), "هم" /hm/ (masculine: them/their), and "هن" /hn/ (feminine: them/their).

As an example that shows the challenges of pronominal anaphora to NLP tasks, consider the result of using Google Translate© to translate two Arabic sentences into English [Al-Sabbagh, 2002]:

رأيت القطة ، فأعطيتها الطعام

Transliteration: ra'ayt alquttah, fa'aetiatuha alttaeam

- Google translation: I saw the cat, so I gave ***her*** food
- Correct translation: (I saw the cat, so I gave ***it*** food

رأيت الطفلة ، فأعطيتها الطعام

Transliteration ra'ayt alttaflah, fa'aetiatha alttaeam

- Google translation: I saw the little girl, so I gave ***her*** food
- correct translation: I saw the little girl, so I gave ***her*** food

The machine translation system fails to identify the correct antecedent indicated by the third personal pronoun "ها" /hA/ (her/hers/it/its) and thus external knowledge is needed in order to correctly identify this antecedent. There are differences between Arabic and English pronominal systems and Arabic is rich in morphology. The Arabic third person pronouns are commonly encliticized which make them ambiguous. Arabic pronominal does not differentiate linguistically between the value of the humanity feature, i.e. ±human. As a result, both the -HUMAN FEMININE noun "القطة" (the cat) and the +HUMAN FEMININE noun "الطفلة" (the little girl), causes ambiguity in the translated English sentence.

Syntactically flexible text sequence

Syntactically-flexible expressions exhibit a much wider range of syntactic variability and types of variations possible are in the form of Verb-Subject-Object constructions (Ray and Shaalan, 2016).

Arabic is generally a free word request language. While the essential word order in Classical Arabic and Modern Standard Arabic is verb-subject-object (VSO), they likewise permit subject-verb-object (SVO), object-subject-verb(OSV) and object-verb-subject (OVS). It is basic to utilize the SVO in daily papers features. Arabic vernaculars display the SVO request. Word order disparity is depicted in Table5. This makes the sentence generation of Arabic NLP applications a challenge. For example, in a question-answering system, the answer to the question "أين كتاب هدى؟" (where is Hoda's book? 'ayn kitab hudaa?) could be any sentence that is shown in Table 5 which indicates that Huda sold the book.

Examples in Arabic	**Transliteration English**	**Translation in English**	**Order**
باعت هدى الكتاب	sold Huda-NOM book-ACC	Huda sold the book	VSO
هدى باعت الكتاب	Huda-NOM sold book-ACC	Huda, she sold the book.	SVO
الكتاب هدى باعته	DEF-book-NOM Huda-NOM sold-it	The book, Huda sold it.	OSV
الكتاب باعته هدى	DEF-book-NOM sold-it Huda-NOM	The book, Huda sold it.	OVS

Table 1: Word order disparity

It is interesting to make a note of the placement of the word "كتاب" (Book, kitab) in Table 5. VSO does not topicalize any constituent as old data, which as a starting sentence in a talk cannot contain new components. Confirms that VSO does not concentrate a specific constituent, as opposed to different requests, which cannot be utilized in light of the fact that they just center a specific constituent (Hammo), Arabic case framework neglects to unmistakably stamp linguistic contentions. This particularly happens when the case marker, which is constantly included toward the end of the noun, cannot be incorporated on the grounds that the noun closes with a long vowel as opposed to a consonant. When this happens, elucidation of word request turns out to be entirely VSO, contrary to VOS.

Additional proof originates from a study of syntactic structures in the dialect, in which we find that VSO has the best dissemination. Bland inserted provisions, notwithstanding, may display both SVO and VSO orders (Usama, 2011).

Agreement

Agreement is a major syntactic principle that affects the analysis and generation of an Arabic sentence which is very significant to difficult NLP applications such as Machine Translation and Question Answering (Shaalan, 2005a; Ray and Shaalan, 2016). Agreement in Arabic is full or partial and is sensitive to word order effects (Abdel Monem et al., 2009). An adjective in Arabic usually follows the noun it modifies " الموصوف"(almawsuf) and fully agrees with respect to number, gender, case, and definiteness, e.g. "الولد المجتهد" (The diligent boy, alwald almujtahad) and "الأولاد المجتهدون" (The diligent boys, al'awlad almujtahidin). The verb is marked for agreement depending on the word order of the subject relative to the verb, see Figure 1. The verb in Verb-Subject-Object order agrees with the subject in gender, e.g. "جاء الولد / الأولاد" (came the-boy/the-boys, ja' alwalad/ ja' al'awlad) versus " / جاءت البنت البنات" (came the-girl/the-girls, ja'at albint/ ja'at albanat). In Subject-Verb-Object (SVO) order, the verb agrees with the subject with respect to number and gender, e.g. "الولد جاء / الأولاد جاءوا" (came the-boy/the-boys) versus "البنت جاءت / البنات جئن" (came the-girl/the-girls). In Aux-subject-verb word order, the auxiliary agrees only in gender while the main verb agrees in both gender and number, e.g. " كانت البنت تأكل الطعام / كانت البنات تأكلن الطعام" (the-girl was/the-girls were eating the food, kanat albint takul alttaeam/ kanat albanat takulun alttaeam). If the subject precedes the auxiliary, then both verbs agree with it in both gender and number "البنت كانت تأكل الطعام / البنات كن يأكلن الطعام" (albint kanat takul alttaeam / albanat

kunn yakuln alttaeam). as shows in figure2.

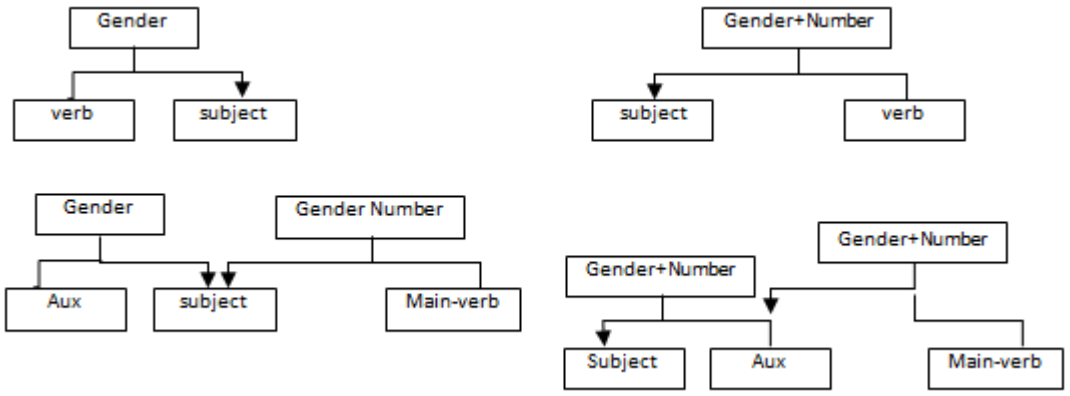


Figure 2- Agreement patterns in verb-subject vs. subject-verb word order

Some other agreements also exist between the numbers and the countable nouns [Shquier et al., 2008]. Number–counted noun agreement is governed by a set of complex rules for determining the literal number that agree with the counted noun with respect to gender and definiteness. In Arabic, the literal generation of numbers is classified into the following categories: digits, compounds, decades, and conjunctions. The case markings depend on the number–counted name expression within the sentence. In the following example, the number, "خمس" (five [masc.sg]) and the ((broken plural) counted noun "متاحف" (museums [fem.pl]) need to agree in gender and definiteness:

الأولاد زاروا خمسة متاحف

al'awlad zaruu khmst matahif

the-boys visited-they five.**fem**.sg museum.**fem**.pl

The boys visited five museums

2.4 Chapter Summary

Arabic as a language is both challenging and interesting. In this chapter, we delved into the basics of word and sentence structure, and relationships among sentence elements. This should help readers appreciate the complexity associated with Arabic NLP. The challenges of Arabic language were depicted by giving examples under MSA. It was found that although Arabic is a phonetic language in the sense that there is one-to- one mapping between the letters in the language and the sounds with which they are associated. An Arabic word does not dedicate letters to represent short vowels

requires changes in the letter form depending on its place in the word, and there is no notion of capitalization. As for MSA texts, short vowels are optional which makes it even more difficult for non-native speakers of Arabic to learn the language and present challenges to analyze Arabic words. Morphologically, the word structure is both rich and compact such that it can represent a phrase or a complete sentence. Syntactically, the Arabic sentence is long with complex syntax. Arabic Anaphora has increased the ambiguity of the language, as in some cases the machine translation system fails to identify the correct antecedent because of the ambiguity of the antecedent. External knowledge is needed to correct the antecedent. Moreover, Arabic sentence constituents (free word order) can be swapped without affecting structure or meaning, which adds more syntactic and semantic ambiguity, and requires analysis that is more complex. Nevertheless, agreement in Arabic is either full or partial and is sensitive to word order effects.

Chapter Three: The Key Challenges of Arabic Machine Translation

3.1 Arabic Machine Translation

Machine Translation has many challenges, and can be divided into linguistic and cultural categories. Linguistic problems include lexicon, syntax, morphology, text differences, rhetorical differences, and pragmatic factors.

Challenges arise for the Arab translator who may find certain phrases in Arabic have no equivalents in English. For example, the term (تيمم, tayammum) meaning "the Islamic act of dry ablution using purified sand or dust, which may be performed in place of ritual washing if no clean water is readily available", doesn't have a synonym concept in English.

Arabic has a complex morphology compared to English. Preprocessing the Arabic source by morphological segmentation has been shown to improve the performance of Arabic Machine Translation (Lee 2004; Sadat 2006; Habash 2010) by reducing the size of the source vocabulary and improving the quality of word alignments. The morphological analyzers that cause most segmentors were developed for Modern Standard Arabic (MSA), but the different dialects of Arabic share many of the morphological affixes of MSA, and so it is not unreasonable to expect MSA seg- mentation to also improve Dialect Arabic to English MT (Zbib et al. 2012).

Quran is a Holy book that teaches Islam, in which, it contains the main principles of Islam and how these principles should be conducted are written. The availability of digitalized translated Quran making the work of finding written knowledge in Quran becomes less complicated, and faster, especially for non-Arabic language familiar or speaker. Machine translations for Quran are available in Internet such as the websites of Islamicity.com and Tafsir.com, and there are more than 100 websites giving access to machine translation for Quran. Much work has been done on Modern Standard Arabic natural language processing and machine translation. MSA offers a wealth of resources in terms of morphological analyzers, disambiguation systems, annotated data, and parallel corpora. In contrast, research on dialectal Arabic (DA), the unstandardized spoken varieties of Arabic, is still lacking in NLP in general and in MT in particular (Alkhatib and Shaalan 2018).

The current work on natural language processing of Dialectal Arabic text is somewhat limited, especially

machine translation. Earlier studies on Dialectal Arabic MT have focused on normalizing dialectal input words into MSA equivalents before translating to English, and they deal with inputs that contain a limited fraction of dialectal words. (Sawaf 2010) presented a new MT system that is adjusted to handle dialect, spontaneous and noisy text from broadcast transmissions and internet web content. The Author described a novel approach on how to deal with Arabic dialectal data by normalizing the input text to a common form, and then processing that normalized format. He successfully processed normalized source into English using a hybrid MT. By processing the training and the test corpora, his method was able to improve the translation quality.

3.1.1 Machine Translation of Classical Quranic Arabic Text

The Holy Quran text has remained identical and unchanged, since its revelation, over the past 1400 years. The millions of copies of the Quran circulating in the world today match completely, to the level of a single letter. God says in the Holy Quran that he will guard the Quran book: “Surely it is we who have revealed the Exposition, and surely it is we who are its guardians”. Translating the Quran has always been problematic and difficult. Many argue that the holy Quran text cannot be mimicked in another language or form. Furthermore, the Quran’s words have shades of meanings depending on the context, making an accurate translation even more difficult. Translating the holy Quran requires more wordiness to get the meaning across, which diminishes the beautiful simplicity of the Quranic message. The various differences between Arabic and English cause many syntactic problems when translating the Holy Quran into English. Verb tense is an obvious syntactic problem that translators usually encounter in translating the Holy Quran. Verb tense means the ‘grammatical realization of location in time’ and how location in time can be expressed in language (Sadiq 2010). In translating the Holy Quran, the verb tense form should be guided by the overall context as well as by stylistic considerations. In the Holy Quran, there is a transformation from the past tense verb to the imperfect tense verb to achieve an effect, which can pose some problems and challenges in translation. For example

إِذْ جَاءُوكُم مِّن فَوْقِكُمْ وَمِنْ أَسْفَلَ مِنكُمْ وَإِذْ زَاغَتِ الْأَبْصَارُ وَبَلَغَتِ الْقُلُوبُ الْحَنَاجِرَ وَتَظُنُّونَ بِاللَّهِ الظُّنُونَا

(Behold! they came on you from above you and from below you, and behold, the eyes became dim and the hearts gaped up to the throats, and ye imagined various (vain) thoughts about Allah! (Yusuf Ali’s Translation 2000) [Surat Al-Aḥzāb 33, verse 10].

The verbs جاءوكم (Ja'ukum, comes against you'), زاغت (zaghat, grew wild) and (وبلغت, wabalaghat, reached) are in the past tense, but the verb (وتظنون, think) moves to the present tense. This move is for the purpose of conjuring an important action in the mind as if it were happening in the present. Tenses, in Classical Arabic or in the Holy Quran, cannot be transferred literally. In some cases, they need to move to convey the intended meaning to the target audience (Ali et al. 2012).

The Holy Quran has been interpreted and translated into many languages, including African, Asian, and European languages. The first translation of the Holy Quran was for Surat Al-Fatiha into Persian during the seventh century, by Salman the Persian. Another translation of the Holy Quran was completed in 884 in Alwar (Sindh, India, now Pakistan) under the orders of Abdullah bin Umar bin Abdul Aziz.

3.1.2 Machine Translation of Modern Standard Arabic Text

A word in Arabic is comprised of morpheme, clitics and affixation, as in the example in Table 1 ""وبجلوسهم" (wabajulusihim, and by their sitting). Since there is hardly any difference between complex and compound words in Arabic, this thesis uses compound words for both. Cells in the first column are the headers of their respective rows. The first row shows the example of a compound Arabic word. The second breaks down the compound word into its four morphemes. The third and fourth rows are the transliteration and translation of each morpheme, respectively. For the translation to be tangible, it must be rearranged (permuted), as shown by the arrows in Figure 3, into the phrase: "and by their sitting." The arrows show the necessary permutation that produces a palpable phrase.

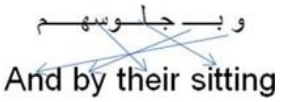


Figure 3: Example of translation

Arabic has different morphological and syntactic perspectives than other languages, which creates a real challenge for Arabic language researchers who wish to take advantage of current language processing technologies, especially to and from English. Moreover, Arabic verbs are indicated explicitly for multiple forms, representing the voice, the time of the action, and the person. These are also deployed with mood (indicative, imperative and interrogative). For nominal forms (nouns, adjectives, proper names), Arabic indicates case (accusative, genitive and nominative), number, gender and definiteness features. Arabic writing is also known for being underspecified for short vowels. When the genus is spiritual or educational, the Arabic text should be fully specified to avoid ambiguity. From the syntactic

standpoint, Arabic is considered as a pro-drop language where the subject of a verb can be implicitly determined in its morphology; the subject is embedded in the verb, unlike in English. For example, the sentence: *I went to the park* can be expressed in Arabic as " ذهبت الى الحديقة " (Dhahabt 'iilaa Alhadiqa, (I) went to the Park). The subject *"I" is dropped,* and the verb *went with the suffix pronoun* are represented in Arabic by the single verb-form "ذهبت" (Thahabat, went) That is, the translated phrase is *I went* to *the park*, with the last part translated as " الحديقة " (Alhadiqa, The Park).

Arabic demonstrates a larger freedom in the order of words within a sentence. It allows permutation of the standard order of components of a sentence—the Subject Verb-Object (SVO), and Verb Subject Object. As an example, the sentence "االطفل أكل الطعام" (Alttifl 'Akl Alttaeam, the child ate the food) can be translated, word-by-word, to the English SVO phrase "the child ate the food". The latter may be permuted to the standard Arabic order of a sentence—the VSO form "أكل الطفل الطعام" ('Akl Alttifl Alttaeam, ate the child the food). Both formspreserve the objective of the sentence. Unfortunately, the word by word English translation of the same VSO form is "Ate the child the food." Ironically, most of the online translation

Word	و بـــ جلـــوسهـــم			
Compound	هـــم	جلـــوس	بـــ	و
Transliteration	Himm	Juloos	Bi	Wa
Translation	Their	sitting	By	And

Table 6: Compound word

3.1.3 Machine Translation of Dialect Arabic Text

Dialect is the regional, temporal or social variety of a language, distinguished by pronunciation, grammar, or vocabulary; especially a variety of speech differing from the modern standard language or classical language. A dialect is thus related to the culture of its speakers, which varies within a specific community or group of people.

Arabic Dialect poses many challenges for machine translation, especially with the lack of data resources. Since Arabic dialects are much less common in written form than in spoken form, the first challenge is to basically find instances of written Arabic dialects. The regional dialects have been classified into five main groups; Egyptian, Levantine, Gulf, Iraqi and Maghrebin.

3.2 The Challenge of Metaphor in Machine Translation

3.2.1 Metaphor Translation

Metaphor is an expression used in everyday life communication to compare between two dissimilar things. It signifies a situation in which the unfamiliar is expressed in terms of the familiar. It is a central concept in literary studies.

Images tend to be universal in languages, as they are basically used to enhance understanding in interaction. Images, especially in speech, economize on time and effort in passing a message to its recipient. Metaphoric expressions are represented by metaphor, simile, and idioms in different languages and contexts.

Metaphor is the key figure of rhetoric, and usually implies a reference to figurative language in general. Therefore, it has always been attended to carefully by linguists, critics and writers. Traditionally, being originally a major aesthetic and rhetorical formulation, it has been analyzed and approached in terms of its constituent components (i.e. image, object, sense, etc.) and types (such as cliché, dead, anthropomorphic, recent, extended, compound, etc. metaphors). However, recently, and in the light of the latest developments of cognitive stylistics, metaphor has received even more attention from a completely different perspective, that of conceptualization and ideologization. Consequently, this change of perspective has its immediate effect on translation theory and practice, which now has to be approached differently with respect to translating metaphor. This thesis is an attempt to consider the translation of metaphor from a cognitive stylistic perspective, viewing it primarily as a matter of the conceptualization of topics, objects and people (Sardinha 2011) .

Metaphor is an expression used in everyday life in languages to compare between two dissimilar things. It signifies a situation in which the unfamiliar is expressed in terms of the familiar. In addition, it is a central concept in literary studies. A metaphor is sometimes confused with a simile, especially for translators who may translate metaphor into simile or vice versa. However, it is not too difficult to decide the case of simile because of the correlative existence of simile markers like “as, similar to and like”

which are not found in the metaphor (Sardinha 2011).

Simile refers to something or someone sharing a feature of something or someone else in which a significant commonality is established through one of the simile particles or through the relevant context. The rhetorical analysis of a simile requires the investigation of the two simile ends (طرفي التشبيه). These are the likened-to (المشبه) and the likened (المشبه) entities. Simile has four components and is divided into four categories. In any simile construction, the likened should be of a higher status, as the characteristic feature is greater than that found in the likened-to. For instance, when we say كلمات كالعسل (words like honey) or وجه كالقمر (her face like the moon), we are comparing (كلمات Kalemat, words) to (عسلAsal, honey) in terms of sweetness and (وجه Wajh, face) to (قمر, Qamar moon) in terms of beauty and brightness. Thus, rhetorically, the likened-to elements are represented by كلمات and وجه and the likened elements are عسل (Asal, honey) and قمر (Qamar, moon). However, the sweetness of honey and the brightness and beauty of the moon cannot be matched and are stronger than the features of the other entities.

Abdul-Raof (2006) stated that simile is realized through the following four components:

a. The likened-to (المشبه): The entity, i.e. a person or thing that is likened to another entity, which is the likened;
b. The likened (المشتبه به): The original entity to which another entity, i.e. the likened-to, is attached;
c. The simile feature: A feature that is common to both the simile ends; and
d. The simile element: The simile particles.

For example: أحمد كالأسد Ahmad Kalasad, Ahmad is like a lion, where:

- The likened to is represented by the noun) أحمد Ahmad);
- The likened is represented by the noun الأسد (Alasad, the lion);
- The simile element is represented by the particle ك (Ka, like); and
- The simile feature is represented by the implicit notion الشجاعة (AlShaja'ah, courage), which is a semantic link that is common to and shared by both nouns الأسد and أحمد.

In Arabic rhetoric, metaphor is referred to as "الاستعارة", which is a form of linguistic allegory and is regarded as the peak of figurative skills in spoken or written discourse. Metaphor is the master figure of

speech and is a compressed analogy. Through metaphor, the communicator can turn the cognitive or abstract into a concrete phrase that can be felt, seen, or smelt. Linguistically, الاستعارة is derived from the verb اعار (A'ar, to borrow), i.e. borrowing features from someone or something and applying them to someone or something else.

Rhetorically, however, metaphor is an effective simile whose one end of the two ends, i.e. the likened-to (المشتبه) and the likened (المشتبه به), has been deleted.

Metaphor represents a highly elevated effective status in Arabic rhetoric that cannot be attained by effective simile. In metaphor, the relationship between the intrinsic and non-intrinsic signification is established on the similarity between the two significations, i.e. there is a semantic link between the two meanings.

The metaphorical meaning, however, is discernible to the addressee through the lexical clue القرينة available in the speech act. In Arabic, metaphor consists of three major components. As there are different kinds of metaphor, these three components may not all be available in a single metaphor. Abdul-Raof (2006) stated that the three metaphor components are:

1. The borrowed-from: equivalent to the likened element in simile;
2. The lent-to: equivalent to the likened-to in simile; and
3. The borrowed: the borrowed lexical item taken from the borrowed-from and given to the lent-to

For example:

a. زيد أسد (Zaid Asad, Zaid is a lion). (effective simile)

b. رأيت أسدا في المدرسة (Ra'ayt 'asadaan fi Almadrasa, I saw a lion at school). (lion refers to a brave man)

- The lent-to is represented by the noun زيد(Zaid);
- The borrowed-from is represented by the noun أسد (Asad, lion); and
- The semantic feature الشجاعة (Alshaja'a, courage) is shared by and establishes the link between زيد (Zaid) and أسد (Asad) (is the borrowed).

In example (b), في المدرسة (Fi Almadrasa, at school) is the lexical clue to represent the metaphorical meaning of أسد lion" in this sentence, where lion refers to a brave man. Although metaphor makes the

text more beautiful and charming in the source language text (SLT) through its use of stereotyped words and new images, it can confuse the reader in the target language text (TLT) due to the linguistic and cultural differences between the two languages.

Kuiper and Allan (year) provide a definition about metaphor, as "an easy way to look at metaphor is to see the breaking down of the normal literal selection restrictions that the semantic components of words have in a sentence". When for example, we talk about " نافذة المستقبل", (Nafethat Almustaqbal, a window on the future), we have to ignore some of the semantic components of the word window;

for example, that it is a concrete object, and just take the fact that windows are things that allow us to look outwards from an enclosed space. The metaphor could also be seen out of a window. The metaphor lies in the suppression of some of each word's semantic features.

Metaphor can function as a means of formatting language in order to describe a certain concept, action or object to make it more comprehensive and accurate.

Hashemi (2002) classifies metaphors, i.e. isti'ara (الاستعارة), into three groups:

1. Declarative metaphors (تصريحية, Tasrihiyya): in which only the vehicle is mentioned and the tenor is deleted. In this type of isti'ara, the vehicle is explicitly stated and used to make a comparison between two fferent concepts that share a feature or a property in order to reveal the senses. A Declarative Metaphor is also considered as a decorative addition to ordinary plain speech. It is also used to achieve aesthetic effects (ibid). For example, in Arabic one might say (وردة, zahra, a rose) "رأيت وردة" *I saw a rose* instead of saying (a beautiful woman) امرأة جميلة, which is the vehicle in a metaphor based on the similarity between a rose and the person in terms of beauty.
2. Cognitive Metaphor (مكنية, Makniya): in which only the tenor is mentioned and the vehicle is deleted. In this type of isti'ara, the vehicle is only implied by mentioning a verb or a noun that always accompanies it. A Cognitive Metaphor is used as a means of formatting language in order to describe a certain concept, action or object to make it more comprehensive and accurate. In this case, it focuses on the denotation rather than the connotation of the metaphor that addresses the receptor in order to highlight its cognitive function.
3. Assimilative Metaphor (تمثيلي , Tamthele): which uses one of the characteristics of a vehicle for tenor. For instance, "إذا رَأيتَ نُيُوبَ اللّيثِ بارزَةً ... فَلا تَظُنَّنَ أَنَّ اللّيثَ يَبْتَسِمُ." when you see a lion baring his

canines, ،never think he is smiling.

Newmark (1988:105−113) provides another classification of metaphor, divided into six types: dead, cliché, stock, adapted, recent and original.

Dead metaphors

Dead metaphors are "metaphors where one is hardly conscious of the image, which relate to universal terms of space and time, the main parts of the body, general ecological features and the main human activities." Here the sense of transferred imageno longer exists. Through overuse, the metaphor has lost its figurative value. For example "خلص الوقت" (run out of time).

English words that represent dead metaphors include: "space, field, line, top, bottom, foot, mouth, arm, circle, drop, fall, and rise are particularly used graphically for the language of science to clarify or define.", some other examples are, *I didn't catch his name*, *foot filed*, *top*…etc., and an example in Arabic "عقارب الساعة" which means (hands of the clock). Dead metaphors are not difficult to translate literally; even though they could lose their figurative meaning through extensive popular use. Another example is (field of human knowledge).

Cliché metaphors

Cliché Metaphors are "metaphors that have perhaps temporarily outlived their usefulness, that are used as a substitute for clear thought, often emotively, but without corresponding to the facts of the matter." One example in English would be *at the end of the day*, and an example in Arabic is (في نهاية المطاف Fi nehayat almataf).

Stock or standard metaphors

Newmark describes this kind of metaphor as "An established metaphor, in an informal context, is an efficient and concise method of covering a physical and/or mental situation both referentially and pragmatically". It has certain emotional warmth, which does not lose its brightness by overuse. These are sometimes difficult to translate since their apparent equivalents may be out of date or now used by a different social class or age group. According to Newmark, a stock metaphor that does not come naturally to you should not be used, which means, if these metaphors are unnatural or senseless in the target language, they should not be used.

Recent metaphors

Recent metaphors, where an anonymous metaphorical neologism has become something generally used in the source language. It may be a metaphor designating one of a number of 'prototypical' qualities that constantly 'renew' themselves in language. For example, (تصفية الخصوم السياسية, Tasfiyat Alkhosoom Alseyaseyah, head hunting).

Adapted metaphor

An adapted metaphor is an adaptation of an existing (stock) metaphor. This type of metaphor should be translated by an equivalent adapted metaphor; it may be incomprehensible if it is translated literally, as in (الكرة في ملعبه ,Alkora fi mal'aboh, the ball is in his court).

Original metaphors

Original metaphors refer to those created or quoted by the Source Language writers in authoritative and expressive texts. These metaphors should be translated literally, whether they are universal, cultural, or obscurely subjective.

3.2.2 Metaphor in Arabic Language

3.2.2.1 Metaphor in Modern Standard Arabic

Metaphor is the process of 'transporting' qualities from one object to another, one person to another, from a thing to a person or animal, etc. When translating a metaphor, it is necessary to start by investigating the concept of metaphor, with the focus on contemporary conceptual approaches of metaphor. There have been rapid and revolutionary changes in communications, computers, and Internet technologies in recent years, along with huge changes in the conceptual studies of metaphor.

A metaphor is a figure of speech that involves a comparison, and a simile is also a figure of speech which involves a comparison. The only difference between them is that in a simile the comparison is explicitly stated, usually by a word such as "like" or "as", while in a metaphor the comparison is implied. Machine translation is much more likely to function correctly for simile than it can for metaphor. For instance, using Google translator:

a. "اشتعل الرأس شيباً" (Eshta'al Alra's Shayban, Flared head Chiba); and
b. "شعره كالثلج " (Sha'aroh Kalthalj, his hair such as snow).

In the second example would help in translation, as it represents a simile, but in the first example the metaphor is implicit and so its translation is much more difficult. Another example is رأيت أسداً في المدرسة (Ra'ayt Asadan fi Almadrasa, I saw a lion in the school), it does not mean that "I saw the lion (the animal), but rather that "I saw a man like a lion in his brave demeanor", here describing the bravery of the man like that of a lion, the king of the forest and the strongest among others.

3.2.2.2 Metaphor in Dialect Arabic

Arabic dialects, collectively referred to here as Dialectal Arabic (DA), are the day to day vernaculars spoken in the Arab world. Metaphorical expressions are pervasive in day-to-day speech. The Arabic language is a collection of historically related variants that live side by side with Modern Standard Arabic (MSA). As spoken varieties of Arabic, they differ from MSA on all levels of linguistic representation, from phonology, morphology and lexicon to syntax, semantics, and pragmatic language use. The most extreme differences are at the phonological and morpho- logical levels. We can see the difference in meaning with the use of the word white in metaphorical expressions. For example, the expression in the dialect Arabic سارة قلبها زي الثلج(Sarah Qalbaha zay Althalj, Sara's heart [is] like snow) expresses that Sara is a good person, whereas the expression كدبة بيضة (Kedba Bedhah, a white lie) means a lie that is "honest and harmless". Another example is praise with the word "donkey" in the expression سارة حمارة شغل (Sarah Hemart Shoghol, Sara is a donkey at work) which means "She is a very patient and hard worker". However, describing a person as a donkey in the dialect Arabic is very offensive and has connotations such as foolish or stupid. In dialect metaphors, we usually use the bad words (bad expressions) to express a good adjective and the vice versa.

Dialect Metaphors expressions are day-to-day speech that people use all the time (Biadsy 2009):

- In arguments like "مافيك تدافع عن موقفك" (Mafeek Tedafe'a an mauqifak, you cannot defend your position) contain the word "تدافع" (defend); it must be for something like country or building. We consider the person in the argument with us as an opponent and we attack his position. Another example "حكيو ضرب على الراس", (Haku Dhareb ala Alras, his speech is hitting it on the head), means that he is getting to the heart of the matter.
- Utilizing ideas and peech as food and commodities: "أفكاره مهضومة", (Afkaroh Mahdomeh, his ideas [are] tasty and sweet), means that his ideas are nice and appropriate, while "أفكاره بلا طعمه" (Bla

Ta’meh, his ideas [are] without taste) means that they are not useful, or even harmful.

- Two other examples are (حط ببطنك بطيخة صيفي, Hoot Bebatnak Batekha Saifi, eat watermelon), which means ‘relax and don’t worry’, and “طحن الكتب طحن” (Tahan, Elkutob Tahen, he smashed the books), which means that he studied the books thoroughly.
- To express time: “إجا وقت الجد” (Eja Waqt aljad, the time of seriousness has come) means that it is time to work hard and be serious. Other examples of time metaphors are “راح آذار’’ (March went away), meaning March has ended, and "شتا صار على الأبواب" (Alshita sar ala alabwab, winter has reached our door- steps), which means winter will start soon.
- Times are used as location: “نط التسعين” (Nat Altes’en, He jumped over ninety) means he is over ninety years old, and “العام الي مرق” (Alam Eli Maraq, the year that passed) means the last year, and here describes the year as a person that has walked away.

Dialect metaphors are difficult to understand correctly, unless we are familiar with them and we are from the same culture with the same dialect, as each country (and even each region) has its own metaphor dialect.

3.3 Arabic Named Entity Recognition Translation

The Named Entity Recognition (NER) task consists of determining and classifying proper names within an open-domain text. This Natural Language Processing task is acknowledged to be more difficult for the Arabic language, as it has such a complex morphology. NER has also been confirmed to help in Natural Language Processing tasks such as Machine Translation, Information Retrieval and Question Answering to obtain a higher performance. NER can also be defined as a task that attempts to determine, extract, and automatically classify proper name entities into predefined classes or types in open-domain text. The importance of named entities is their pervasiveness, which is proven by the high frequency, including occurrence and co-occurrence, of named entities in corpora. Arabic is a language of rich morphology and syntax. The peculiarities and characteristics of the Arabic language pose particular challenges for NER. There has been a growing interest in addressing these challenges to encourage the development of a productive and robust Arabic Named Entity Recognition system (Shaalan 2014)

The NER task was defined so that it can determine the appropriate names within an open domain text and

categorize them as one of the following four classes:

1. Person: person name or family name;
2. Location: name of geographically, and defined location;
3. Organization: corporate, institute, governmental, or other organizational entity; and
4. Miscellaneous: the rest of proper names (vehicles, brand, weapons, etc.).

In the English language the determination of the named entities (NEs) in a text is a quite easy sub-task if we can use capital letters as indicators of where the NEs start and where they end. However, this is only possible when capital letters are also supported in the target language, which is not the case for the Arabic language. The absence of capital letters in the Arabic language is the main difficulty to achieving high performance in NER (Benajiba 2008; Benajiba and Rosso 2007; Shaalan 2014). To reduce data sparseness in Arabic texts two solutions are possible: (i) Stem- ming: omitting all of the clitics, prefixes and suffixes that have been added to a lemma to find the needed meaning. This solution is appropriate for tasks such as Information Retrieval and Question Answering because the prepositions, articles and conjunctions are considered as stop words and are not taken into consideration when deciding whether or not a document is relevant for a query. An implementation of this solution is available in Darwish and Magdy (2014); (ii) Word segmentation: separating the different components of a word by a space (blank) character. This solution is more appropriate for NLP tasks that require maintaining the different word morphemes such as Word Sense Disambiguation, Named Entity Recognition, etc.

NER in Dialect Arabic is completely different than it is in MSA. For example, a person name in either DA or MSA could be expressed in DA by more than one form; for example, the name "قمر طارق"(Qamar Tareq) in MSD , can be "امر طاریء" (Amar Tare'a) and " كمر طارك " (Kamar Tarek); the main complication is that the first name is a girl's name, when translated it can be 'moon' and not appear as a Name Entity for a person.

Another issue in NER is the ambiguity between two or more NEs. For example consider the following text: (عيد سعيد عيد مبارك). In this example, the (Eid) is both a person's name and a greeting for Al Eid , thereby giving rise to a conflict situation, where the same NE is tagged as two different NE types. The same in the following names (حصة ، موزة ، شمس ، هند، جمعة},for example "حصة مرحة" (Hesa is a funny girl) and "موزة مهضومة" , which means "Mouza is cute ",another example is the name "أحمد الفهد الصباح"; these

are all person-names and do not refer to an animal or a timing period

In Machine Translation, NEs require different translation techniques than the rest of the words of a text. The post-editing step is also more expensive when the errors of an MT system are mainly in the translation of NEs. This situation inspired (Babych and Hartley 2003) to conduct a research study in which he tagged a text with an NER system as a pre-processing step of MT. He found achieved a higher accuracy with this new approach which helps the MT system to switch to a different translation technique when a Named Entity is detected (Othman 2009).

3.3.1 Arabic Named Entity Recognition Characteristics

Arabic language is one of the richest natural languages in the world in terms of morphology and inflection. Applying NLP tasks in general and NER task in particular is very challenging when it comes to Arabic language because of its characteristics. The main characteristics of Arabic language that act as challenges for NER task as follows:

a. No Capitalization

Capitalization in not a feature of Arabic script unlike several natural languages such as where a NE usually begins with capital letter. Therefore, the usage of this orthographic features is not an option in Arabic NER. However, the English translation of Arabic words may be exploited in this aspect

b. The Agglutinative Nature

Arabic is a great inflectional language; a single word has more than one affix. It is expressed as a combination of prefix, lemma, and suffix. Prefixes are articles, prepositions, and conjunctions, while suffixes are objects or personal anaphora. For example, (وجعلناهم, wjElnAhm, and we made to them).

c. Spelling Variant

Arabic spelling and typographic forms are different from other languages. A word can be spelled differently and still refer to the same meaning, which will create a many-to-one ambiguity. For example, ―Jeddah‖ can be written as _ جدة ‘ or جده ‘ and the word ―Gram‖ can be written as _ جرام ‘ or _ غرام‘, both of which have the same meaning.

d. No Short Vowels

Arabic texts can have different meanings (sorts of ambiguities). For example, أكلنا العيشت بالجبن ― ‖ can

mean ―we ate bread with cheese‖ or ―we made a living by being cowardly‖; this is the lexical ambiguity of the Arabic word.

3.4 Word Sense Disambiguation Challenge in Machine Translation

The Arabic Language contains several kinds of ambiguity; many words can be in various characteristics based on certain contexts. For example, the word دين has two meaning; the first refers to religion and the second refers to deptmoney. Such ambiguity can be easily distinguished by a human using common sense, while machine translation cannot distinguish the difference. Instead, MT requires more complex analysis and computation in order to correctly identify the meaning; this process is called Word Sense Disambiguation (WSD) (Mussa and Tiun 2015); (Hadni, Alaoui, and Lachkar 2016). Word Sense Disambiguation (WSD) is the problem of identifying the sense (meaning) of a word within a specific context. In Natural Language Processing (NLP), WSD is the task of automatically determining the meaning of a word by considering the associated context (Ponzetto and Navigli 2010). It is a complicated but crucial task in many areas, such as Topic Detection and Indexing, Information Retrieval, Information Extraction, Machine Translation, Semantic Annotation, Cross-Document Co-Referencing and Web People Search. Given the current explosive growth of online information and content, an efficient and high-quality disambiguation method with high scalability is of vital importance to allow for a better understanding, and consequently, improved exploitation of processed linguistic material (Hadni 2016).

3.4.1 Word Sense Disambiguation Characteristics

One example of an ambiguous Arabic word is "خال" (Khal), which can be translated to any of the following three words: "empty", "imagined", "battalion" or "uncle." Due to the undiacritized and unvowelized Arabic writing system, the three meanings are conflated. Generally, Arabic is loaded with polysemous words. One interesting observation about the Arabic language is its incredible reuse of names of the human body parts. For example, imagining the word رأس 'head' one could think of the neck, nose, eyes, ears, tongue and so on (Abuelyaman et al. 2014).

Apparently, when many researchers translating Quran to English language, several semantic issues have been appeared. Such issues pose the ambiguity of words, for example ليلاًونهاراً (laylan wanaharan)

and يوم الحساب (Yaum Alhesab), which are translated into “day and night” and “judgment day”, respectively. Such ambiguity has to be omitted by determining the correct sense of the translated word. In MSA, synonyms are very common, for example the word year has two different synonyms in Arabic for example (سنة sanah, and عام Aam) and both of them are widely used in everyday communication. Despite the issues and complexity of Arabic morphology, this impedes the matching of the Arabic word.

Ambiguity is not limited to Arabic words only, but also to Arabic letters when they affixed to morphemes, lead to ambiguous compound words. Table 2 shows how affixing the letter ‘ب’ which corresponds to ‘b’ in English, to an atomic word will turn it into a compound one. This is because, as a prefix, the letter ‘ب’ takes on any of the following senses: through, in, by, for and at. Table 7 shows only five of the ten possible roles the letter ‘ب’ plays when prefixed to different words (Abuelyaman et al. 2014).

Arabic texts without diacritics pose the greatest challenge for WSD, as they increase the number of a word’s possible senses and consequently make the disambiguation task much more difficult. For example, the word صوت Sawt (sound) without diacritics has 11 senses according to the Arabic WordNet (AWN) (Bouhriz and Benabbou 2016), while the use of diacritics for the same word وَSawata cuts down the number of senses to two. Another example is the word مال, which hasseven senses in) (Bouhriz and Benabbou 2016):

(Bouhriz and Benabbou 2016):

- {,فلوس,ثَروة,دراهم,مَال,مَ} Sense1
- {,نقود,مَال}Sense 2
- {,تمايل,رنح,تَ, مال,مَ} Sense 3
- {,انحدر,مَال,مَ} 4 Sense
- { إلى نزع ،مال} Sense 5
- { مال,أقنع,أَمال, } 6 Sense 6
- {مال,انحنى,انحرف.} Sense 7

Word	Translation	Word‖ بـ	Translation of word‖ بـــ
بركة	Blessing	بـــبركة	**Through blessing**
المدرسة	The school	بالمدرسة	**In the school**
المال	The money	بـــالمال	**By the money**
أي	What	بـــأي	**For what**
الباب	The door	بـــالباب	**At the door**
القلم	**The pen**	**بـــالقلم**	**Using the pen**

Table 7:Letter ambiguity

The WSD approach has shown that two words before and after an ambiguous word are sufficient for its disambiguation in almost all languages (Mohamed and Tiun 2015). For the Arabic language, the information extracted from this local context is not always sufficient. To solve this problem, an Arabic WSD system has been proposed that is not only based on the local context, but also on the global context extracted from the full text (Bouhriz and Benabbou 2016). The objective of their approach is to combine the local contextual information with the global one for a better disambiguation using the resource Arabic WordNet (AWN) to select word senses.

All of the WSD approaches make use of words in a sentence to mutually disambiguate each other (Chen et al. 2009; Agire et al. 2009; Ponzetto et al. 2010). The distinction between various approaches lies in the source and type of knowledge made by the lexical units in a sentence. Thus, all of these approaches can be classified into either corpus-based or knowledge-based methods. Corpus-based methods use machine-learning techniques to induce models of word usages from large collections of text examples. Statistical information that may be monolingual or bilingual, raw or sense-tagged is extracted from corpora. Knowledge-based methods instead use external knowledge resources that define explicit sense distinctions for assigning the correct sense of a word in context. (Dagan and Itai 1994); (Gale et al. 1992) used Machine-Readable Dictionaries (MRDs), thesauri, and computational lexicons, such as WordNet (WN). (Dagan and Itai 1994) was the first to resolve lexical ambiguities in one language using statistical data from the monolingual corpus of another language. That approach exploits the differences between the mappings of words to senses in different languages.

3.5 Chapter Summary

The chapter presents the key the challenges of translating the Arabic language into the English language according to the classical Arabic, Modern Standard Arabic and Dialect Arabic. It also has suggested a line of argument in favors of the conceptualization of Word Sense Disambiguation, Metaphor, and Named Entity Recognition. Up to date, little work has been published on Arabic language translation. Arabic sentences are usually long, the punctuation is not affecting on the text interpretation. Contextual analysis is very important in the Arabic text translation, in order to understand the exact meaning of the word. The absence of diacritization in most of the MSD and completely in Dialect Arabic pose a real challenge in Arabic Natural Language Processing, especially in Machine translation. The Arabic language has many features that are inherently challenging for NLP researchers. The difficulties associated with recognizing the need for full-verbs likes of “is”, and adverbs-of-places—the likes of “there”, recognizing the appropriate senses of un-diacritized words, and the practice of performing translation at the compound word level are some of the main issues. Classical Arabic is regarded as rhetorical and eloquent because of its stylistic and linguistic manifestations. Translators who are not well-acquainted with this religious discourse cannot succeed in relaying the linguistic, stylistic and cultural aspects in the translated language. Unlike an ordinary text, the classical discourse is featured is noted to be sensitive; its language is euphemistic, indirect, and solicitous of people’s feelings. While Dialect can be a crucial element in the process of describing and individualizing characters in literature and therefore should be handled with great care. Dialect phonetic, grammatical and syntactic effect should directly or indirectly be preserved in the target language

Chapter Four: Deep Neural Network

4.1 Introduction

In this chapter, I will discuss the following Language Model (LM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN). Language modeling is an imperative concept in natural language processing enabling us to predict words, i.e., speculating which word will come next within a pre-existing context. Neural machine translation, as will be revealed through the text started from language modeling. But first, we will discuss the basics of the neural network, the core of sequence-based NMT, to illustrate how NNs can naturally and effectively model variable-length inputs or sentences in the context of the translation task. One particular type of RNN will be covered in-depth, the Long Short-term Memory (LSTM) that facilitates training RNNs. Interested readers can find all the information detailing how to manually implement LSTM with detailed formulas on gradient computation in contrast to the automatic differentiation feature given by nowadays deep learning frameworks. The understanding of language modeling should enable us to extend RNNs into recurrent neural language models which enable language generation, a key step in NMT. Lastly, with RNN as a basic building block, I describe key elements of an NMT system.

4.2 Language Model

Language modeling plays an indispensable role in MT to ensure that systems produce fluent translations. Specifically, the function of the LM is to specify the probability distribution over sequences of symbols (often, words) so that one can judge if a sequence of words is more likely or "fluent" than another. To accomplish that, an LM decomposes the probability of a word.

To model these conditional probabilities, traditional n-gram LMs have to resort to the Markovian assumption to consider only a fixed context window of $n-1$ words, effectively modeling:

$$(y_i | y_{i-n+1} \cdots \cdot y_{i-1}),$$

In fact, n-gram LMs have to explicitly store and handleall possible n-grams occurred in a training corpus, the number of which quickly becomes enormous. As a result, despite much research in this area (Rosenfeld, 2000; Stolcke, 2002; Teh, 2006; Federico et al., 2008; Heafield, 2011), inter alia, n-gram

LMs can only handle short contexts of about 4 to 6 words, and does not generalize well to unseen n-grams.

Neural language models (NLMs), first proposed by (Bengio et al., 2003) and enhanced by others such asMorin and Bengio (2005);Mnih and Hinton (2009);Mnih and Teh (2012), have addressed the aforementioned concerns using two ideas: (a) dense distributed representations for words which encourage sharing of statistical weights between similar words; and (b) feed-forward neural networks to allow for better composition of unseen word sequences at test time without having to explicitly store all enumerations of n-grams. These features function as a way to combat the "curse" of dimensionality in language modeling (Luong et al., 2015a).

As a result, NLMs are compact and can extend to longer context. As a natural development, subsequent MT systems (Schwenk, 2007; Vaswani et al., 2013; Luong et al., 2015a), inter alia, started adopting NLMs alongside with traditional n- gram LMs and generally obtain sizable improvements in terms of translation quality. To make NLMs evenmore powerful, recent work (Schwenk, 2012; Son et al., 2012; Auli et al., 2013; Devlin et al., 2014) proposes to condition on source words as well as the target context

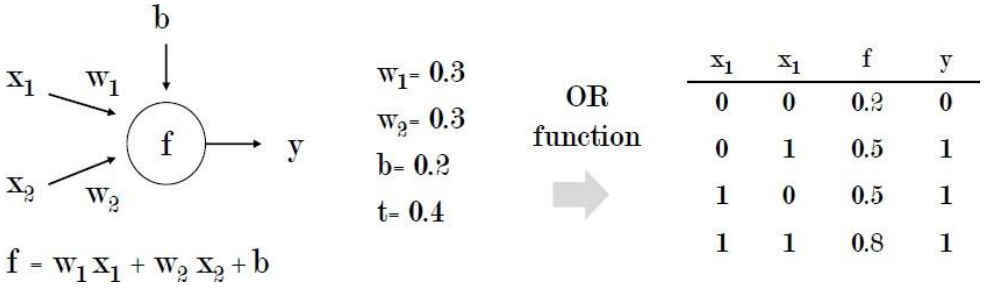
to lower uncertainty in predicting next words. These hybrid MT systems with NLM components, while better than statistical MT systems, still translate locally and fail to capture long-range dependencies. More problematically, the entire MT pipeline is already complex with different components needing to be tuned separately such as translation models, language models, and reordering models. Now, it becomes even worse as different neural components are incorporated in to the translation framework. This inspires the birth of neuralmachine translation with a goal of redesigning the entire MT pipeline completely. To start, we will first learn about recurrent neural network, a building block for NMT as well as a key component to address the local translation problem in statistical MT systems.

4.3 Deep Neural Network

Theoretical results show that deep architectures are required to learn complicated functions (Bengio et al. 2009). A deep architecture consists of many simple computational units, such as in NNs, with many hidden layers and neurons. In this chapter, we discuss the fundamentals of DNNs and explain why

transforming NNs to DNNs is necessary. DNNs involve a huge set of parameters that should be optimized to produce desirable outputs. Clearly, this is a challenging optimization task which complicates the training of DNNs. In this section, we review issues regarding such problems and study prospect solutions. Any machine learning technique modeled for big data and complex problems can be dubbed deep. The common feature of all these models is the presence of a humongous number of computational units, making them suitable for large-scale settings. Although deep learning is not limited to a specific group of models, almost all successful solutions have been implemented by NNs which are distributed computational models drawing from the human brain. They are distributed because an input signal is processed by many intertwined computational processing units (neurons). Neurons are found in layers and connected in sequence to each other via weights. Weights show the connection strength between nodes which are simple mathematical functions. Each NN usually has a cost function so that errors can be computed. An input is passed through various layers and produces an output. The prediction error is computed according to the output and back-propagated to the network to modify weights (network parameters) accordingly. The ultimate goal is to devise an optimal configuration of weights. Afterward, the network can process any random input in order to map it to the output form.

Figure 4 illustrates a single neuron, where it takes two input signals x_1 and x_2 and applies the simple summation function. Input signals are connected to the neuron via weights exhibiting each signal's influence. An optional bias signal b may be added. If the summation result f exceeds a predefined threshold t, the neuron's output y would be 1. Otherwise, it is set to 0. Using such a straightforward mechanism, basic functions like the logical OR can be modeled (Bengio et al. 2009) . For such a function we can set $w_1 = 0.3, w_2 = 0.3, b = 0.2 \; and \; t = 0.4$



x_1	x_1	f	y
0	0	0.2	0
0	1	0.5	1
1	0	0.5	1
1	1	0.8	1

Figure 4: Architecture of a simple neuron to model the logical OR function.

A single neuron can model simple functions but for more sophisticated functions, more complex

structures are needed. As a classic example in machine learning, there is no single-neuron NN capable of modeling the logical XOR function. However, we can easily learn it through a combination of two neurons. As another example, it is impossible to learn $f(x)) = x * sin(a * x + b)$ with a simple NN (Bengio, 2009),), but if we make a network as in figure 5; we can force each unit to model a specific part of $f(x)$. The network on the left-hand side distributes the estimation of the final output over different nodes compelling each all of them to approximate a computation related to a specific part of f. On the other hand, the right-hand side network labors to compute the output via a single unit, which is impossible.

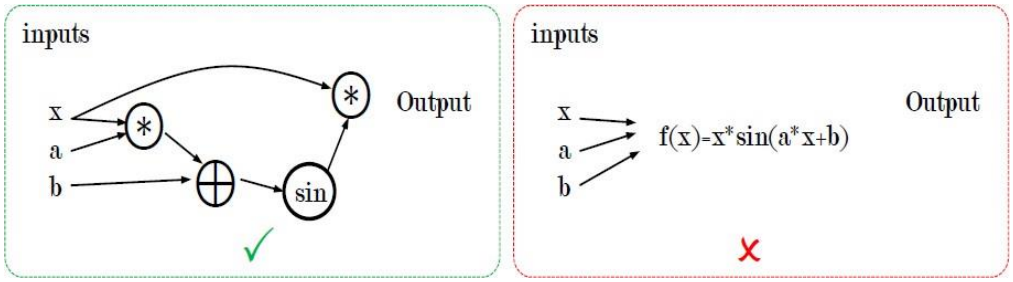


Figure 5: Distributed function approximation using DNNs (Bengio, 2009)

Figure 3 shows the concept behind DNNs. Simple units and architectures cannot provide precise approximations for complex functions. Through a distributional, multi-step procedure, different nodes and units are stacked atop each other to enable NNs to learn functions. To learn a complex data distribution, the network's architecture and a training algorithm oversee the learning procedure and allocate the function approximation over different nodes. Figure 6 illustrates an NN trying to learn a real-world classification problem over a complex data distribution.

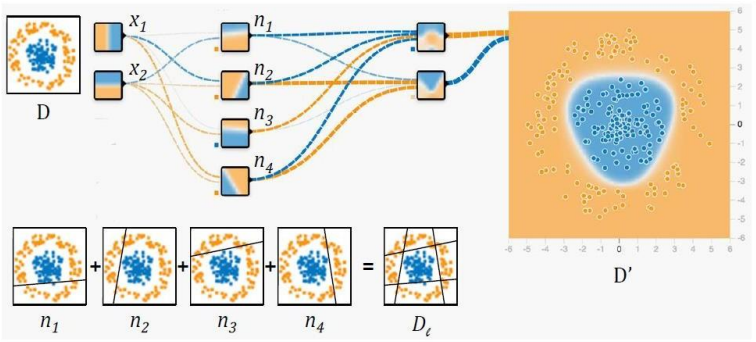


Figure 6: Visualizing the process of data-distribution learning via NNs.

The NN in figure 4 tries to learn the distribution D. D_1 is an approximation learned by the NN specified for

the real distribution. The figure also shows the nodes (magnified) in the first layer, showing which node learns which part of the distribution. A single node in the first layer cannot learn D on its own; however, a combination of nodes can yield an acceptable approximation. $D_1$is a distribution learned by the nodes $n_1$to n_4. Now, we understand the necessity of designing DNNs. In order to have precise approximations, we need several layers to be stacked atop each other. Next, we will discuss how to adjoin layers and interpret various neural architectures. For more information concerning the fundamentals of deep learning, check (Schmidhuber 2015).

4.4 Recurrent Neural Models

Recurrent Neural Networks (RNN) present a powerful and robust type of neural networks. They belong to the most encouraging contemporary algorithms because they are the only ones with an internal memory.

Same with most of the other deep learning algorithms, RNN's are relatively antiquated. They were initially developed in the 1980s, but only reached their real potential a few years ago because of the surge in available computational power thus the massive amounts of data that we have nowadays and the invention of LSTM in the 1990s.

Because of their internal memory, RNN's are capable of remembering important things about the received input, enabling quite the precision in predicting what is coming next.

That is why they are the preferred algorithm for sequential data, e.g., time series, speech, text, financial data, audio, video, weather, and much more; they can form a more profound comprehension of a sequence and its context compared to other algorithms. Recurrent neural networks produce predictive results in sequential data that other algorithms cannot.

A RNN is a MLF with one, or more than one, feedback connection. This means that in RNNs, there is a loop or recurrency connection over one—or more—hidden layer. The loop accumulates the last state or the output of the hidden layer to its input. This is simply formulated as in Equations 2 and 3:

$$h_t = f_h(W_{i:h}x_t + W_{h:h}h_{t-1}) \quad (2)$$

$$y_t = f_0(W_{h:0}h_t) \quad (3)$$

Where $x \in R^n$ is an input vector and $W_{i:h} \in R^{n \times d}$ _d is a weight matrix which connects the input layer to the hidden layer. Recurrency is applied through the $W_{h:h} \in R^{d \times d}$ matrix. $h_t \in R^d$and h_{t-1} indicate the hidden states at the time steps t and t _ 1, respectively. The hidden layer is connected to the next layer $y \in R^m$ through $W_{h:o} \in R^{d \times m}$ $f_h$and $f_o$are non-linear functions which are applied to the input and output of the hidden layer. This architecture is not exactly different from MLFs, but rather a simple extension. Any recurrent network can be converted into an MLF by unfolding over time, so RNNs by nature inherit all mathematical properties of MLFs. Figure 7 gives an example of an unrolled version of an RNN and the recurrence mechanism.

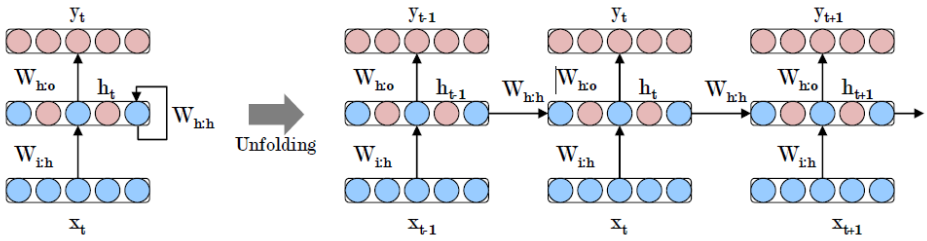


Figure 7: Unrolling an RNN over time

The loop mechanism enables RNNs to accept variable-length sequences as their inputs. Furthermore, at each time step t a summary of all preceding elements before $x_t$resides in hidden states. Clearly, this mechanism is very useful for NLP tasks, which we will discuss in the next chapters. Simple RNNs are not powerful enough to summarize complex structures and capture their properties. They also have problems in remembering long-distance dependencies. To mitigate these shortcomings, extended RNNs with memory units (Sukhbaatar et al., 2015) have been proposed which we also use in our research (see chapter 9).

4.5 Long Short-Term Memory

Long Short-term Memory Networks (LSTMs) (Alex Graves 2013) is a particular type of recurrent neural network that operate on sequential data. Given some input as a sequence of vectors, an LSTM network should return a decision about each vector in the sequence. LSTMs' are designed to address dependencies in long sequences by using a memory-cell to maintain the state of the operations on earlier vectors in the sequence and to prevent vectors from ignoring those through multiple iterations.

Recently, LSTMs have been successfully applied to various tasks, such as speech recognition (Alex Graves 2013; A. Graves and Jaitly 2014), machine translation (M.-T. Luong and Manning 2016; M. Luong and Manning 2015)(M.-T. Luong, Pham, and Manning 2015a), and natural language generation (Wen et al. 2015). In this chapter, we follow the implementation of LSTM as used in (Alex Graves, Mohamed, and Hinton 2013) and (Rei and Yannakoudakis 2016).

The encoder takes as input a sentence S of length n, described by a sequence of vector $X = [x_1, x_2, \ldots, x_n]$. In an LSTM recurrent neural network, input X is processed over time and provides a series of memory states $[c_1, c_2, \ldots, c_n]$ and hidden states $[h_1, h_2, \ldots, h_n]$. To balance the impact of time on hidden states, we process the input X twice, forward and backward, to fully encode the information that the classifier needs..

4.6 Convolutional Neural Models

Convolutional models implement the mathematical convolution operation. They are famous for their good performance in classification problems and ability to extract crucial and related features that help with the classification task. Earlier studies have shown that CNN is an effective approach to extract morphological information e.g., the prefix or suffix of a word, from the characters of words and encode them into neural representations. Convolution is a process on two signals, the input and filter (or kernel), that produces an output. The output is typically considered an altered version of the input, or a non-linear combination of the input and filter. The convolution function is formulated as in formulated as in

$$y_n = \vec{x} \otimes \vec{f} = \sum_{i=-\infty}^{+\infty} x_i f_{n-i} \quad (4)$$

where x, f, and y are one-dimensional signals and y has m elements ($1 \leq n \leq m$). The model assumes that the output signal is a deformed iteration of the input affected by a filter. The filter is applied to highlight and extract certain features of the input. This is the main intuition behind the convolution operation.

Convolutional Neural Networks (CNNs) are usually considered as solutions for complex data structures such as RGB images or natural language sentences. An RGB image is a result of a non-linear (fusional) integration of different complex pixels. A natural language sentence has a parallel structure that

combines words and characters per syntactic and morphological rules. These fundamental units (pixels, words, etc.) are sophisticated because they carry and combine information from various sources. In RGB images, all red, green, and blue sources take part in the composition of a pixel; in sentences, words are affected by various morphological, syntactic, semantic, and contextual limitations (different sources of information).

To draw information from such complex structures, a hierarchical and fusional architecture is needed. This architecture should be able to take simple elements (RGB signals or characters) and combine them to construct basic units (colorful pixels or words). Then a hierarchical mechanism should be utilized to integrate the said basic units to construct the final output (images or sentences). The forward or generation pass is a bottom-up procedure going from very basic elements toward a complex result, while the backward or decomposition path is a top-down procedure which crunches a complex constituent into subunits and divulges information hidden at each hierarchy. CNNs inherently provide such a mechanism.

The computation explained in equation 3 can be extended for 2D or 3D settings, and implemented to different tasks such as image processing (Krizhevsky, Sutskever, and Hinton 2017) or sentence modeling (Kalchbrenner, Grefenstette, and Blunsom 2014). In such CNNs, it is assumed that the state $h_t$is a complex version of the state h_{t-1}. Accordingly, elements in h_t have more complex structures than those of h_{t-1}. This can be viewed as a procedure in which several elements in a layer contribute to make a high(er)-level element in the following layer(s). The procedure is gradually and continuously applied layer-bylayer to reach the final output. The convolution operation is usually followed by a pooling operation, so that the main transformation is applied to an input signal, and then a pooling operation is used to select average, minimum, or maximum values from the convolution's output. Pooling is carried out to attenuate the impact of noise and select high-grade signals.

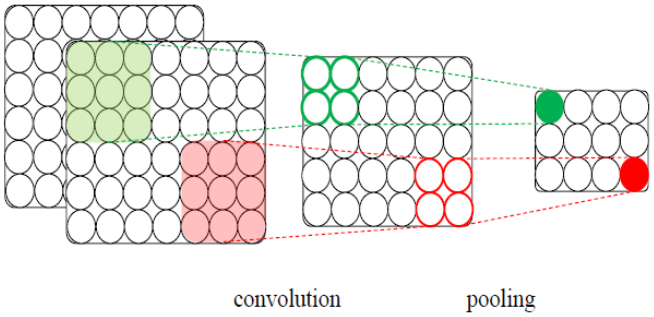


Figure 8: The convolution operation in CNNs.

Figure 8 illustrates the process of gradually reaching a high-level description of an RGB image by use of a CNN. The input data consists of 2D matrices (an input with several channels). A 2D convolution function is applied to different regions (red and green windows) of the input to compose new forms (red and green cells). Applying the convolution function and the filter to all regions creates a convolved signal which encodes information into a far denser structure. Based on the type of filter used, some important properties of the input are also highlighted in the convolved signal. Finally, a pooling operation is applied, e.g., a 2-by-2 max pooling function.

These operations are successively applied to extract high(er)-level representations and reach the final output in the end. The final output could be a word which describes the class of the input image, or any description about it. This pipeline is not exclusive for RGB images, i.e. a natural language sentence can be represented by a cube or a matrix at the input layer. Then a convolution operation is applied. At each layer a new representation of the sentence is generated, and based on the task for which the NN is trained, some features of the sentence are highlighted. For example, in the sentence compression task, redundant words are truncated as the sentence is passed through layers. In chapter 10, CNNs are used for similar (NLP) tasks.

4.7 Dropout

Dropout is a technique that addresses both these issues. It prevents overfitting and provides a way of approximately combining exponentially many different neural network architectures efficiently. The term “dropout” refers to dropping out units (hidden and visible) in a neural network. By dropping a unit out, we mean temporarily removing it from the network, along with all its incoming and outgoing connections. The choice of which units to drop is random. In the simplest case, each unit is retained

with a fixed probability p independent of other units, where p can be chosen using a validation set or can simply be set at 0.5, which seems to be close to optimal for a wide range of networks and tasks. For the input units, however, the optimal probability of retention is usually closer to 1 than to 0.5.

Applying dropout to a neural network amounts to sampling a "thinned" network from it. The thinned network consists of all the units that survived dropout Figure 6. A neural net with n units, can be seen as a collection of 2^n possible thinned neural networks. These networks all share weights so that the total number of parameters is still $O(n^2)$, or less. For each presentation of each training case, a new thinned network is sampled and trained. So training a neural network with dropout can be seen as training a collection of 2^nthinned networks with extensive weight sharing, where each thinned network gets trained very rarely, if at all.

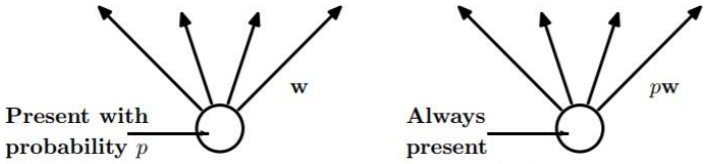


Figure 9: Left: A unit at training time Right: At test time.

At test time, it is not feasible to explicitly average the predictions from exponentially many thinned models. However, a very simple approximate averaging method works well in practice. The idea is to use a single neural net at test time without dropout. The weights of this network are scaled-down versions of the trained weights. If a unit is retained with probability p during training, the outgoing weights of that unit are multiplied by p at test time as shown in Figure 7. This ensures that for any hidden unit the expected output (under the distribution used to drop units at training time) is the same as the actual output at test time. By doing this scaling, 2^n networks with shared weights can be combined into a single neural network to be used at test time. We found that training a network with dropout and using this approximate averaging method at test time leads to significantly lower generalization error on a wide variety of classification problems compared to training with other regularization methods.

4.8 Conditional Random Field (CRF) Networks

For sequence labeling (or general structured prediction) tasks, it is beneficial to consider the correlations

between labels in neighborhoods and jointly decode the best chain of labels for a given input sentence. For example, in POS tagging an adjective is more likely to be followed by a noun than a verb, and in NER with standard BIO2 annotation I-ORG cannot follow I-PER. Therefore, we model label sequence jointly using a conditional random field (CRF) (Lafferty, McCallum, and Pereira 2001), instead of decoding each label independently. Formally, we use $z = \{z_1, z_2, \ldots\ldots, z_n\}$ to represent a generic input sequence where $z_i$is the input vector of the ith word. $Y = \{y_1, \ldots\ldots, y_n\}$ represents a generic sequence of labels for z. $Y(z)$ denotes the set of possible label sequences for z. The probabilistic model for sequence CRF defines a family of conditional probability $p(Y|Z: W, b)$over all possible label sequences y given z with the following form:

$$p(y|z; W, b) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, z)}{\sum_{y' \in Y(z)} \prod_{i=1}^{n} \psi_{i(y'_{i-1}, y'_i, z)}} \quad (5)$$

Where $\psi_i(y_{i-1}, y_i, z) = \exp(W^T_{y',y} z_i + b_{y',y})$ are potential functions , and $W^T_{y',y}$ and $b_{y',y}$ are the weight vector and corresponding to label pair (y', y) respectively.

4.9 Chapter Summary

In this chapter, we explained the fundamental concepts of our research as essential prerequisites of the thesis, studied DNNs, and explained what a DNN is and what types of architectures it might be. Chapter 2 provided the first question to form our research questions. In the next chapter, we study and interpret how we can build and collect training data corpus.

Part II: Design and Implementation
Chapter Five Error Detection and Correction

5.1 Introduction

Spelling error detection and correction is a classical problem. Spelling correction solutions have vital importance for handling the input of a variety of applications and natural language processing tasks, including Optical Character Recognition OCR (Bassil and Alwani 2012), search query processing (Rachidi et al. 2012), pre-editing or post-editing for parsing and machine translation (A. El Kholy and Habash 2010), and intelligent tutoring systems (Shaalan, Magdy, and Fahmy 2010), to name just a few.

A spelling error correction system typically involves two primary modules: detecting errors within the text and correcting those errors (K. Shaalan, Magdy, and Fahmy 2015). The simplest approach for detection is to match the primitive form of the input word against a lexicon's entries. If any given word is not listed in the lexicon, it is considered as an ill-formed word which flags a spelling error. The role of the error correction module is to generate a list of ranked candidates that might be considered as corrections for the erroneous word. In automatic spelling correction, systems usually suggest only one probable word that should be chosen carefully.

Grammatical error correction is a challenging task, and existing methods that attempt to solve or detect this type of error take recourse to in-depth linguistic or statistical analysis (Shaalan 2005). In general, grammatical error correction may partly assist in solving issues related to Natural Language Processing (NLP) tasks like chunking or parsing. Today's grammar error checkers are still far from perfect; even though they are much better and easier to use, they still have limitations that require post-editing. Grammatical errors are usually complicated and require extensive research and linguistic resources for their detection and resolution (Wang, Jia, and Zhao 2014). It can be pretty difficult to find the best solution to every grammatical error. Our goal here is to detect and correct different types of spelling and grammatical errors that commonly occur in Arabic text using recent advances in statistical models, especially in artificial neural networks.

Recent advances in Artificial Intelligence technologies have highlighted Neural Network models achieving great success in various English statistical NLP tasks, such as language modeling (Shi 2011;

Hinton et al. 2012) and speech recognition (Zeyer et al. 2016; Lu and Renals 2017; Hinton et al. 2012). The most recent developments in NLP have found a way of representing words as vectors to measure the distances between words as well as to indicate a sense of similarity and difference between words. Texts of different lengths can be developed as fixed-size vectors by using convolutional networks (Kalchbrenner and Blunsom 2013) or recurrent neural networks (Young et al. 2016; Cho et al. 2014a). Little work has been done in Arabic NLP to determine the relations between a word and its contextual words described in a large vector space using deep learning, a process which can be used to recognize and correct erroneous Arabic text.

In this chapter, we present how Neural Network models can be used for the task of detection and correction of Arabic grammar and spelling errors at the word level. We propose a novel deep-learning framework for performing error detection in Arabic text, which achieves state-of-the-art results on many gold standard datasets that have ill-formed words annotated, validated and manually revised by expert Arabic linguistic specialists. The basic idea is to add error detection and correction as a binary classification with a fixed-size context window. The effects of different datasets on the overall performance are investigated by incrementally providing additional training data to the system. As far as we know, we are the first to employ word-level embedding to build a system for error detection and correction for Arabic texts with extensive expert evaluation.

5.2 Types of Errors

Assigning a single category to a particular error is difficult, especially for a linguistically rich language such as Arabic, in which a one-word surface form may even act as a sentence. Three types of an ill-formed word can be differentiated in a sentence: typographic, cognitive, and phonetic errors. While some grammatical or semantic word errors can also have a typographical origin, these can also have classified into simple errors, i.e., single error misspellings, or multi-error misspellings (Shaalan 2005).

The task of Arabic spelling error detection and correction addresses errors that arise due to editing, adding, splitting, merging, the use of punctuation, orthography variations, and dialectal mixing, among other error types. Figure 10 illustrates examples of these errors along with their source or cause. The words in red belong to errors identified by expert linguistic specialists.

Interestingly, based on a statistical analysis, approximately 80% of all misspelling errors in Arabic and English (Haddad and Yaseen 2007) refer to single error misspelling. In the rest of this section, we discuss all of the causes for sources of error patterns that can be found in Arabic and more generally in any electronic document.

5.2.1Morphological Errors

Morphological errors are usually related to an incorrect derivation or inflection, or an incorrect templatic or concatenative morphology. For knowledge about Arabic computational morphology, we refer the reader to (Habash 2010). Addressing this type of error requires an awareness of the Arabic inflection linguistic rules along with their exception.

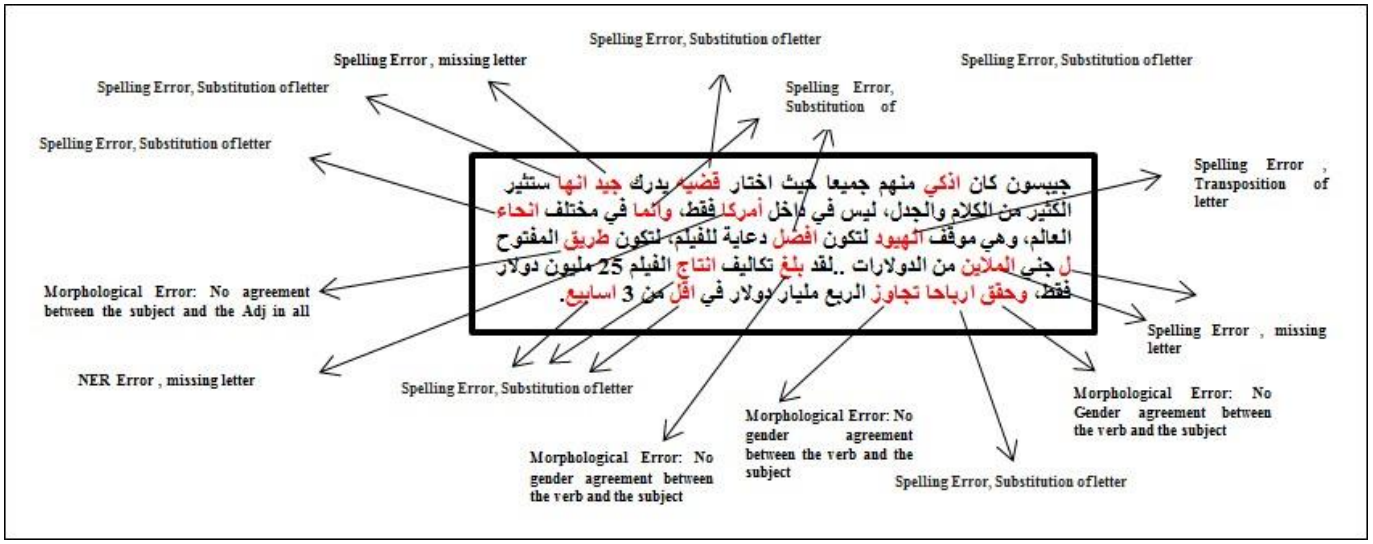


Figure 10: An example of Arabic text with illustrations of the source of errors

5.2.2Spelling Error

Spelling errors usually appear when at least one of the characters in a word is eliminated or substituted with another character, or when an extra character is inserted (K. Shaalan 2003). Some of these errors result in non-words and some result in semantically incorrect words in context. For example, consider an error due to substituting the first letter, the non-word (ظباب, thabab) should be corrected to (ضباب, dabab, fog). Another frequent source of spelling error is due to the merging of two words which generates a non-word. For instance, we consider the two words (رئيس الجامعة , rˆyys Aljameaa~, the president of the

University); when merged together, they result in the incorrect word (رئيسالجامعة r^yysAljamiaa~, the president-of-the-University). The annotator should add a space to split the word in such cases.

The following four types of spelling error are particularly prevalent in Arabic:

1. Hamza] consonant has seven possible lexographic forms which Arabic writers often confuse. The most critical use of Hamza letter (“ الهمزة”) “ء” brings in more challenges. With the very significance of Hamza being an additional letter seen at the top or bottom of the letters following the sounds of “ ا”, “و”, or “ى”, i.e. “ أ”, “ؤ”, or “ ئ”, respectively. As these rules are confusing even for native speakers, Hamza is ignored most of the time while typing. NLP based systems should handle this assumption. There are many orthographical forms of the Hamza letter “the seat of “Al-Hamza”, which is decided by the diacritics (“Tashkeel”) of both the “Hamza” itself as well as the letter preceding it, i.e. either “Fatha”, “Dama”, “kasra” or “Sukun”. Exceptionally, when Hamza comes at the beginning of the word, we always write it over an “Alef”, e.g. “ أنا” (I, 'ana), or under it, e.g. (إيمان, 'iiman ,faith).

   According to its appearance and pronunciation, there are two types of Hamza: (همزة قطع, Hamza Al-Qata’) and (همزة وصل, Hamza Al-Wasl). Distinguishing each type is a challenge for both text and speech processing. Hamza Al-Qata’ is the regular Hamza and is always written and pronounced, e.g. “ إيمان” and ”أنا”. On the contrary, Hamza Al-Wasl is neither written nor pronounced unless it is at the start of the utterance; a bare Aleh is used instead. A simple rule to recognize Hamza Al-Wasl is to add (و, waw, and) before it and see whether or not it is pronounced; hence, written. For example, the Hamza in "إقرأ الكتاب"(iq-ra' AL’Kitab, read the book), is pronounced and written. However, if we add (و, waw, and) at the beginning of sentence as in (واقرأ الكتاب,waq-ra’ Al-Kitab, and read the book) the Hamza is neither pronounced nor written. A more complicated example is (أخذت ابننا, a-khadh-tu ibnana, I grabbed our son). In the first word (أخذت, grabbed), the Hamza is a glottal stop (pronounced

strongly) and should be pronounced, but in the second word "ابننا", it is neither written nor pronounced.

Arabic:	أ	إ	آ	ا	ؤ	ئ	ء
Transliteration:	Â	Ǎ	Ā	A	ŵ	ŷ	'

2. Particular cases of ta-termination (/ ة, tā'- marbūtah) in feminine or broken plural noun and the (/ ت, t) are frequently confused. For example, compare their occurrences in the sentences: دلائل على وجود حياة في أعماق البحر الميت, "Evidence of life deep in the Dead Sea", versus دلائل على وجود حيات في أعماق البحر الميت, "Evidence of a serpent deep in the Dead Sea", which results in a cognitive error because the word sense (حياة, hayāh, life) is different from the word sense (حيات, hayāt, serpent).
3. The Ta-Marbuta (/ ة, tā') and the (/ ه, Ha) are frequently confused; for example, compare (مكتبة, mktb~, library) with مكتبه), mktbh, his office).
4. The Alif-Maqsura (ى, ý) and (ي ,Ya, y) are constantly inverted or confused; for example, compare (علي, Ali) with (على, above).
5. Errors caused by erroneous mechanical pressing of neighboring keyboard keys; for example, (جديد, jaded, new) versus (حديد, hadyd, iron).

5.2.3Syntactic Errors

Arabic syntax is intricate. Automating the process that makes the computer analyze the Arabic sentences is genuinely a challenging problem from the computer perspective. Syntactic errors may arise due to the mismatching or disagreement of syntactic features, e.g., an agreement between constituents in gender, number, person, or definiteness or case, as well as incorrect case assignment, incorrect tense use, incorrect word order, a missing word or redundant/extra words.

Agreement in Arabic is full or partial and is sensitive to word order effects. An adjective in Arabic usually follows the noun it modifies" الموصوف" (almawsuf) and fully agrees with respect to number, gender, case, and definiteness, e.g. " الولد المجتهد" (The diligent boy, alwald almujtahad) and " الأولاد

المجتهدون" (The diligent boys, al'awlad almujtahidin). The verb is marked for agreement depending on the word order of the subject relative to the verb.

The verb in Verb-Subject-Object order agrees with the subject in gender, e.g. " جاء الولد / الأولاد" (came the-boy/the-boys, ja' alwalad/ ja' al'awlad) versus " جاءت البنت / البنات" (came the-girl/the-girls, ja'at albint/ja'at albanat). In Subject-Verb-Object (SVO) order, the verb agrees with the subject with respect to number and gender, e.g. الولد جاء / الأولاد جاءوا (came the-boy/the-boys) versus " البنت جاءت / البنات جئن" (came the-girl/thegirls).

In Aux-subject-verb word order, the auxiliary agrees only in gender while the main verb agrees in both gender and number, e.g. (كانت البنت تدرس الدرس, kanat albint tadrus aldars, , the-girl was studying the lesson), and (كانت البنات تدرسن الدرس, kanat albanat tadrusun aldars, the-girls were studying the lesson. If the subject precedes the auxiliary, then both verbs agree with it in both gender and number (البنات كن يدرسن ,albint kanat takul alttaeam), and (البنت كانت تدرس الدرس , albanat kunn yakuln alttaeam).

Other agreements exist between the numbers and the countable nouns. Number–counted noun agreement is governed by a set of complex rules for determining the literal number that agree with the counted noun with respect to gender and definiteness. In Arabic, the literal generation of numbers is classified into the following categories: digits, compounds, decades, and conjunctions. The case markings depend on the number–counted name expression within the sentence. In the following example, the number, " خمس" (five [masc.sg]) and the (broken plural) counted noun " متاحف" (museums [fem.pl]) need to agree in gender and definiteness:

Arabic grammar distinguishes between two types of sentences: verbal and nominal. Verbal sentences usually begin with a verb, and they have at least a verb ("فعل", faeal) and a subject ("فاعل", faeil). The conjugation of the verb can indicate the subject as well as the object, and not written separately. For example, the conjugated verb (سمعتك, sami'tuk, I heard you, Heard-I-you) has a subject and an object suffix pronoun attached to it. Another example of a verbal sentence is (يلعب الولد, yala'b alwald, playing the boy). This type of sentence is not applied in English sentences. All the English sentences begin with a subject, and followed by a verb, for example, "the boy is studying".

In Arabic, a nominal sentence begins with a noun or a pronoun. The nominal sentence has two parts: a subject or topic "مبتدأ", (mubtada) and a predicate " خبر", (khabar). The nominal sentences have two types: with or without a verb. The nominal verbless sentence is a typical noun phrase.

When the nominal sentence is about being, which in some languages such as English requires the presence of the linking verb 'to be' (i.e. copula) in the sentence. This verb is not given in Arabic. Instead, it is implied and understood from the context. For example, (الطقس جميل, alttaqs jamil) has two nouns without a verb; its English translation is "The Weather [is] wonderful". This can be confusing to second language learners who speak European languages and are used to have a verb in each sentence. Arabic grammar allows complex sentence structure formation which is discussed in the following subsections.

5.2.4Named Entity Recognition (NER) Errors

Arabic has no different sign rendering the recognition of a Named Entity (NE) all the more difficult. On the other hand, English, in line with numerous other Latin script-based dialects, has a particular marker in orthography, in particular, the upper casing of the underlying letter, and showing that a word or succession of words is a named substance. Arabic does not have capital letters; this trademark speaks to a massive hindrance for the essential task of Named Entity Recognition because, in different languages, capital letters speak to a vital highlight in distinguishing formal people, places or things. Along these lines, the issue of identifying appropriate names is especially troublesome for Arabic. For instance, in English, capital letters are used, e.g., "Sarah," but no capital letter in the same name in Arabic, e.g. " سارة"Another reality about Arabic to consider is that the vernacular has no capital letters (e.g., for proper names: the names of people, countries, months, days of the week); therefore, cannot make use of acronyms. This can lead to confusion, especially during Information Extraction in general and Named Entity Recognition in particular. It makes it difficult to see the names of substances. For example, the NE "الامارات العربية المتحدة " has the acronyms UAE in English but not in Arabic. Therefore, it is common to resolve the nonappearance of capital letters by analyzing the context surrounding the Named Entity. Because of that, NER errors appear in the spelling of persons, organizations, and locations, especially those of foreign origin which could be wrongly transliterated or that are transliteratable in different ways (K. Shaalan 2014b).

5.2.5 Punctuation Errors

Punctuation errors must be corrected according to the most commonly accepted Arabic punctuation rules. Moreover, aspects of punctuation use differ from author to author and can be considered a choice. While punctuation in the English or European language is managed by a series of set grammar-related rules, in another language such as Arabic, punctuation is a recent addition as pre-modern Arabic did not use punctuation (Zaghouani, Zerrouki, and Balla 2015) . According to (Awad 2015), there is an inconsistency in the punctuation rules and usage in the Arabic language and ignoring punctuation marks is a persistent error. Punctuation errors are often present in student essays and online news comments, mainly because some punctuation mark rules are not clearly outlined in Arabic writing references.

As mentioned earlier, the different types of errors can only be detected using in-depth (syntactic, semantic and statistical) knowledge. We, therefore, combine a statistical-based approach with a lexical, morpho-syntactical and heuristics knowledge-based approach designed to detect and correct various types of errors proceeding from the characteristics of Arabic word analysis.

5.3 Related Work

5.3.1 Arabic Corpus

Most research in Spell Checking attempts to create an Arabic corpus which may include words from a specific domain or from more than one domain. In this section, we present the efforts to collect a large-scale Arabic corpus for misspelling errors. To the best of our knowledge, there is no Arabic corpus for grammatical errors.

Al-Jefri and Mahmoud created a large corpus collected from Al-Riyadh newspaper articles on three topics, health, economics, and sports, with a total number of (4,136,833), (24,440,419) and (12,593,426) words each, acquired from (7,462), (49,108), and (50,075) articles, respectively. These combine to form a general corpus composed of (41,170,678) words. In their model, they organized a sample of confusion set from non-native Arabic speakers, and an Arabic OCR system which showed that not all not types of errors were covered (Al-Jefri and Mahmoud 2015).

Alfaifi et al. created the Arabic Learner Corpus (ALC)[2].This open-source corpus was developed at Leeds University and is comprised of 282,732 words collected from learners of Arabic in Saudi Arabia over the course of 2012 and 2013. The corpus includes written and spoken data produced by 942 students from 67 different nationalities studying at pre-university and university levels (Alfaifi, 2013).

Attia et al. developed an Aracomplex extended corpus filtered from accepted words normalized by omitting diacritics, numbers, symbols, punctuation marks and English letters. These filtered and normalized words were then processed through a Microsoft spell checker generate a list of 9,306,138 words. This list was used to check and replace [correct] misspelled words. This is the most extensive corpus for Arabic spelling detection and correction that can be integrated with text processing tools. Making searches of this significant corpus efficient is essential, but unfortunately, the tool used to create the AraCompLex list[3] is not available (Attia et al., 2015).

Alkanhal et al. state that the aim of compiling the Learner Corpus of Arabic Spelling Correction was to build and test a system developed to automatically correct misspelled words in Arabic texts. The corpus consists of 65,000 words that were manually revised for spelling to annotate all of its misspelled words. This data covers diverse essays written by students studying at two universities. The Learner Corpus includes two sources of errors: real spelling mistakes generated by the students, and transcription mistakes generated by the transcribers (Abandah et al. 2015).

Farwaneh and Tamimi created the Arabic Learners Written Corpus (ALWC). Materials were produced by non-native Arabic speakers from the USA and were collected over a period of 15 years. This corpus contains only 50,000 words, organized into three levels (beginner, intermediate and advanced) and three text classes (descriptive, narrative and instructional). It was formed over two phases. The purpose of the first phase was to provide a data source for hypothesis testing and for generating teaching materials, while in the second phase, the corpus was designed to be tagged for morphological, syntactic and orthographic errors as well as for the characteristics at each level. The ALWC is only available for download in PDF format files (Farwaneh and Tamimi 2012).

[2] http:// arabiclearnercorpus.com
[3] http://aracomlex.sourceforge.net

5.3.2 The Computational Model

As far as the techniques are considered, systems designed for error detection and correction tasks utilize relevant language resources such as textual corpora and dictionaries.

Haddad and Yaseen proposed a hybrid model for non-word detection and correction. Their hybrid approach uses morphological knowledge in the form of consistent root-pattern relationships, and some morpho-syntactical knowledge based on affixation and morpho graphemic rules to define the word recognition and non-word correction process. The main goal of this work is to develop a context-dependent syntax and a semantic checker (Haddad and Yaseen 2007).

Hamza et al. developed an independent spell-checking corpus that includes ill-formed words recognized from the failure of the morphological analysis. Although it utilizes a stem dictionary to reduce the size of a significant amount of Arabic words, it did not discover all types of errors (Hamza et al. 2014).

Al-jefri & Mahmoud proposed two approaches: context word and n-gram. These approaches can easily manage errors caused by widely used confusing words. But, such approaches can only discover particular predefined errors that are presented in the form of confusion sets. Hence, an extension is suggested to increase the number of confusions sets to solve most of the detected errors. Although the experimental results of the techniques applied in this study display promising correction accuracy, it is not possible to link the results of this study with those of previous works since they did not make their benchmarking datasets available (Al-Jefri and Mahmoud 2015).

5.4 Background on The Proposed Model

Detecting and correcting errors is one of the main problems that faced NLP researchers from an early stage. In this section we define the neural network technique for error detection and correction and provide information about some approaches.

5.4.1 Problem Statement and Formulation

The aim of the proposed error detection model is to identify misspellings and grammar errors at the word level. For example, consider the following two sentences where our error detection system is expected to correctly recognize the erroneous word (نقطة, point) highlighted by an underline: (تلخص النجاح في عشر نقطة, successes summarized in ten point), and the erroneous word (فلاعاً, falaan) highlighted by an underline

(معظم الدول العربية تعتبر اليوم فلاعاً للحرية, most Arabic countries today are considered a falaan of freedom).

The task of word-level misspelling and grammatical error detection is adopted from (Rei and Yannakoudakis 2016) as follows: given a sequence of tokens as input, $X = [x_1, x_2, \ldots, x_n]$, the error detector outputs its prediction $Y = [y_1, y_2, \ldots, y_n]$ where y_i indicates the correctness of x_i in terms of grammaticality and misspelling errors.

We tackle this problem as a binary classification problem. To predict y_t given the current word x_t and the whole sentence$X = [x_1, x_2, \ldots, x_n]$, we need to determine a function $g[\cdot]$ to compute the conditional probability of each y_t given $x_t$and the whole input sequence X:

$$p[y_t|x_t] = g[x_t, X], \qquad (5)$$

where

$$y_t = \begin{cases} 1, & if\ x_t\ is\ correct \\ 0, & otherwise \end{cases}$$

Our goal is to build a convenient classification model for $g[\cdot]$.

5.4.2 Polynomial Networks Model for Error Detection

A natural approach is to use Polynomial Network (PN) to perform a classification (Alkhatib, Monem, and Shaalan 2017a). A PN is trained given a training dataset in the form of $\{[x_1, y_1], \ldots, [x_n, y_n]\}$, where x_i denotes a token with a set of selected linguistic features, and y_i indicates the (level of) grammatical or misspelling correctness of the token. The PN finds a maximum-margin hyperplane that separates correct words from incorrect ones.

The difficulty with this approach is that we must manually design features in x_i. Since humans are unable to specify precisely which features are relevant, human-designed features are inadequate in some cases while being redundant in others. As a result, these designed features are unable to accommodate all regularities, which might hurt the performance of our proposed error detector.

5.4.3 Convolution Network through Fixed Window Size

To avoid the problem with feature engineering, one natural approach is to utilize the capability of neural

networks in automatic feature extraction (Rei and Yannakoudakis 2016). The best way is to consider a fixed-size window of words around the current word as its context by utilizing temporal convolution with that fixed-size window. We explain our proposed model using a convolution network with fixed window size below.

In the first example sentence given in subsection 4.1, when considering the grammatical correctness of the word "عشر, ten" given a context window of size 3, the context window would be "عشر نقطة, ten point." The assumption that underlies this method is that only nearby words are grammatically related to the current word.

Here we formalize the method of the neural network with a fixed-size window**.** Given a word x_i, its context is:

$$c_i = [x_{i-w/2}, \ldots, x_i, \ldots, x_{i+w/2}]. \qquad (6)$$

Let $f[\cdot]$ indicate a temporal convolution operation with the input frame size equal to the dimension of x_i, the output frame size equal to 1, and the kernel width equal to the size of a fixed-size window. A score s_i of the current word x_i is calculated by $s_i = f[c_i]$, which represents the grammatical features within the window. This score then goes into a sigmoid layer and yields the probability of y_i: $p(y_i | x_i, c_i) = \sigma[s_i]$. (Liu and Liu 2017)

The first difficulty with this method is that it is ineffective at capturing long-distance dependency. With a fixed window size, the error detector is unable to take into consideration word contexts beyond the window size, while long-distance grammatical dependency is a very common phenomenon. For instance, to discover whether "summarized" is wrong, we would need to take "successes" into consideration, which needs a large context window.

5.4.4 Model Architecture

In one-hot preparation phase, we replace each word in the sentence by its corresponding word embedding from a pre-trained distributed word representations. We use the AraVec (Soliman, Eissa, and El-Beltagy 2017) word embedding which was pre-trained on Arabic data following Word2vec (Mikolov et al., 2013) approach. Each Arabic sentence is represented by a $2D$ vector of dimension $n \times d$. Where n is the

number of words in the sentence and d is the dimension length of the word's vector representation.

We use the skip gram model of dimension 300. i.e.$d = 300$. In order to make ensure that all of the sentences are having the same fixed size, we follow the same approach in (Liu and Liu 2017b), by padding each sentence's representation by zeros. In this way each sentence will be of size $n' \times d$, where we chose n' to be 40.

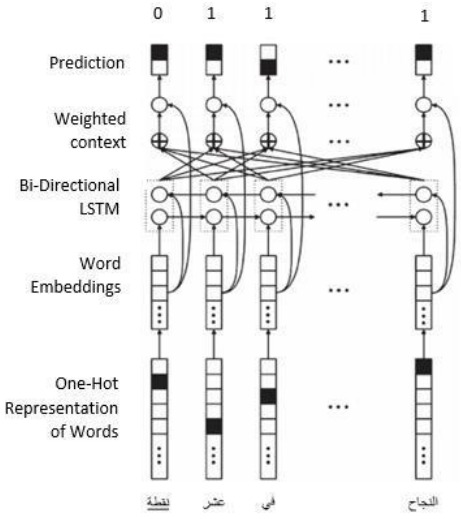


Figure 11: Our Model Architecture

We first encode the input sequence into a sequence of hidden states that contain relevant grammatical information and spelling lists, and then make predictions for words and their context (see figure 11). Thus, our model consists of two parts: an encoder that adopts the architecture of a Bi-Directional LSTM network (Mikolov et al., 2013), and a classifier that makes predictions based on the hidden states of the encoder. The input of the network is a sentence, such as (تلخص النجاح في عشرنقطة, successes summarized in ten point), in the form of one representation. The representation is then transformed into continuous word embeddings and encoded by a Bi-Directional LSTM encoder. The encoded information is reweighed by an intra-attention mechanism at each time-step, based on which the classifier determines the grammatical and the spelling errors of each word.

5.4.5 Word Embeddings

Word Embeddings (Lebret, Grangier, and Auli 2016) refers to a model for feature learning used in NLP tasks to transform input words into a classification representation of real numbered vectors. The model

results in the construction of low-dimensional vector representations for all words in a given input text corpus. Words sharing ordinary contexts in that corpus are transformed into vectors that are close to each other in the vector space. As such, it can be stated that word embeddings produce an artistic representation for words, as it captures their semantics. Word embeddings are usually constructed from huge corpora, so it is very beneficial in these tasks to make use of pre-trained word embeddings in order to have a generalized model with a reliable estimation of the problem parameters.

5.4.6 Long Short-Term Memory

Long Short-term Memory Networks (LSTMs) (Alex Graves 2013) is a particular type of recurrent neural network that operate on sequential data. Given some input as a sequence of vectors, an LSTM network should return a decision about each vector in the sequence. LSTMs' are designed to address dependencies in long sequences by using a memory-cell to maintain the state of the operations on earlier vectors in the sequence and to prevent vectors from ignoring those through multiple iterations.

Recently, LSTMs have been successfully applied to various tasks, such as speech recognition (Alex Graves 2013; A. Graves and Jaitly 2014), machine translation (M.-T. Luong and Manning 2016; M. Luong and Manning 2015) (M.-T. Luong, Pham, and Manning 2015a), and natural language generation (Wen et al. 2015). In this chapter, we follow the implementation of LSTM as used in (Alex Graves, Mohamed, and Hinton 2013) and (Rei and Yannakoudakis 2016).

The encoder takes as input a sentence S of length n, described by a sequence of vector $X = [x_1, x_2, \ldots, x_n]$. In an LSTM recurrent neural network, input X is processed over time and provides a series of memory states $[c_1, c_2, \ldots, c_n]$ and hidden states $[h_1, h_2, \ldots, h_n]$. To balance the impact of time on hidden states, we process the input X twice, forward and backward, to fully encode the information that the classifier needs.

The forward LSTM updates its memory state $\overrightarrow{c_i}$ and its hidden state $\overrightarrow{h_i}$ at each time-step t:

$$\left[\overrightarrow{h_t}; \overrightarrow{c_t}\right] = \overrightarrow{LSTM}\left[\overrightarrow{h_{t-1}}; \overrightarrow{c_{t-1}}\right] \qquad (7)$$

Similarly, memory state $\overleftarrow{c_i}$and hidden state$\overleftarrow{h_i}$ are updated by the backward LSTM at time-step t:

$$\left[\overleftarrow{h_t}\,;\overleftarrow{c_t}\right] = \overleftarrow{LSTM}\left[\overleftarrow{h_{t+1}}\,;\overleftarrow{c_{t+1}}\right] \qquad (8)$$

The encoder outputs a hidden state $\tilde{h} = [\widetilde{h_1}, h_2, \widetilde{\dots\dots}, \widetilde{h_n}]$, with $[\cdot]$ denoting the concatenation of vectors.

6.5.3.1 A Classifier with Intra-Attention

To predict whether the word at time-step t is grammatically or spelling problematic, the classifier computes a score given the current word x_t and its context a_t. This score s_t then goes through a sigmoid layer and makes a binary prediction, with 1 denoting a correct word and 0 denoting an incorrect word. Note that the classifier does not hold its state as a decoder does in a traditional encoder-decoder architecture (Huang, Xu, and Yu 2015).

To tackle the problem of long-distance dependency, we incorporate an intra-sentence attention mechanism ("Bahdanau et Al. - 2014 - Neural Machine Translation by Jointly Learning to .Pdf," n.d.) in our classifier, in which all the hidden states of the encoder are taken into consideration, and thus the attention of the classifier is dynamically adapted at all positions of the sentence. To express this formally, we calculate the context a_t around the word x_t as an attention weighted sum of $\{\widetilde{h_1}, h_2, \widetilde{\dots\dots}, \widetilde{h_n}\}$:

$$a_t = \alpha_{t,i} \cdot \widetilde{h_i}\,,$$

where

$$\alpha_{t,i} = \frac{exp[E_{t,i}]}{\sum_j exp[E_{t,i}]},$$

$$E_{t,i} = \widetilde{h_t}, \widetilde{h_i}$$

Vector a_t represents the grammatical and semantic context at position t. A word is considered to be wrong if the word x_t does not fit into the current context, i.e., it is incompatible to place x_t at position t given the context a_t. The score s_t is computed as follows (Liu and Liu 2017):

$$s_t = x_t^T \cdot W \cdot \mathrm{a}_t + b, \qquad (9)$$

where b is the bias.

The probability p can then be computed as

$$[y_t|x_t, \{x_1, x_2, \cdots, x_n\}] = \sigma[s_t]. \quad (10)$$

By incorporating an intra-attention mechanism, we give a latent structure for the model, allowing it to determine grammatical relations between words. This works because the grammaticality of a word is more dependent on the words that have strong grammatical relationships with it, while other words are negligible when making predictions. For example, in fig.2, when the model tries to determine if "نقطة, point" is correct in terms of the number of the noun, it will pay more attention to "عشر, ten", which indicates that the counted noun "نقطة, point" should take its plural rather than its singular form.

6.5.3.2 Noise Generation

We adopt the concept of using artificial errors for training purposes. It is important to find an appropriate algorithm for the error generator to generate realistic grammatical and spelling errors, as the performance of the model relies strongly on the paradigm it recognized during training. Since our task is to detect grammatical errors on the word level, we only consider substitution errors for this type of error. We compare two ways of substituting a correct word for an erroneous one.

6.5.3.3 Uniform Random Substitution

The easiest way is to substitute a word in a random position with a random word from the vocabulary. The difficulty with this approach is that some artificial errors formed in this way are clearly irrelevant. For example, it could substitute a word from the sentence "The successes summarized in ten points." To generate such a sentence as" the successes summarized the contract in ten points." one potential problem is that it might be too simple for our classifier to distinguish such erroneous words from the correct ones.

5.4.7 Substitution with Linguistic Knowledge

We carefully examined several erroneous paradigms and exposed some characteristics common to all grammatical errors, despite variations in the terminology and commonly observed patterns of the domain. Grammatical errors ordinarily appear when a correct word is substituted by another word; these arise from a finite set of words linguistically linked to the substituted word because this set of words maintain the same

lemma or the identical part-of-speech (POS)tag.

Combining the two methods, uniform random substitution and substitution with linguistic knowledge, we are able to produce 16 forms of grammatical errors out of the 27 specified by a CoNLL -2014 shared task (Ng et al., 2014.). These 16 error types: (Verb Tense, Verb Model, Verb Form Missing Verb, Subject-Verb Agreement, Article or Determiner, Noun Number, Noun Possessive, Pronoun Form, Pronoun Reference, Preposition, Wrong Collocation/Idiom, Word Form, Parallelism, Linking Words/Phrases, and Duplication, include Morphological and Syntactic Errors). Most of the remaining error types that we are unable to produce are semantic errors (Dangling modifiers, Redundancy, Unclear meaning), style problems (Acronyms, Citations), or sentence-level problems (Sentence fragment, Incorrect word order, Incorrect adjective/adverb order). The details of our artificial error generation process that incorporates linguistic knowledge are described in Algorithm 1, which formalizes the construction of substitution set, and in Algorithm 2, which formalizes the process of error generation by using the substitution set built in Algorithm1.

5.5 Implementation

Our system operated well in some cases while it failed to recognize others. We conducted an experiment to show the differences between our system and the available tools. To analyze what type of errors it handles very well, and the reasons that cause its failure, we sampled some predictions. The Arabic text shown in fig. 1 was used as an input to this experiment. The results are shown in table 9 . As shown in Table 1, both Ayaspell version 3.4 and Microsoft Office 2013 detect spelling errors and typographical errors, while they do not cover the detection of Arabic grammatical errors. Note that the table only contains a partial list of the error types our model detected. Since we only detect errors without indicating their types, we are unable to provide the full list of the error types our model can detect in the table

Algorithm 1. Building Substitution Set

PoS-tag the input text
Build a dictionary $\mathbb{D}$ of $(token, PoS-tag)$
for all $(token, pos)$ in $\mathbb{D}$ **do**
  if *pos* in {CC} **or** {DT, PDT} **or** {PRP, PRP$} **or** {IN, TO, RP} **or** {WDT, WP, WP$, WRB} **then**
    Add *token* to the corresponding substitution set $\mathbb{C}_i$
  else if *pos* in {NN, NNP, NNPS, NNS} **or** {VB, VBD, VBG, VBN, VBP, VBZ} **then**
    lemma ← Lemmatise *token*
    Add *token* to the corresponding substitution set $\mathbb{C}_i$
  else if *pos* in {JJ, JJR, JJS} **or** {RB, RBR, RBS} **then**
    stem ← Stem *token*
    Add *token* to the corresponding substitution set $\mathbb{C}_i$
  end if
end for

Algorithm 2. Error Generation

for all sentences S in training text **do**
  Get word w at a random position of S
  $w' \leftarrow w$
  Search for substitution set $\mathbb{S}_i$ that contains w
  if such $\mathbb{C}_i$ does not exist **or** $\mathbb{C}_i$ contains only 1 element **then**
    while $w' == w$ **do**
      $w' \leftarrow$ Select a random word from dictionary $\mathbb{D}$
    end while
  else
    while $w' == w$ **do**
      $w' \leftarrow$ Select a random word from $\mathbb{C}_i$
    end while
  end if
  Replace w in S with w'
end for

Table 8: Comparative results from three tools that detect Arabic text error

Ayaspell version 3.4	جيبسون كان اذكي منهم جميعا حيث اختار قضيه يدرك جيدا انها ستثير الكثير من الكلام و الجدل، ليس في داخل أمريكا فقط، وانما في مختلف إنحاء العالم، و هي موقف اليهود لتكون افضل دعاية للفيلم، لتكون الطريق المفتوح ل جني الملايين من الدولارات.. لقد بلغ تكاليف انتاج الفيلم 25مليون دولار فقط، وحقق ارباحا تجاوز الربع مليار دولار في اقل من 3 اسابيع.
Microsoft Office 2013	جيبسون كان اذكي منهم جميعا حيث اختار قضيه يدرك جيدا انها ستثير الكثير من الكلام والجدل، ليس في داخل أمركا فقط، وانما في مختلف انحاء العالم، وهي موقف اليهود لتكون افضل دعاية للفيلم، لتكون طريق المفتوح ل جني الملاين من الدولارات.. لقد بلغ تكاليف انتاج الفيلم ٢٥ مليون دولار فقط، وحقق ارباحا تجاوز الربع مليار دولار في اقل من ٣ اسابيع.
Our system	جيبسون كان اذكي منهم جميعا حيث اختار قضيه يدرك جيدا انها ستثير الكثير من الكلام والجدل، ليس في داخل أمركا فقط، وانما في مختلف انحاء العالم، وهي موقف اليهود لتكون افضل دعاية للفيلم، لتكون طريق المفتوح ل جني الملاين من الدولارات.. لقد بلغ تكاليف انتاج الفيلم ٢٥ مليون دولار فقط، وحقق ارباحا تجاوز الربع مليار دولار في اقل من ٣ اسابيع.

5.6 Linguistic Resources

5.6.1 The Training Phase

We relied on the mentioned four available corpora in chapter five for the training phase, after omitting the redundant words from the list. The concept of data cleanliness and its impact on machine learning processes has been discussed in the literature (El-Haj, Kruschwitz, and Fox 2011), with the conclusion that data will tend to be noisy, incomplete and inconsistent, and therefore should undergo some sort of cleaning or preparation. So, the first step of the training phase is to measure the amount of noise in each data subset. We selected sentences that end with a period, with a length that ranges from 5 to 50 words, and which may contain several clauses separated by commas, colons or semicolons. Formulae and references are excluded, numbers are substituted with a particular “num” token, and parentheses are

removed together with their content. In order to measure the cleanliness of our training data, we created a list of the most common grammatical and spelling errors, which covers around 150 errors, including all types of morphological errors, spelling errors, syntax errors, punctuation errors, and NER errors. This list of errors was annotated manually using a human taxonomy expert in language acquisition**.**

We created an extensive word list for this research that includes around 15 million Arabic words from diverse resources, such as the publicly available "Arabic word list for spell checking," a precious resource that fits well with our study (Attia et al., 2015). That list contains 9 million Arabic words including 30,587 lemmas, processed using AraComLex, an open-source finite-state transducer. It was validated against the Microsoft Word spell checker tool in order to find a replacement for each misspelled word.

5.6.2 The Evaluation phase

We evaluated our system using datasets developed by Open Source Arabic Corpora (OSAC)[4] (Saad 2010) that contain two newswire corpora. The first is the "BBC Arabic corpus" from the BBC Arabic website bbcarabic.com. This corpus contains 4,763 text documents. Each document refers to one of seven categories (Middle East News 2356, World News 1489, Business & Economy 296, Sports 219, International Press 49, Science & Technology 232, and Art & Culture 122). The corpus contains 1,860,786 (1.8M) words. The second corpus is the "CNN Arabic corpus", collected from the CNN Arabic website cnnarabic.com, and contains 5,070 text documents. Each document belongs to one of six categories (Business 836, Entertainments 474, Middle East News 1462, Science & Technology 526, Sports 762, and World News 1010). The corpus contains 2,241,348 (2.2M) words.

The dataset has been manually error annotated using an expert linguistic during language acquisition. The combined dataset includes manually-annotated error spans of several types of errors, together with their recommended corrections. We converted this dataset to a token level error detection task by labeling each token inside the error measure as wrong.

5.7 Evaluation and results

In a real-life setting, the proportion of sentences that contain grammatical or misspelling errors depends

[4] http://sourceforge.net/projects/ar-text-mining/files/Arabic-Corpora

on the language proficiency of the persons who produce those sentences. We combined grammatical sentences with misspellings and ungrammatical ones with various ratios to form several test sets and evaluated our model on them. Each sentence contains at least one grammatical and one misspelling error.

We conducted experiments that evaluate our system and compared its results with the output of two well-khnown tools: Ayaspell version 3.4[5], and Microsoft Office 2013. The AraComlex tool has been removed from the experiments as it is not publicly available, along with the other (unavailable) tools.

We use F-measure metric, the foremost evaluation measure for error detection, which was also the measure adopted for other error correction tasks. It combines both Precision and Recall scores while assigning twice as much weight to Precision, since accurate feedback is often more important than coverage in error detection applications (Ng et al., 2014). Our system achieved the best accuracy, 93.89%, and performed very well compared to the other tools, as Microsoft Office 2013 achieved only 53.8%, and Ayaspell version[5] 3.4 achieved 76.4% (see Table 10)

Tool	**F-measure**
Ayaspell version 3.4	76.4%
Microsoft Office 2013	53.8%
Our System	93.89%

Table 10: performance of our model on specific types error measures by F%

Table 11 presents the result: precision (P) and recall (R), of the experiments on our model (BiLSTM), focusing on limited number of errors types (word form, noun number, verb form, and verb tens). The model's performance is better previous experiments on the test set. we focus heavily on these common types of errors and the other types of errors are neglected.

Method	**P**	**R**	**F0.5**
BI-LSTM	95.6	94.88	95.19

Table 10: performance of our model on specific type's error measures by F%

[5] http://ayaspell.sourceforge.net

To evaluate our model, we did the same experiment but, we removed the intra-attention that helps the model performance. The comparison of two models with / without attention is shown in Table 12.

Although the intra-attention mechanism is improving the performance of our model over all,
but it fails in some individual cases.

Method	P	R	F0.5
BI-LSTM	91.3	90.6	90.94

Table 11: performance of our model on without intra-attention measures by F%

5.8 Chapter Summary

We have shown that our proposed neural networks model, which uses bidirectional LSTMs, Word Embeddings and a Classifier, achieves state-of-the-art results in building an Arabic error detection and correction system. One of the ways of improving machine translation outputs is by performing the task of error detection and correction through pre and post-editing, which nowadays, is becoming a common practice in machine translation. To the best of our knowledge, we are the first to explore the impact of word embeddings and of a PN classifier to develop a deep learning system for Arabic Error Detection and Correction for Arabic Text. We created a list of the most common grammatical and spelling errors, covering around 150 errors, which includes almost all types of morphological, spelling, syntax, punctuation, and NER errors. This list of errors was annotated manually using a human taxonomy expert in language acquisition. We also developed the largest (publically-available) corpus of 15 million fully-inflected Arabic words [types] validated and manually revised, with an F-measure 93.89%. The proposed model showed a considerable advantage in terms of results when compared to two well-known systems.

Chapter Six: Building Arabic WordNet from Al Hadith Al Shareef

6.1 Introduction and Background

WordNet (WN) is a linguistic resource that consists of words interconnected by their meaning, through lexical (i.e. single words connected to one another word) and semantic-conceptual relations that may be expressed by more than one word (G.A. et al. 1993). Wordnet contains words along with their classes (nouns, verbs, adjectives, or adverbs), roots, and concepts (synsets), in addition to the relations among these concepts. These relations provide semantic information about concepts and their original words. These concepts and their relations are exploited to improve Arabic Information Retrieval, Text Classification, and Text Summarization

Technically, WN is a lexical database that is made up of words interconnected by their meanings. WordNet consists mainly of two parts:

- *Synsets*: Arabic words are structured by WN into sets of synonyms, so-called *synsets*, the smallest unit of the WordNet lexicon. When a word contributes in several synsets, the relationship between those words is called a "polysemy". WordNet quantifies this relationship by a frequency score that acts as a weight. Synsets are obtainable in their order of frequency.
- *Semantic relations*: Most synsets are connected to other synsets through semantic relations. These semantic relations are between categories of word vocabulary, such as synonymy/antonymy, hypernymy/hyponymy, and meronymy/holonymy. For example, Figure 12 displays that the word أخ Akh "brother" , and the word أخت Okht "sister," are antonyms , the term ألجزاء العضوية "organic substance" is a hyponym of the word جزء "substance", and the two words ذراع "arm" and رجل "leg" are meronyms of the word الجسد "body" (Sun, Huang, and Liu 2011).

All Islamic scholars are interested in the examination of the Al-Hadith Al Shareef, which contains all of the transmitted reports on the authority of the Messenger of Allah (Prophet Muhammad, peace and blessings upon him), including the traditions of Islam, its rulings, rewards, punishments, motivations and admonishments, as well as other descriptive topics. It has been communicated through chains of narration which were related by and circulated between Islamic scholars, the so-called أهل العلم "Ahl ul- Ilm".

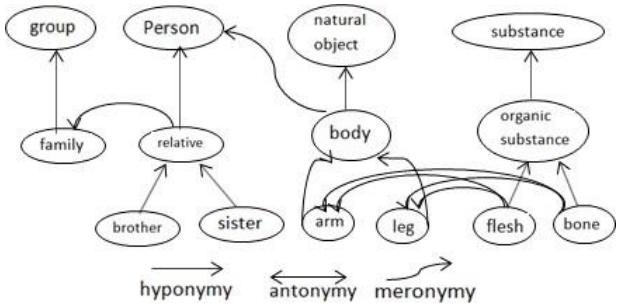


Figure 12. The semantic relation in WordNet

In fact, Al-Hadith Al-Shareef helped Arabic language to obtain the universal status which it has continued to enjoy since the middle ages, developing as one of the main world languages.

The proposed Al-Hadith WordNet services two purposes. It allows anyone who seeks to increase his/her understanding of Islam to also expand their knowledge of classical Arabic vocabulary and the deep meaning of words. This is realized by the rich semantic relationships specified by Al-Hadith WordNet. One important semantic relationship is synonymy. Apparently, in AL-Hadith Al-Shareef, one can find many words that are conceptually synonyms, but if we look up their meanings in a dictionary, we figure out differences in their precise meaning. For instance: the words: مَرْضاةِ *mardat*، سَبِيلِ *sabīl* and وَجْهِ *wağh* are synonyms of the concept of how Muslim people spend their wealth for seeking the pleasure of Allah and of the Prophet. Table 12 gives some examples that illustrate variations in this meaning.

Example 1: extraction from The Book of the Merits of the Companions
"عَنْ عَائِشَةَ، أَنَّ النَّاسَ، كَانُوا يَتَحَرَّوْنَ بِهَدَايَاهُمْ يَوْمَ عَائِشَةَ يَبْتَغُونَ بِذَلِكَ مَرْضَاةَ رَسُولِ اللَّهِ صلى الله عليه وسلم ."
"A'isha reported that people sent their gifts when it was the turn of 'A'isha seeking thereby the pleasure of Allah's Messenger (SAW)".
Example 2: extraction from The Book of Jihad
"قَالَ رَسُولُ اللَّهِ صلى الله عليه وسلم (تَضَمَّنَ اللَّهُ لِمَنْ خَرَجَ فِي سَبِيلِهِ - إِلَى قَوْلِهِ - مَا تَخَلَّفْتُ خِلاَفَ سَرِيَّةٍ تَغْزُو فِي سَبِيلِ اللَّهِ تَعَالَى) ."
"Allah takes care of one who goes out in the way of Allah -till the words-I would not lag behind any expedition which is undertaken to fight in the way of Allah, the Exalted."
Example 3: extraction from The Book of the Prohibited actions
"قام النبي صلى الله عليه وسلم يصلي فقال: أين مالك بن الدخشم؟ فقال رجل: ذلك منافق لا يحب الله ورسوله، فقال النبي صلى الله عليه وسلم: لا تقل ذلك ألا تراه قد قال: لا إله إلا الله يريد بذلك وجه الله! وإن الله قد حرم على النار من قال لا إله إلا الله يبتغي بذلك وجه الله ."

"When the Prophet (SAW) stood up to offer As-Salat (the prayer) he asked, (Where is Malik bin Ad-Dukhshum?) A man replied: (He is a hypocrite. He does not love Allah and His Messenger.) The Prophet (SAW) said, (Do not say so. Do not you know that he said: La ilaha illallah (there is no true god except Allah),' seeking His Pleasure. Allah has made the fire of Hell unlawful for those who affirms that none has the right to be worshipped but Allah.)"

Table 12: Examples of **مَرْضَاةِ** mardat، **سَبِيلِ** sabīl and **وَجْهِ** wağh synonyms in different hadiths.

Al-Hadith WordNet has been utilized by Machine Learning-based Word Sense Disambiguation (WSD) to disambiguating a word that has multiple meanings, because it is crucial to distinguish among these different senses (Diab and Hall, n.d.) .Table 13 shows examples[6] of the word أقام Aqam which has two different senses.

Example 1: extraction from The Book of Marriage
"قَالَ النَّبِيُّ صلى الله عليه وسلم وَلَكِنْ قَالَ السُّنَّةُ إِذَا تَزَوَّجَ الْبِكْرَ أَقَامَ عِنْدَهَا سَبْعًا، وَإِذَا تَزَوَّجَ الثَّيِّبَ أَقَامَ عِنْدَهَا ثَلاَثًا."
"The tradition, (of the Prophet) is that if a married man someone marries a virgin and he has already a matron wife (with him), then he should stay with the virgin for seven days; and if someone marries a matron (and he has already a virgin wife with him) then he should stay with her for three days"
Example 2: extraction from The Book of Prayer
"قَالَ عَبْدُ اللَّهِ إِنَّ الْمُشْرِكِينَ شَغَلُوا النَّبِيَّ صلى الله عليه وسلم عَنْ أَرْبَعِ صَلَوَاتٍ يَوْمَ الْخَنْدَقِ فَأَمَرَ بِلاَلاً فَأَذَّنَ ثُمَّ أَقَامَ فَصَلَّى الظُّهْرَ ثُمَّ أَقَامَ فَصَلَّى الْعَصْرَ ثُمَّ أَقَامَ فَصَلَّى الْمَغْرِبَ ثُمَّ أَقَامَ فَصَلَّى الْعِشَاءَ ."
"Abdullah said: 'The idolators kept the Prophet (SAW) from (offering) four prayers on the day of Al-Khandaq, so he commanded Bilal to call the Adhan, then he said the Iqamah and prayed Zuhr, then he said the Iqamah and prayed 'Asr, then he said the Iqamah and prayed the Maghrib, then he said the Iqamah and prayed 'Isha'."

Table 13: Examples of different senses of the word **اقام**

Hence, we can get a well understanding of the meanings of Al-Hadith Al-Shareef by developing an Al-Hadith WordNet model and deriving a computational linguistic theory for Arabic that combines the new technologies of Natural Language Processing (NLP), the strength of traditional Arabic linguistic theory, and the powerful classical Arabic dictionaries such as the "المعجم الوسيط" Al Waseet Dictionary and the "المعجم المحيط" AL Moheet dictionary.

[6] Al-Hadith in these examples has been taken from the Sunah website: https://sunnah.com/

Hence, we can get a well understanding of the meanings of Al-Hadith Al-Shareef by developing an Al-Hadith WordNet model and deriving a computational linguistic theory for Arabic that combines the new technologies of Natural Language Processing (NLP), the strength of traditional Arabic linguistic

For Al-Hadith Wordnet, we have considered two different possible techniques. Moreover, we created lists of proposed Arabic translations for the different words enclosed in the English synsets consistent with the set of Base Concepts. These Base Concepts are the major building blocks on which the other word meanings in the wordnets depend. In this case the input to the lexicographical task is the English synset, its set of synonyms and their Arabic translations. Moreover, we derived new Arabic word forms from Arabic verbal synsets, that is already exist and manually built, using inflectional and derivational rules and produced a list of English synset related to each form. In this case the input is the Arabic verb, that is the set of all possible derives and English synsets. This would be connected to a corresponding Arabic synset. In both cases, the list of proposals was confirmed by lexicographers in manual process.

Our major design goal is to make searching text possible using either Arabic or English words. In Arabic part, the search can be done using an input that is either an Arabic word or its transliteration, with or without diacritics, a conjugated verb or root form. In the case of the absence of diacritics the search result displays all of the different analyses, considering all possible diacritical marks of the word. In English mode, the search supports a word sense which allows a user to navigate through hyponym and hypernym relations between synsets. A grouping of word sense search and tree navigation allows a user to efficiently, and quickly browse translations for English into Arabic, a very important feature for those who speak English but are unfamiliar with Arabic and wish to understand Islamic instructions.

6.2 Related Work

WordNet has proven to be an effective resource for both Information Retrieval, in particular, and NLP, in general(Al-zoghby and Shaalan 2015).

The first WordNet was developed for the English language, the so-called Princeton WordNet (PWN). The WordNet for Modern Standard Arabic (MSA) was built many years later. The first Arabic WordNet (Black et al. 2006) and (Elkateb et al. 2002) was released in 2007. It followed the development

procedure of the English PWN (Miller et al. 1990) and that of Euro WordNet3 (Ellman 2003). The first Arabic WordNet developed in two stages: first building a core wordnet around the most important concepts, the so-called Base Concepts, and then extending the core wordnet downward to more specific concepts using additional criteria. The base concepts were defined from a European perspective and are, therefore, limited and most likely subject to biases when other languages are considered. Accordingly, words and concepts that are frequent and important in other languages, such as Arabicare likely to be overlooked when WordNets are built using the Base Concepts, even when the core is extended with hyponyms and meronyms.

The most recent version of Arabic WorldNet (AWN) was produced during the Arabic WordNet Project (Rodr et al., 2008), within the Global WordNet Association and according to the methodology of Euro WordNet (Ellman 2003). The project used 12114 distinct words and 9576 synsets, which combine themselves into 17419 senses. The nouns and verbs cover 14665 and 2454 senses, respectively.

To the best of our knowledge, the studies done on Arabic WordNet (Elkateb et al. 2002) have not included the Classical Arabic of "the Holy Qur'an and Al-Hadith Al-Shareef (Almaayah, Sawalha, and Abushariah, 2015), nor have they proposed a Model of Quranic Arabic WordNet. As such, there are no systems that we can evaluate or compare with our proposal.

6.3 WordNet Challenges

Arabic is a Semitic language which differs from European and English languages syntactically, morphologically and semantically. The term "classical Arabic" refers to the standard form of the language used in the Holy Quran and in the Al-Hadith Al-Shareef. The writing system of the Arabic language differs from the other languages alphabets. Arabic script is written from right to left, and in different shapes for the same letter based to its position in a word; For example, the four different shapes for one letter (عـ,ـعـ, ع ، ـع) one at the beginning, one in the middle, and two at the end of a word.

Arabic words are only vocalized in some cases while they remain unvocalized in other cases.Many types of diacritic signs are used in written Arabic script: فتحة Fatha (َ), كسرة Kasra (ِ), ضمة Damma (ُ), سكون Sukun (ْ), شدة Shadda (ّ) , مد Mad (آ) and تنوين Tanwin (ً ٍ ٌ). These are the short vowels that are used to display the correct pronunciation and meaning of words. In the text of Classical Arabic, in

particular in Al-Hadith Al Shareef and in the Holy Quran, vowel diacritics appear in full (every letter). However, it is very rare to find diacritic signs in Modern Standard Arabic, except in cases where words might be ambiguous or difficult to read. For instance, consider the word (سلم) that consists of the three letters ‘s’ , ‘l’, and ‘m’. This word is very ambiguous without including vowel diacritics, as shown in Table 15. It is obvious that diacritics are easier to read and, hence, used to resolve ambiguity, but harder to be written properly, even for native speakers, because the process requires a deep knowledge of the Arabic language. This can cause a serious problem for writing input for information retrieval systems and in the development of computerized lexical resources as they depend on well- formed user input and may even result in users avoiding the use of such systems. The only way to disambiguate the diacritic-less Arabic words is to write them within the context.

Word	Transliteration	POS	Arabic Meaning	Translation
سَلَّمَ	Salam	Verb	{قبل ،رضي ،صدّق } { قول السلام عليكم على الناس او عند الانتهاء من الصلاة }	Accepted Say salaam
سَلِمَ	Salim	Verb	{شفي وبرئ من المرض }	Saved
سُلِمَ	Solim	Verb	{ اوصل، أعطى، نقل }	Transmit
سِلْم	Silm	Noun	{الصلح والاسلام }	Peace and safety
سُلَّمْ	Solam	Noun	{مايصعد عليه ، درج }	Ladder

Table 14: Vowel diacritics

Another point about Arabic to study is that the Arabic language has neither capital letters (i.e., for proper names: the names of people, countries, cities, etc.) nor acronyms. This caused increased ambiguity and complicates tasks in Information Extraction generally and in Named Entity Recognition particularly (Shaalan 2014a).

6.4 Al-Hadith Al- Shareef

Al-Hadith Al-Shareef is one of the two fundamentals bases of Islam besides the Holy Qur’an (“Building Hadith Ontology to Support the Authenticity of Isnad Building Hadith Ontology to Support” 2016). Together they form the very pillar of Islam: its faith conviction, jurisprudence, knowledge, wisdom and the future. Al-Hadith is one of various reports describing the words, actions, or habits of the Islamic prophet Muhammad, Peace Be Upon Him (PBUH).

The two main branches of Al-Hadith are called the "Isnad" and the "Matn". Together they form the basic components (segments) of each Hadith. The "Isnad" is concerned with the explanation of the individual traditions whose process can be traced back to the prophet's. The "Matn" essences on the actual validated traditions, and are considered as a source of religious authority. From the day Islam was formalized, this source of authority has been regarded as a second resource after the Qur'an (Alkhatib 2010)

Al-Hadith scholars agree that research in Isnad is important for the science of Al-Hadith (Abdul Karim and Hazmi 2005). In order to know whether or not an Al-Hadith is authentic, Al-Hadith scholars follow clear steps in judging the Isnad. These steps are considered to be strict traditional methods.

Software tools like electronic Al-Hadith encyclopedias and some Hadith websites have been used to help in judging particular Isnads. Recent tools, such as ontologies related to the semantic web can also be used to help in the process of judging Isnads. Ontology is a formal explicit description of concepts in a domain (classes); the properties of each concept describing various features and attributes of that concept (slots or properties). It is a semantic web building block that can be used in many applications, such as Information Retrieval systems and Decision-Support Systems. Ontology together with a set of individual instances of classes constitutes a knowledge base (Jaafar and Pa 2017).

On the other hand, many researchers around the world, such as Natural Language Processing scholars and certain research groups, focus on the Matn in their studies. For example, in (Harrag, El-Qawasmah, and Al-Salman 2011) the authors provide an evaluation of several stemming methods for Hadith text categorization, whereas (Baraka and Dalloul 2014) classified Al-Hadith in many chapters using different methods. In (Harrag, El-Qawasmah, and Al-Salman 2011), a text mining tool was developed to search a query from an Al-Hadith dataset, providing a list of relevant Hadiths sorted by the extent of their similarity.

One of the advantages of applying natural language processing tasks to an Islamic text would be the implementation of intelligent systems which can answer virtually any question with data from the Qur'an and the Hadiths, thereby helping Muslim and non-Muslim societies to learn about and appreciate the Quran and the Hadiths. However, another important Islamic source, i.e., the Al-Hadith, has been overlooked by most academics in the computer science field.

All the Al-Hadith texts in this chapter have been taken from Sahih Bukhary and Sahih Muslim, which are the recognized collection of the authentic assortment of the Sunnah or the Prophet's saying (Hadith) over the time. Sahih Bukhary contains roughly 7500 hadiths classified in 97 books, and Sahih Muslim contains 7500 hadiths classified in 57 books (Faidi et al. 2015).

6.5 Al-Hadith WordNet Construction

In this section, we describe the steps that we followed in building the semantic relations of Al-Hadith WordNet from Al-Hadith text.

a) Al-Hadith preprocessing by

- Removing the Isnad component from each Hadith in order to focus on analyzing the Matn;
- Elimination of the stop words;
- Word Tokenization;
- Word stemming; and
- Parts-of-Speech (POS) tagging for each word in the Al-Hadith text.

b) Formation of Synonym sets (synsets by grouping words of similar meaning together and assigning their POS category. For example, the set of words {أعطى Aa'ta , ساهم Saham , تبرع Tabar'a } that share the sense "give" are grouped together in a synset that has its POS as a verb. The basic criteria for selecting synonym sets to be covered in an Al-Hadith AlShareef WordNet are listed below, and illustrated in Figure 13.

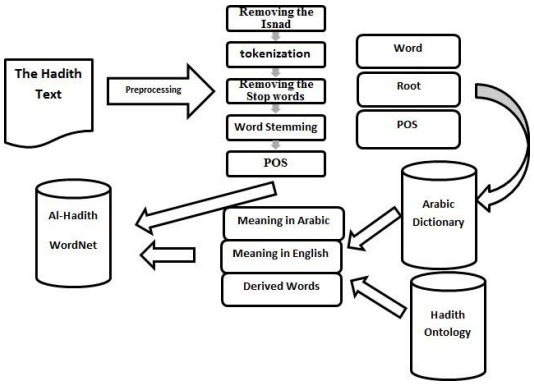


Figure 13: A Framework for Al-Hadith Al-Shareef WordNet Construction

a) Building Semantic Relations between Synsets: the semantic relationships for the Al-Hadith WordNet are wide-ranging. In this study, we decided to focus only on the following significant relationships:

i. **Synonymy:** Links to words that have similar meanings. For example, the words {حول *Hawl*, سنة *Sanah*, عام and A *ʿām*} are synonyms, and they mean "a year".

ii. **Antonyms:** Links to words that give opposite meanings, such as الدنياldubnai "life", and الآخرة Al a'kherah "the afterlife", and النور Alnoor "light" and الظلماتAltholomat "darks" are labeled 'antonyms'.

iii. **Citation of Books:** Links each word to the books that mention them. These books have gone through rigorous classification and labeling processes.

c) The Arabic WordNet lexicon provides a good semantic structure for computing the semantic similarity between words. The semantic structure of Arabic WordNet can be presented as a tree graph. The nodes of the tree graph are synsets and the edges of the tree graph are semantic relations. Our research measured the semantic relations by calculating the frequency score. This method functions well because, as the frequency of the lexicon becomes bigger, the weight of the concepts increases and gives a more accurate classification.

The Al-Hadith WordNet was developed using:

- The proper text of the Al-Hadith Al-Shareef corpus;
- A word stemmer (Boudchiche et al. 2017a) for exploring roots and POS tags;
- An Al-Hadith ontology for providing the English translation (meaning) of each root in the corpus (Baraka and Dalloul 2014).
- The Arabic meaning and the derived words from each root found in classical Arabic dictionaries; and
- The connected Arabic and English words and their meanings along with their semantic relations.

6.6 Al-Hadith Al-Shareef WordNet Evaluation

Al-Hadith WordNet was evaluated using a text classifier that we developed for this purpose. It was applied on around 8500 synsets which include: 6126 nominal, 1990 verbal, 310 adjectival, and 71 adverbial expressions.

The classifier can classify an Al-Hadith text according to the Al-Hadith reference book(s) it belongs to. The text classification task is comprised of three main components: pre-processing the data , building

classifier and document classifications(Baraka and Dalloul 2014). For the data pre-processing, we used the recent version of AlKhalil Morphological Analyzer, i.e. Sys2, to find the word stem and its POS. AlKhalil Morpho Sys2 is a morpho-syntactic analyzer of words taken out of context; either partially or totally vowelized (Boudchiche et al. 2017b) . The IF-IDF and Polynomial Networks (PNs) algorithms were used to compute the text classification using Al-Hadith WordNet, as illustrated in Figure 14. Alkhalil Morpho Sys2 generates the stem and POS tag for each word in the text after removing all stop words. The classification accuracy would be more precise if the data fits well with the model (Do and Poulet 2006).

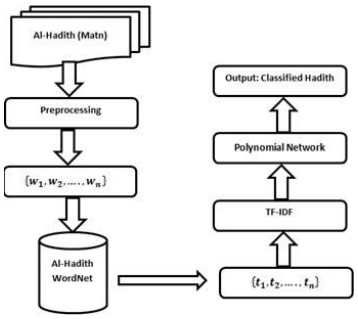


Figure 14: Evaluation system for Al-Hadith WordNet

6.6.1 Polynomial Networks (PNs)

Polynomial neural Network (PN) classifiers have been known in the literature for many years , and have proven to be competitive with the highest performers in the field of English Language (Al-tahrawi and Al-khatib 2015). In this research, we use the Polynomial Neural Networks algorithm to classify Al-Hadith Al–Shareef documents. Details of the algorithm and its equation in Text Classification are clarified in the following subsections.

6.6.1.1 *Polynomial Networks Architecture*

The representation of the PN model adopted in this research consists of two layers (Al-tahrawi and Al-khatib 2015; Campbell, Assaleh, and Broun 2001)

The first layer (the input layer) is a set of inputs (features) $x(x_1, x_2, \ldots, x_N)$ where N is the number of input features, which are used to form a set of monomial basis functions $p(x)$:

$$\prod_{j=1}^{N} x_j^{k_j} \ , where\ k_j \geq 0\ , and\ 0 \leq \sum_{j=1}^{N} k_j \leq K \quad (11)$$

The second layer of the PN model is the average of all feature scores $w^t\ p(x)$, that combined from all the outputs of the basic functions, where w is the verification model:

$$s_j = \frac{1}{M} \sum_{i=1}^{M} w^t\ (x_i) \qquad (12)$$

6.6.1.2 Polynomial Network classifiers' training phase

Polynomial Networks are trained to approximate an highest output using mean squared error as the objective criterion. The polynomial expansion of the ith class term vectors (documents) is indicated by (Al-Tahrawi and Al-Khatib 2015; Campbell, Assaleh, and Broun 2001):

$$M_i = [p(x_{i,1})p(x_{i,2})p(x_{i,3}) \dots p(x_{i,N_i})]^t \qquad (13)$$

where N_i is the number of training feature vectors for class i, and $p(x_i, m)$ is the basis function of the mth feature vector for class i. The matrix M is generated, whose rows are the polynomial expansion of M_i classes data.

$$M = [M_1 M_2 M_3 \dots M_{nclasses}]^t \qquad (14)$$

where $nclasses$ is the number of training classes. The training problem formed as:

$$w_i^{opt} = arg_w min \|Mw - O_i\|_2 \qquad (15)$$

where O_i is the column vector consisting of N_i ones in the rows where the ith class are located, and 0s otherwise. A class model w_i^{opt} can be achieved by applying the method of normal equations (13) to (15) gives the following problem

$$M^t M w_i^{opt} = M^t O_i \qquad (16)$$

Finally, w_i^{opt} is calculated as follow:

$$w_i^{opt} = (M^t M)^{-1} M^t O_i \qquad (17)$$

6.6.1.3 Polynomial Network classification phase*:*

Classification of a new unclassified input involves two steps: identification that involves finding the top corresponding class of a new input, and verification that can either accept or reject the clam mode of

identification step. In our experiments, we accepted classifications with scores above 0.47. The new unclassified input is then allocated to the class c , as following (Al-Tahrawi and Al-Khatib 2015; Campbell, Assaleh, and Broun 2001)

$$c = arg_i \, max w_{ii}^{opt} . P(x) \;\; for \; i = 1,2,3, \dots, nclasses \qquad (18)$$

6.6.1.4 Text Classification (TC) using PNs

The training phase of TC starts by generating a term vector x for each training document, using the vector space model. Terms are usually represented by their tf-idf weights, as in our experiments. The desired-order PN basis function is then generated for each training document in the corpus, as in Equation (11). PNs of degree 2 were used in our experiment. After generating the basis function for each input document, the polynomial expansion of class i is formed as given in Equation (13). Next, the matrix for all classes M is calculated as presented in Equation (14). The PN is now trained to approximate the best output using the mean-squared error criterion as in Equation (15), and each for class the weights are calculated as in Equations (16) and (17). Finally, examine the classifier of the basic function to the nearest class as in Equation (18).

6.6.1.5 Feature Selection (FS)

We used TF-IDF as an FS metric for selecting the best discriminating features in the dataset. The TF-IDF is a calculation formed by multiplying the term frequency (TF) by the inverse document frequency $log_2(^n/_{n_j})$, explained by the equation:

$$TF - IDF = TF * log_2 \left({}^n/_{n_j} \right) \qquad (19)$$

The logarithm of the IDF is used instead of the IDF weight as a way to reduce the amount of the weight for the words compared to the exact number of words in a document.

6.7 Experiments and results

We tested Al-Hadith WordNet by entering around 2671 Hadiths from 24 books to evaluate the WordNet database, structure and relations. Table 16 illustrates the evaluation results for classifying Al-Hadith Al-Shareef. Description of Performance Evaluation Measures and Results of experiments shown in this

research are explained in Sections 6.1 and 6.2.

Books	Doc	Books	Doc
Call to Prayers (Adhaan) كتاب الأذان	270	Representation, Authorization, Business by Proxy كتاب الوكالة	18
Belief كتاب الإيمان	200	Oppressions كتاب المظالم	40
Knowledge كتاب العلم	12	Wills and Testaments (Wasaayaa) كتاب الوصايا	40
Ablutions (Wudu') كتاب الوضوء	102	Fighting for the Cause of Allah (Jihaad) كتاب الجهاد والسير	95
Rubbing hands and feet with dust (Tayammum) التيمم	12	One-fifth of Booty to the Cause of Allah (Khumus) كتاب فرض الخمس	60
Prayers (Salat) كتاب الصلاة	550	Beginning of Creation كتاب بدء الخلق	125
Funerals (Al-Janaa'iz) كتاب الجنائز	60	Wedlock, Marriage (Nikaah) كتاب النكاح	179
Obligatory Charity Tax (Zakat) كتاب الزكاة	95	Divorce كتاب الطلاق	195
Hajj (Pilgrimage) كتاب الحج	250	Laws of Inheritance (Al-Faraa'id) كتاب الفرائض	45
Fast كتاب الصوم	110	Blood Money (Ad-Diyat) كتاب الديات	45
Kafalah كتاب الكفالة	8	(Statements made under) Coercion كتاب الإكراه	12
Asking Permission كتاب الاستئذان	68	Afflictions and the End of the World كتاب الفتن	80

Table 15: List of books in AL-Hadith Al-Shareef

6.7.1 Analysis of the Results

The results of PNs classification algorithm on an Al-Hadith dataset disclose that PNs have a very good performance accuracy, by taking 1% for each class as a feature reduction. The average for Precision is 95.4%, the average for Recall is 93.5%, and the average for F-measure is 94.5%, as illustrated in table 17. All of the keywords in the Al-Hadith texts were generated in WordNet, which offered their synonym meanings in Arabic, their semantic relation(s), their associated book(s), and their translation into English.

Most of the books share the same synsets, and so the evaluation could not give 100% accuracy. For instance, the Prayer book and the Call to Prayers (Adhaan) share most of their words.

Books	Precision	Recall	F1-measure
Call to Prayers (Adhaan) كتاب الأذان	97.5	93.8	95.61
Belief كتاب الإيمان	90.3	89.9	90.10
Knowledge كتاب العلم	96.6	94.7	95.64

Ablutions (Wudu') كتاب الوضوء	98.7	98.1	98.40
Rubbing hands and feet with dust (Tayammum) التيمم	98.2	97.6	97.90
Prayers (Salat) كتاب الصلاة	97.5	95.7	96.59
Funerals (Al-Janaa'iz) كتاب الجنائز	97.4	95.5	96.44
Obligatory Charity Tax (Zakat) كتاب الزكاة	98.6	96.8	97.69
Hajj (Pilgrimage) كتاب الحج	97.4	96.1	96.75
Fast كتاب الصوم	97.7	96.3	96.99
Kafalah كتاب الكفالة	94.2	92.1	93.14
Asking Permission كتاب الاستئذان	93.8	90.4	92.07
Representation, Authorization, Business by Proxy كتاب الوكالة	89.6	88.7	89.15
Oppressions كتاب المظالم	89.5	86.2	87.82
Wills and Testaments (Wasaayaa) كتاب الوصايا	92.3	90.7	91.49
Fighting for the Cause of Allah (Jihaad) كتاب الجهاد والسير	97.5	96.44	96.97
One-fifth of Booty to the Cause of Allah (Khumus) كتاب فرض الخمس	94.4	92.6	93.49
Beginning of Creation كتاب بدء الخلق	93.6	91.1	92.33
Wedlock, Marriage (Nikaah) كتاب النكاح	97.1	94.6	95.83
Divorce كتاب الطلاق	96.3	94.6	95.44
Laws of Inheritance (Al-Faraa'id) كتاب الفرائض	91.5	90.2	90.85
Blood Money (Ad-Diyat) كتاب الديات	96.5	94.3	95.39
(Statements made under) Coercion كتاب الإكراه	95.8	93.6	94.69
Afflictions and the End of the World كتاب الفتن	97.2	95.1	96.14
Average	95.4	93.5	94.5

Table 16: The evaluation Metrics of Al-Hadith al-Shareef

6.8 Chapter Summary

We proposed the use of WordNet synsets in a syntax-based reordering model to be used in neural machine translation (NMT) to enable the model to generalize to phrases not seen in the training data but that have equivalent meaning in Arabic and English using paraphrasing technique (see chapter 8). A novel approach was used to develop an Al-Hadith WordNet by building semantic connections between words in order to provide a better understanding of the meanings of Al-Hadith words, using traditional Arabic dictionaries and Al-Hadith ontology. This Al-Hadith WordNet gives the similarity meaning

(synonym) for each synset, the semantic relation(s) between synsets, and the book(s) that each word belongs to. Al-Hadith WordNet can also be used for evaluating Modern Standard Arabic words. Al-Hadith WordNet is an interesting language resource, as it allows the user to discover the relationship of words to each other; it is also valuable in a number of language processing tasks demanding an understanding of the meaning of language, such as Information Retrieval, Word Sense Disambiguation, Automatic Text Classification, Automatic Text Summarization, Question Answering, and Machine Translation. Most of the Arabic WordNets have been defined from a European perspective and are therefore limited and most likely subject to biases, while in al-Hadith WordNet words having similar senses are grouped together and the groups are interconnected through some lexical and semantic relations.

Chapter Seven: Machine translation for Arabic Metaphor using Neural Paraphrasing

7.1 Introduction

While everyone may be familiar with the concept of paraphrase in its most fundamental sense, there is still a room for elaboration on how paraphrases may be automatically generated or extracted for use in language processing applications. The rest of this section formalizes the notion of a paraphrase and scopes out the collusion of the Arabic language.

Our goal in this chapter is, therefore, to introduce the first result on automatic Arabic translation using paraphrases with neural machine translation employing a bilingual corpus and WordNet (Alkhatib, Monem, and Shaalan 2017d). We also aim to solve the problem of metaphors in both directions (i.e. Ar→En and En→Ar). The experiments reveal that our paraphrase neural machine translation system shows superior performance compared to the standard phrase-based system. We used METEOR (Isabelle, Cherry, and Foster 2017), which measures translation quality *averaged* over all the sentences in a corpus.

7.1.1 What is a Paraphrase?

The principle of semantic equivalence most generally defines the concept of paraphrasing: A paraphrase is an alternative surface form in the same language describing the same semantic content as the original form. The idea of paraphrasing has been examined in conjunction with and employed in many natural language processing applications. Given the difficulty inherent in analyzing such a complex task, an unfortunate but necessary improvement is to impose specific limits on the scope of our discussion (K. Shaalan, Hendam, and Rafea 2012) .

Individual lexical items sharing the same meaning are usually referred to as lexical paraphrases, or, more commonly, synonyms, for example, (حار, har, hot) versus (دافئ, dafe', warm), and (تناول, tnAwl, eat) versus (استهلك, Asthlk, consume). However, lexical paraphrasing cannot be limited strictly to the notion of synonymy. There are several other methods such as hyperymy, where one of the words in the paraphrastic relationship is either more general or more particular than the other, for example, (رد, rd, reply) and (قول, qwl, say)qwl, say) (K. Shaalan, Hendam, and Rafea 2012).

7.1.2 Linguistic Description

In Arabic, machine translation (MT) is bound to face many problems in producing exact coherent translations between Arabic and English. When evaluating the output of MT, the transferred meaning is the most significant focus point. Semantics is a critical aspect of translation both as a theory and in its application; it therefore, requires our utmost attention. Very few systems have addressed the problem of Arabic syntactic generation within MT and in Interlingua-based multilingual translation in particular, due both to the language complexity and to a lack of resources (Abdel Monem et al. 2008).

Arabic metaphor	Transliteration	Translation
تحديات تنال من هوية الأمة واستقرار مجتمعاتها على نحو يستهدف الروابط بين دولها وشعوبها و تفكيك نسيج مجتمعاتها	tHdyAt tnal mn hwyt Alomt wa estqrar mjtmeatha mjtm'eatha 'ela nhw ysthdf alrwabt byn dwlha wsh'ewbha wa tfkyk nsyj mjtm'eatha	The challenges that compromise the identity of the nation disrupting the stability of their societies and specifically target bonds between their states and peoples, causing disintegration of societies relations.
نريد حاليا صناعة التاريخ	Nryd halya snaet AltArykh	We want now to make the history

Table 17: Examples of different metaphors

The use of metaphor is ubiquitous in natural language text, and it is a severe bottleneck in automatic text understanding. Improving methods to identify and deal with metaphors is an open problem in Arabic natural language processing, especially in its Machine Translation. The complexities involved in any metaphor can semantically modify the meaning of the machine-translated text. This makes metaphors a vital research area for computational and cognitive linguistics; their automatic identification and interpretation is indispensable for any semantics-oriented Arabic natural language processing (Alkhatib and Shaalan 2018). Table 18 shows different examples of metaphors.

7.2 Related Work

In the literature, previous work on using neural networks for Arabic translation has mainly focused on using neural networks to induce an additional feature for phrase-based statistical machine translation systems (see, e.g., (Devlin et al. 2014); (Setiawan et al. 2015)). The paraphrase database project (PPDB) has paraphrase resources for multiple languages, including Arabic. The paraphrases are achieved using parallel bilingual corpora by implementing the pivot method, where one language is used as a bridge or for common meaning representation (Bannard and Callison-Burch 2005).

Turker-assisted paraphrasing is used to improve English-Arabic MT (Denkowski, Al-Haj, and Lavie 2010). A comparison between several paraphrase acquisition techniques on sentential paraphrasing is given in (Hutchison et al. 2010), but it does not carry experiments on Arabic sentential paraphrasing. To the best of our knowledge, there is no study that has solved for or adequately covered the metaphor and word sense ambiguity of Arabic language.

7.3 Neural Paraphrasing

In this section, we present our Arabic paraphrasing approach, which is based on NMT. It uses neural machine translation to first paraphrase the Arabic metaphor to a pivot the language (Modern Standard Arabic) with the same meaning, and then translates it to English. In the following, we shortly overview the basic encoder-decoder NMT framework and then explain how it can be extended to paraphrasing.

7.3.1 NMT Background

NMT has shown promising results lately (Sutskever, Vinyals, and Le 2014; Bahdanau, Cho, and Bengio 2014; M.-T. Luong, Pham, and Manning 2015b). Most NMT methods follow the encoder-decoder framework proposed by (Cho et al. 2014b), which, as its name indicates, typically consists of two RNNs. The encoder is a recurrent neural network (RNN) that reads the source sentence and compresses the meaning into a sequence of vector representations. Next, the decoder RNN takes the vector representation and generates the target sentence word by word. The decoder stops once a particular symbol denoting the end of the sentence is generated. figure shows the Archetecture of Encoder-Decoder model as illustrated in figure 15.

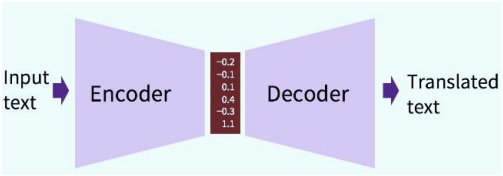


Figure 15: Encoder- Decoder Archetecture

For a language pair, an encoder takes in a source sentence $X = \{x_1, \ldots, x_{T_x}\}$ as a sequence of linguistic symbols and generates a sequence of context vectors$V = \{h_1, \ldots, h_{T_x}\}$. Our Arabic Neural Paraphrasing method uses a bidirectional RNN, where each context vector h_t is the sequence of the forward and the backward RNN's hidden states at time *t*. The decoder is a conditional RNN language model that, given a source sentence, generates a probability distribution over the translation. The decoder's hidden state is updated at each time t':

$$D_{t'} = RNN(D_{t'-1}, y_{t'}, V_{t'}) \ldots\ldots (20)$$

The update uses the previously hidden statez_{t_0-1}, the previous target symbol $y_{t'-1}$ and the time-dependent context$V_{t'}$, calculated by an attention mechanism $\alpha_{t,t'}$ over the source sentences' context vectors:

$$V_{t'} = \sum_{t=1}^{T_X} \alpha_{t't} h_t \ldots\ldots (21)$$

The probability of the target sentence$Y = \{y_1, \ldots, y\}$ is the product of the probabilities of the symbols within this sentence:

$$P(Y|X) = \prod_{t'=1}^{T_Y} p(y_{t'}|y < t', X) \ldots\ldots (22)$$

Please refer to (Bahdanau, Cho, and Bengio 2014;Sutskever, Vinyals, and Le 2014; Cho et al. 2014) for more details.

7.4 Arabic Neural Paraphrasing

Our approach to Arabic paraphrasing is the pivot method which is inspired by (Bannard and Callison-Burch 2005). Pivoting is often used in MT to overcome the deficiency of parallel data, i.e., when there is no direct translation path found from the source language to the target. Instead, pivoting takes advantage of indirect paths by an intermediate language.

The concept dates back at least to 1997 when Kay observed that ambiguities in the translation from one

language into another may be resolved if a translation through a third language is possible (Nirenburg, Somers, and Wilks 2003). This approach has met with success in traditional phrase-based SMT (Wu and Wang 2007); (Utiyama and Isahara 2007), and more recently in NMT systems (Firat et al. 2016). In our case of paraphrasing, pivoting offers a path from the ambiguities of Arabic to English, through a translation to simple Arabic forms. In other words, we translate a source sentence into a pivot language and then translate the pivoted phrase into the target language. Pivoting using NMT ensures that the entire sentence is considered when choosing a pivot. Contextual information is thus considere when translating, which allows for a more accurate pivoted sentence. This approach also places greater emphasis on capturing the full meaning of the sentence, a crucial aspect of paraphrasing.

To extract paraphrases, we first obtain a parallel corpus through Arabic metaphor and English. We prune the corpus to only those containing sentences with less than 60 words each. We tokenize words of those sentences using the Stanford NLP Arabic Tokenizer (Manning et al. 2014). Then, we perform sentence alignment in order to calculate the conditional probabilities for our paraphrase equation. Consequently, we run the corpora on GIZA++ (Och, F. J., Tillmann, C., and Ney, H. 1999), the alignment tool most widely-used with MT involving Arabic (Al-Raisi, Bourai, and Lin 2018). Once we have a database of paraphrase mappings, we can then replace phrases with their corresponding paraphrases by selecting the phrases with the highest probability.

A crude approach to pivoting is one-to-one translation. The ambiguous source sentence *A1* is translated into English phrase E, which has a similar or exact metaphorical meaning. The English E is then translated back into Arabic, producing the intermediate pivot phrase, which gives a probability distribution over English sentences, *E*. This substitution approach was used by (Bannard and Callison-Burch 2005). Our approach to obtain a paraphrase is summarized in the following mathematical equation:

$$P(E|A1, A2): P(E|A1, A2) = P(E|A2) \qquad (23)$$

In encoder-decoder models, care is taken during each decoding step to indicate which words are the relevant source words. In our case, each word of the paraphrase relates to words in the pivot sentence, and each word in the pivot sentence relates to words in the source sentence.'

Original احمر وجه الوزير حين ظهر علي الشاشة وطعن في كلامه
Translation The minister's face blushed when he appeared on the screen and when doubted his words
Paraphrasing احمر وجه الوزير خجلاً شديداً حين ظهر علي الشاشة وشكك في كلامه

Table 18: An example for Arabic Neural Paraphrasing Approach

An example of this approach is given in Table 19, where close attention has successfully identified the semantically equivalent parts of two sentences. Beyond providing interpretable paraphrasing, attention scores can be used in both generation and classification tasks. Furthermore, our approach can readily be used to perform text generation via an NMT which takes the advantage of semantic processing offered by the WordNet database (Alkhatib, Monem, and Shaalan 2017d). Figure 16 shows and example of alignment process.

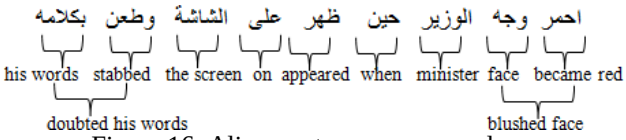


Figure 16: Alignment process example

7.5 Evaluation

Designing the appropriate automated metrics for evaluating machine translations is challenging due to the variety of acceptable translations for each different sentence. Favorite metrics produce scores mainly based on matching the sequences of words in the system translation to those in one or more reference translations. The metrics primarily differ in how they account for reordering and synonyms.

7.5.1 Measuring the Results

METEOR (Cer, Manning, and Jurafsky 2010) is an evaluation measure that calculate a one-to-one alignment between mapping words in a candidate with a reference translation. If a word matches multiple other words, choice is given to the alignment that reorders the words the least, with the amount of

reordering estimated by the number of crossing alignments. Alignments are first created for exact matches between words. Additional alignments are then created by repeatedly running the alignment procedure over unaligned words, first allowing for matches between word stems (Pasha et al. 2014), and then allowing matches between words listed as synonyms in WordNet (Alkhatib, Monem, and Shaalan 2017b).

After realizing the final alignment, METEOR calculates the candidate translation's unigram Precision (P) and Recall (R), $P = \frac{matches}{length\ trans}$, and $R = \frac{matches}{length\ ref}$, respectively. These two values are then gathered into a weighted harmonic mean (5). To penalize reordering's, this value is computed by a fragmentation penalty based on the number of parts the two sentences would need to be broken into to allow them to be reordered with no crossing alignments,

$$P_{\beta,\gamma} = 1 - \gamma(\frac{chunks}{matches})^{\beta} \qquad (24)$$

$$F_{\alpha} = \frac{PR}{\alpha P+(1-\alpha)R} \qquad (25)$$

$$METEOR_{\alpha,\beta,\gamma} = F_{\alpha}.P_{\beta,\gamma} \qquad (26)$$

The free parameters $\alpha, \beta, and\ \gamma$ can be used to tune the metric to display human judgments on a specific language and to adjust to any variation of the evaluation task (e.g., ranking candidate translations vs. reproducing judgments of translations' adequacy and fluency). We used cosine similarity to compute the matches between our proposed system and human translation.

METEOR was implemented to explicitly address the weaknesses in BLEU. It evaluates a translation by computing a score based on explicit word-to-word matches between the translation output and a reference translation. If more than one reference translation is available, the given translation is scored against each reference independently, and the best score is reported (Banerjee and Lavie 2005).

7.5.2 Setup

The training dataset for Arabic Metaphor translation consists of a corpus of bilingual metaphors comprising 90k sentences extracted manually from Arabic rhetoric books. The corpus includes almost all of the Arabic metaphors with their translation into English by a bilingual group. The group consisted of native speakers of Arabic who had lived in the United States for the past several years and who had

worked as annotators.

The test dataset corpus consists of 2000 Arabic metaphor sentences, including the headlines. The corpus covers political, sport and art topics extracted using "HTML Text Extractor"[7] (Allam, Kouta, and Sakre 2015) from the homepage of the Egypt State Information Service (SIS)[8] over 5 weeks in 2018.

The best evaluation metric to use here is the one that ultimately points to the best translations according to human judges. We performed a human evaluation of selected models using the METEOR score to measures the translation quality averaged over all the sentences in the corpus. These evaluations used two encoder-decoder NMT models (one-to-one pairs): Arabic → English and English → Arabic, as illustrated in Table 20, which displays the METEOR scores with our model and with the Google translator. We also calculated the cosine similarity between our model and human judgment, as shown in Table21

We considered the datasets from SIS for the human judgment result, as they were translated manually, i.e. gold standard dataset. The table shows the high correlation between human evaluation and our model, in both directions. We examined the dataset to determine if the results were biased by sampling or data peculiarities. For sentence pairs, including the headline and the following sentence, both human and evaluation metric scores were high, at 89.5% and 91.7%, respectively.

Translation	**Our Model**	**Google translator**
Arabic – English	88.6%	53.2%
English – Arabic	94.9%	57.2%

Table 19: METEOR score evaluations %

Translation	**Cosine Similarity**
Arabic – English	89.5%
English – Arabic	91.7%

Table 20: Similarity match % for our model and human judgment

Table 22 show the result of the example after calculating the Meteor scores:

[7] HTML Text Extractor is a program that extracts only raw text (i.e., without HTML or script format)
[8] The web link for the Egypt State Information Service (SIS) is http://www.sis.gov.eg 9 from April 1st, 2018 to May 5th ,2018

وفيما يتعلق بالضغوط المنتظرة في روسيا قال النجم المصري:

"الضغط كان التأهل للمونديال، نريد حاليا صناعة التاريخ، وتحقيق شيء مختلف".

Best Alternative

نريد	We want (87.34%)	We need(75.4%), we choose (53.1%), we care (34.5%), we intent (31.2%), we will (23.4%)
حاليا	now (82.8%)	Currently (81.7%), recently (66.8%), lately (65.3%), presently(45.7%), Actually (44.9%), at present(33.8%), at the moment(22.8%),
صناعة	to make (95.6%)	Industry (93.7%), manufacture (88.9%), metier (74.2%)
التاريخ	the history (96.7%)	Era (33.7%)

Table 21: Meteor score results

7.6 Chapter Summary

We propose a framework for paraphrasing NMT which solves the ambiguity in Arabic metaphors and introduces an auxiliary score to measure the sufficiency of translation candidates. The advantage of the proposed approach is two-fold. First, it improves metaphor translation, thereby producing better translation candidates. Second, it consistently improves the translation performance of NMT by using paraphrasing combined with the use of the pivoting method. We applied the pivoting method to construct a large coverage paraphrase database for Arabic metaphors that includes over 90K phrase pairs. Experimental results show that the two advantages indeed help our approach to improve translation performance consistently, particularly when compared to the Google translator. Our work offers encouraging results in terms of its correlation with human judgment.

Chapter Eight: Machine translation for Arabic Name Entity Recognition

8.1 Introduction

Named Entity Recognition (NER) task consists of determining and classifying proper names within an open-domain text. This Natural Language Processing task is acknowledged to be more difficult for the Arabic language, as it has such a complex morphology. NER has also been confirmed to help in Natural Language Processing tasks such as Machine Translation, Information Retrieval and Question Answering to obtain a higher performance. NER can also be defined as a task that attempts to determine, extract, and automatically classify proper name entities into predefined classes or types in open-domain text. The importance of named entities is their pervasiveness, which is proven by the high frequency, including occurrence and co-occurrence, of named entities in corpora. Arabic is a language of rich morphology and syntax. The peculiarities and characteristics of the Arabic language pose particular challenges for NER. There has been a growing interest in addressing these challenges to encourage the development of a productive and robust Arabic Named Entity Recognition system (K. Shaalan 2014b) .

The NER task was defined so that it can determine the appropriate names within an open domain text and categorize them as one of the following four classes:

1. Person: person name or family name;
2. Location: name of geographically, and defined location;
3. Organization: corporate, institute, governmental, or other organizational entity; and
4. Miscellaneous: the rest of proper names (vehicles, brand, weapons, etc.).

In the English language the determination of the named entities (NEs) in a text is a quite easy sub-task if we can use capital letters as indicators of where the NEs start and where they end. However, this is only possible when capital letters are also supported in the target language, which is not the case for the Arabic language. The absence of capital letters in the Arabic language is the main difficulty to achieving high performance in NER (Benajiba and Rosso 2007; Benajiba, Diab, and Rosso 2008; K. Shaalan 2014b). To reduce data sparseness in Arabic texts two solutions are possible: (i) Stemming: omitting all of the clitics, prefixes and suffixes that have been added to a lemma to find the needed meaning. This solution is appropriate for tasks such as Information Retrieval and Question Answering because the

prepositions, articles and conjunctions are considered as stop words and are not taken into consideration when deciding whether or not a document is relevant for a query. An implementation of this solution is available in (Darwish 2014);(ii) Word segmentation: separating the different components of a word by a space (blank) character. This solution is more appropriate for NLP tasks that require maintaining the different word morphemes such as Word Sense Disambiguation, Named Entity Recognition, etc.

In Machine Translation (MT), NEs require different translation techniques than the rest of the words of a text. The post-editing step is also more expensive when the errors of an MT system are mainly in the translation of NEs. This situation inspired (Babych and Hartley 2003) to conduct a research study in which he tagged a text with an NER system as a pre-processing step of MT. He found achieved a higher accuracy with this new approach which helps the MT system to switch to a different translation technique when a Named Entity (NE) is detected (Othman 2009).

8.2 The challenges of Arabic Named Entity Recognition

Arabic language is one of the richest natural languages in the world in terms of morphology and inflection. Applying NLP tasks in general and NER task in particular is very challenging when it comes

to Arabic language because of its characteristics. The main characteristics of Arabic language that act as challenges for NER task as follows:

No Capitalization

Capitalization in not a feature of Arabic script unlike several natural languages such as where a NE usually begins with capital letter. Therefore, the usage of this orthographic features is not an option in Arabic NER. However, the English translation of Arabic words may be exploited in this aspect.

The Agglutinative Nature

Arabic is a great inflectional language; a single word has more than one affix. It is expressed as a combination of prefix, lemma, and suffix. Prefixes are articles, prepositions, and conjunctions, while suffixes are objects or personal anaphora. For example (وصنعنا لهم , wSnEnA lhm, and we made to them).

Spelling Variant

Arabic spelling and typographic forms are different from other languages. A word can be spelled differently and still refer to the same meaning, which will create a many-to-one ambiguity. For example, Jeddah‖ can be written as _ جدة ‘ or جده ‘ and the word ―Gram‖ can be written as _ جرام or _ ,غرام‘ both of which have the same meaning.

No Short Vowels

Arabic is a highly inflectional language providing one of the richest and most challenging sets of linguistic and statistical features resulting in long and complex wordforms. It's a relatively free word order language, it can have different meanings (sorts of ambiguities). Thus, NEs can appear in subject and adjective positions making the identification of NEs difficult. For example, ‖ ―اليمن والبركة can mean the pleasing and pond, not The Yemen and pond; this is the lexical ambiguity of the Arabic word.

8.3 Related Work for Arabic Named Entity Recognition

Many related works have been conducted to recognize Arabic name entities. Generally, ANER systems were developed based on any one of the following approaches: Dictionary/Lexicon-based, Heuristics/Rule-based, Machine- Learning based, Deep Learning based, and Hybrid based. In this section, we present the most relevant surveys and comments on some of such works on ANER.

8.3.1 Dictionary Based ANER

A dictionary-based approach is one of the first attempts to solve a named entity problem. It depends on data collection, also known as a dictionary. All possible terms match with a particular name entity. NooJ dictionary is one of the dictionary tools; it recognizes relations of person names, organizations, and between the named entities (Hadni, Ouatik, and Lachkar 2013). In (Al-Arfaj, and Al-Salman 2015)

, a named entity recognition system was developed for the NooJ platform, which is an automatic analysis of text written in Arabic. The NooJ platform is used for descriptive grammars and represents predefined rules based on internal and external evidence. The NER system within NooJ builds on the use of gazetteers. It contains a list of names of persons, personal titles, organizations, dates/time, and currencies, and contains trigger words. A dictionary-based approach detects parts of the text referred from datasets, websites, or text. Ontological concepts are used in various classification approaches for named entity recognition. A simple dictionary-based approach includes three phases: matching, entity

resolution, and filtering (Oudah and Shaalan 2012). A lexicon-driven approach is similar to the dictionary-based approach. These approaches consider an extensive database which stores a massive amount of Arabic name entities. NE lexicon is created by exploiting Arabic WordNet and Arabic Wikipedia (AWN and AWK). Different processes are involved in this construction, including mapping, NE identification, post processing, and discretization. Microblogs (Tweets) presented in ANER for recognizing the named entities by a straightforward and efficient manner. However, it suffers from many challenges including informal language, inconsistent capitalization, and shortened named entities.

8.3.2 Heuristics Based ANER

Heuristics/rule-based approaches make use of heuristic rules to write in either Lexico-syntactic or Lexico-semantic pattern. These approaches are closely related to linguistic preprocessing and morphological analysis to find possible relationships between entities. In (Mesfar 2007), Person Name Entity Recognition System was developed for Arabic language Text using this rule-based approach. This system consists of a lexicon and grammar. The lexicon is in a form of gazetteer name lists, and the grammar is in a form of regular expressions (REs). Both are responsible for recognizing a person named entities. One of the existing rule-based approaches (K. Shaalan and Raza 2007) proposed a NER system

for extracting named entities from crime documents. Different syntactical rules and patterns of Arabic NER are induced and then formalized in this system. Finally, these rules classify named entities in Arabic crime documents. Rules mostly depended upon POS tags, verbs, and a list of names (gazetteer). The positions of IVL (Introductory Verb List), e.g. ―to say, to operate, to walk‖, and IWL (Introductory Word List), e.g. ―location, person name, stop words, company name‖, decide the heuristics to recognize the Arabic named entities (ASHAREF et al., 2012). A set of grammar rules is used to extract

and classify the three types of NEs (person, location, and organization). The order of examining the rules is organization rules, location rules, and finally person rules. Since an organization name may contain a person name or a location name, it is examined first, then followed by location and then person name (Elsebai and Meziane 2011). In this approach, plenty of rules are required to separate the classes and extract named entities. However, the rule-based methods failed to perform on larger datasets.

8.3.3 Machine Learning Based ANER

Machine-learning based approaches learn NE tagging decisions from annotated texts. The most common machine learning techniques are Supervised, Semi-Supervised, and Unsupervised. These techniques solve a NER problem in a classification task and require the availability of large annotated datasets. Some of the supervised techniques utilized for NER are SVM (Support Vector Machine) (Al- Ahmari and Abdullatif Al-Johar 2016) , ME (Maximum Entropy) (K. Shaalan and Oudah 2014), CRF (Conditional Random Fields), HMM (Hidden Markov Model)(Mouhcine, Mustapha, and Zouhir 2018), Genetic Algorithm, Naive Bayesian Classifier (A. Alsayadi and M. ElKorany 2016), and Decision trees (Hamadou, Piton, and Fehri 2010). Artificial Neural Network (ANN), Combined classifiers based on distant learning and semi-supervised methods for ANER were proposed in (Benajiba, Rosso, and BenedíRuiz 2007). In (Freihat, Abed Alhakim et al. 2018) a single Arabic model is introduced based on the concept of Segmentation, POS tagging and Entity Recognition. Maximum Entropy based segmentation and POS tagger has used for ANER. The problem with this machine learning approach is the ―Lack of Accuracy‖ since a single classifier cannot produce sufficient results for training and testing of ANER.

8.3.4 Hybrid ANER

Hybrid approaches are the combination of the rule-based and machine-learning based approaches. The process of this hybrid approach flows from the rule-based system to the machine-learning approach or from machine-learning to rule based approach. In the following literature, we have utilized challenges of a hybrid approach for NER in the Arabic language (Hamadou, Piton, and Fehri 2010; Boujelben, Jamoussi, and Ben Hamadou 2014). Mainly, a machine-learning approach followed by a rule-based approach is used in an attempt to enhance the system performance. Genetic algorithm (ML algorithm) is used to extract and generate the most significant and exciting rules. It is suitable for searching problems, and it suffers from the memory resources and high computation that are required if the size of the individuals of the problem solution is increased. GA selects the best part of the problem to give the best solution using basic operations (initialization selection, crossover, and mutation) and simple logics. However, in such cases probabilistic approaches (fuzzy logic) do not scale well since it deals with imprecision and vagueness, but not uncertainty. After ML, hand-crafted rules are proposed to treat both invalid examples and unseen relations (David O'Steen and David Breeden 2009). Early work on Arabic

NER focused on language-based features. Agglutinative problem is overcome by these features (language generic and language-specific features) (Al-Ahmari and Abdullatif Al-Johar 2016). Hybrid (rule-based and machine-learning based approach) approaches do not always support for the large corpus. We require rules to solve this issue. All the approaches mentioned previously (dictionary/lexicon-based approaches, heuristics/rule-based approaches, machine learning approaches, and hybrid approaches) have been comparatively reviewed in (Althobaiti, Kruschwitz, and Poesio 2015; Ayedh, Tan, and Rajeh 2016). This section also covers hybrid machine-learning algorithms. Hybrid machine-learning algorithms improve the accuracy of individual classifiers. In our current work, hybrid ML classifiers (D-CNN and GA) are proposed for accurate NE, which will overcome the single classifier problem with less accuracy. However, these combination methods are not feasible, and they require further enhancement to improve the system performance (Althobaiti, Kruschwitz, and Poesio 2015). In (A. Alsayadi and M. ElKorany 2016), CRF with lexicon based ANER scheme is bring together for automatic Arabic text classification. Rule-based ANER consume large amount of time, but CRF can

resolve the drawbacks of rule-based approaches. The performance of ANER has improved in terms of accuracy, but it fails to address the issue of Time Consumption. Similarly, in (Abdallah, Shaalan, and Shoaib 2012),CRF with predefined gazetteers are focused for ANER, which saves time of named entity recognition for the types of P, O, and L. In (Tsai et al. 2004), authors have developed a fused ANER system using rule-based approach is combined with machine-learning algorithm. Feature space is designed according to set of features: language specific and independent features. Extracted features are comprised into six set of categories by logically: word-level, morphological, contextual, POS, rule- based component, and gazetteer. Decision tree algorithm is proposed to offer effective results for ANER. Number of rules to be implemented in decision tree is high. When number of entities for recognition will increases, then number of rules will also increase, which make decision tree classifier to be complex and challenging task for any scale of dataset (small and large).

(Benajiba et al.2008) investigated the sensitivity of different NE types to various types of features. They built multiple classifiers for each NE type, adopting SVM and CRF approaches. ACE datasets were used in the evaluation process. According to their findings, it cannot be stated whether CRF is better than SVM or vice versa in Arabic NER. Each NE type is sensitive to different features and each feature plays

a role in recognizing the NE in different degrees. Further studies, such as (Benajiba et al, 2010) have also confirmed the importance of considering language-independent and language-specific features in Arabic NER

8.3.5 Deep Learning Based ANER

Deep Learning (DL) based ANER has been emerged currently. Researchers have proposed different DL algorithms (e.g. DNN) for ANER (Khalifa and Shaalan 2019).. Currently, NER is based on deep learning approaches, which supported for multi-dimensional input data (Ju, Miwa, and Ananiadou 2018). In (Ali, Tan, and Hussain 2018), a novel hybrid deep learning scheme is recommended for word level and character level representation of Arabic texts. Bidirectional Long Short-Term memory (LSTM) and CRF are combined to use as the hybrid scheme for ANER. Combination ANER approach on basis of deep learning must require huge volume of corpus. Both LSTM and CRF increase computational overhead for ANER system. In (Gridach and Haddad 2018), researchers have designed

a novel architecture based on neural networks. It is a consolidation of bidirectional Gated Recurrent Unit (GRU) and Conditional Random Fields (CRFs). The minimal set of features are used (pretrained character level and word level embedding). Most significant improvement in this work is that it uses minimal set of features for ANER. This work is failed to produce high recognition rate.

8.4 Proposed ANER using Bi-LSTM-CNN-CRF

In this chapter, we firstly describe the problem statement. Then, we describe briefly of our proposed ANER system. Figure.2. indicates the proposed system architecture.

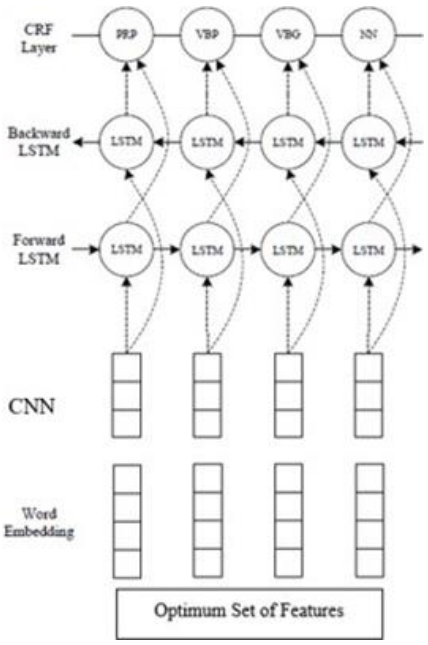


Figure 17: The main architecture of our NER neural network.

8.4.1 Problem Statement

As we mentioned earlier, the Arabic language is a highly inflectional language. The task of ANER is considered as an optimization problem: to classify a sequence of words according to the sequence of three classes since the output of the NER is tagging text, we must have a set of categorizations. In this section we presented current problem statement in ANER. Our POS tagging in chapter is implemented in this chapter. Our POS tagging is adaptable for both small-scale and large-scale Arabic corpora. In (Helwe and Elbassuoni 2019), deep co-learning is introduced for ANER. Deep Co-learning algorithm is tested on three different datasets for categories of Person, Organization, Location and Others. Accuracy of Deep Co-Learning is less due to insufficient features extraction and lack of pre-processing since it does not consider as morphological features and POS tagging.

8.4.2 System Overview

Our proposed ANER translation comprised of five steps: Text- Pre-processing, Multi-Feature Extraction, Feature Selection, Entities Recognition, and translation. Both CNN, Bi-LSTM and CRF are well defined classifiers, and each of them performed well when applied separately. By combining them, we achieved excellent performance regarding F-measure, precision, and recall. To make the Arabic Named Entity Recognition and Translation system domain independent, the researchers introduced various processes

of recognition. We used corpus to translate the Arabic Named Entity Recognition to English Language.

8.4.2.1 Text Pre-processing

The pre-processing stage is one of the crucial steps in text classification. It retrieves the canonical representation of textual illustration. A predictable pre-processing phase usually consists of the following steps: tokenization, stop word removal, morphological analysis and POS tagging. Before pre-processing we describe of the named entity recognition for Arabic language as following:

- In Arabic language, a character may contain three different forms and each form position corresponds to the character in the word (Beginning, End and Middle).
- It does not have capital letters and this characteristic means a considerable obstacle for the task of named entity recognition since other languages uppercase letters defines most important feature.
- It comprised of short and long vowels, but short vowels are not used anywhere (i.e. newspapers) and this results NER task quite high ambiguity in texts (disambiguation based on the short vowels).
- Arabic language is very complex due to its morphology inflections.

To do pre-processing of named entity, we used Madamira Morphological Analyzer and POS tagging (Pasha et al. 2014). The pre-processing steps are described:

A. Tokenization: It is very important in Natural Language Processing (NLP). It is the process of scanning of each document word by word, extracts the words in the document, and divides each document into multiple tokens based on whitespace characters.

B. Stop Words Removal: This is the second step in preprocessing. This process takes place to remove the Arabic stop words, e.g., he, she, and, which, and so forth. It removes the stop words such as pronouns, conjunctions, numbers, and propositions.

In the Arabic language, identifying and classifying names is not an easy task; we took a manually- created stop word list for Arabic from the web and compared our text to the stop word list. Then, we applied the frequency and eliminated the most frequent word.

C. Morphological Analysis: In Arabic language, morphological analysis can be made by three techniques: root, stemming and lemma.

In the stemming process, the words are reduced to their roots; that is, the prefixes and suffixes are removed. Stemming is done to improve the efficiency of the classification by reducing the number of

terms being input to the classification and working with Arabic texts without stemming results in a massive number of tokens being inputted to the classification, which will inevitably degrade the system performance, increasing the classifier complexity and reducing its scalability.

D. POS Tagging: It is the process of annotate proper grammatical meaning for the given input text. It is an essential step for most NL applications. In order to tag POS, we take the full tag set with information of each token of the words

8.4.2.2 Multi-Feature Extraction

We considered the set of features of Arabic language such as contextual features using -/+1 token window. Lexical features using N-Grams n range from 1 to 3. Gazetteers (dictionary) automatically harvested and manually cleaned named entities, POS, and BPC using Madamira tool (Pasha et al. 2014); morphological features are grammatical features (noun, verb, adjective); nationality using nation list manually Table 1 shows the multi-feature extracted set extracted using Madamira Morphological Analyzer and POS tagging. Detailed descriptions of some of the features are follows:

(i). Contextual (CXT) features: Contextual features are local features defined over the targeted word. This targeted word decision is based on the features of its two immediate left and right neighbors. Usually, they are defined in terms of a sliding window of tokens/ words, i.e. a-/+1 window

(ii). Lexical features (LEXi): lexical features avoid complex morphology by extracting the word prefix and suffix sequence of a word from the character n-gram of leading and trailing letters where n ranges from 1-3. Lexical features denoted as token.

(iii). Gazetteers features (GAZ): Gazetteers include harvested, and hand-crafted dictionary entries pre-defined NEs. It manually uses three cleaned classes/ Gazetteers:

- PER (Person NE class)
- GPE (Geopolitical/Location Entity NE class),
- ORG (Organization NE class)

Gazetteers belong to only one of the NEs types, e.g. Location Gazetteers consist of names of countries, states, areas, towns, villages, political regions, continents and so on. For person names, Gazetteers must be in full or partial Named entity, i.e. person names (first name, last name, full forms, nickname, and middle name) not to be considered as a single entry. It could have separate gazetteers for each entry.

However, this is always true for person names and does not hold for location and organization NEs. **(iv). POS (Part of Speech) Tag and Base Phrase Chunk (BPC):** POS tagging is one of a good distinguishing feature for Arabic NEs. It identifies the specific morphological information and finds proper nouns of each word. POS tag and BPC automatically tagged using Madamira Tool (Pasha et al. 2014).

(v). Morphological (MORPH) Features: Generally, an Arabic word contains a rich set of morphological information since it is a language specific feature. We found that Madamira Morph is more useful to extract the relevant morphological information about a person, gender, number, definiteness, and each word aspects.

vi). Corresponding English Capitalization (CAP) features: This feature indicates whether an Arabic text has any capital letters or not.

(vii). Syntactic based features: It estimates writing style at the sentence level like length of sentence and use of function weights, etc.

(viii). Structure based features: It reflects organization of texts (lengths of paragraph and chapter)

8.4.2.3 Feature Selection

Feature selection plays a critical role in text categorization. Due to several reasons, feature selection for Arabic language recognition is still few compared with other languages. The main reason is that the nature of Arabic writing structure is differing to other languages. Arabic documents may contain a large number of redundant and irrelevant words. It deteriorates the performance of the learning algorithm. Therefore, feature selection is highly required to avoid this situation. In this section, we used the TF-IDF term weighting scheme for feature selection to exploit the most important and discriminant features for Arabic Named Entity Recognition.

discriminant features for Arabic Named Entity Recognition.

TF-IDF scheme is one of the best schemes for term weighting. It is widely used for feature selection in information retrieval, and as a metric for estimating the importance of a word in a document within a collection. TFIDF is a product of two schemes, Term Frequency (TF) and Inverse Document Frequency (IDF). Term Frequency or local frequency refers to a word or a stem that appears many times in a training document. It is assumed to be more important than a root word (stem) that appears only once. In

other words, TF represents the number of times a term t appears in document d. Inverse Document Frequency (IDF) or global frequency refers to a stem or word that occurs a few times in training documents. It is assumed to be a better discriminator than a stem or word that appears in most of the training documents. In other words, IDF refers to discriminating measures for a term t in document d.

$$TF(t,d) = \frac{F_d(t)}{\max_{w \in d} F_d(w)}$$

$$IDF(t,D) = \ln\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$$

$$TF - IDF(t,d,D) = TF(t,d)$$

$$TF - IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

Where

$$Fd(t) = \ Frequency\ of\ term\ t\ in\ dcument\ d$$

$$D = Corpus\ of\ Documents$$

Selection of optimum set of features is an optimization problem, which is solved using spider monkey optimization algorithm. Based on the TF-IDF of Arabic texts, the optimum sets of features are selected. This optimization algorithm aids to analyze the relevancy among multiple features, which is performed on two ways: Feature-Feature relevancy computation, and Feature-Class (P or O or L) relevancy computation. SMO is a stochastic optimization algorithm, which runs similar to the population based optimization and inspired by spider monkeys social behaviour. It mimics FFSS (Fission-Fusion Social Structure) based spider monkeys foraging behaviour. The main steps involving in SMO are local leader process, global leader process, local leader learning process, global leader learning process, local leader decision process, and global leader decision process. In SMO, Spider Monkeys (SM_i) position is updated based on the selection probability (SP_i), which is computed by fitness value. Of spider monkey computed based on the following.

$$SP_i = 0.9 \times \frac{i_{fitness}}{max_{fitness}} + 0.1$$

Where $i_{fitness}$ is the fitness value of spider monkey i, and $max_{fitness}$ represents the maximum fitness value of the group.

Basic Feature Set	Aspect features	BF1	No. of commands
		BF2	No. of perfective
		BF3	No. of imperfective
	Case Features	BF4	No. of nominative
		BF5	No. of genitive
		BF6	No. of accusative
	Gender Feature	BF7	Masculine
		BF8	Famine
	Grammatical person features	BF9	1st person
		BF10	2nd person
		BF11	3rd person
	Number features	BF12	No. of singular words
		BF13	No. of plural words
		BF14	No. of dual words
	Mood features	BF15	Jussive
		BF16	Subjunctive
		BF17	Indicative
	POS tag features	BF18	No. of words
		BF19	No. of foreign letters
		BF20	No. of nouns
		BF21	No. of pronouns
		BF22	No. of digital numbers
		BF23	No. of proper nouns
		BF24	No. of interjections

		BF25	No. of adjectives
		BF26	No. of conjunctions
		BF27	No. of adverbs
		BF28	No. of verbs
		BF29	No. of punctuation
		BF30	No. of particles
		BF31	No. of abbreviations
		BF32	No. of prepositions
	State features	BF33	No. of definitive
		BF34	No. of indefinitive
		BF35	No. of construct
	Voice features	BF36	Active voice
		BF37	Passive voice
Advanced Feature Set	Character based lexical features	BF38	Total no. of characters
		BF39	No. of letters
		BF40	No. of white spaces
		BF41	No. of multiple elongation
		BF42	No. of digits
		BF43	No. of tab spaces
		BF44	No. of elongations
		BF45	No. of special characters

	Content specific features	BF46	No. of political terms
		BF47	No. of sports terms
		BF48	No. of social phrases
		BF49	No. of economic phrases
		BF50	Function words
	Structural features	BF51	Total no. of lines
		BF52	Total no. of paragraphs
		BF53	No. of title words
		BF54	No. of blank lines
		BF55	Average no. of words (paragraph)
		BF56	Average no. of sentences (paragraph)
	Syntactic features	BF57	No. of single quotes
		BF58	No. of commas
		BF59	No. of semi-colons
		BF60	No. of double quotes
		BF61	No. of ellipsis
	Word based lexical features	BF62	Total no. of words
		BF63	No. of long words

		BF64	No. of short words
		BF65	No. of dislodgement
		BF66	Average word length

Table 22: Multi Feature Set

8.4.3 Classification

Existing deep learning approaches are not suitable for ANER. we propose a neural network architecture for sequence labeling. It is a truly end-to-end model requiring no task-specific resources, feature engineering, or data pre-processing beyond pre-trained word embeddings on unlabeled corpora. Thus, our model can be easily applied to a wide range of sequence labeling tasks on different languages and domains We used a combination of two algorithms, Bi- LSTM, and CNN. These two techniques improve the classification process and easily handle unknown words and ambiguity. CNN is the multilayer supervised learning framework. Its purpose is to promote the training set. It provides sufficient training data; this feature enables the network to work efficiently regardless of different classes. We first use convolutional neural networks (CNNs) (LeCun et al., 1989) to encode character-level information of a word into its character-level representation. Then we combine character- and word-level representations and feed them into bi-directional LSTM (BLSTM) to model context information of each word. Finally, the output vectors of BLSTM are fed to the CRF layer to jointly decode the best label sequence. As shown in Figure 14, dropout layers are applied on both the input and output vectors of BLSTM. Experimental results show that using dropout significantly

Our end-to-end model outperforms previous state of-the-art systems, obtaining 96.88% F1 for NER. The contributions of this work are (i) proposing a novel neural network architecture for linguistic sequence labeling. (ii) giving empirical evaluations of this model on benchmark data sets for two classic NLP tasks. (iii) achieving state-of-the-art performance with this truly end-to-end system. Figure 18 illustrates the architecture of our network in detail.

8.5 Experimental Results and Analysis

In this section, experimental results are shown to verify the effectiveness of the proposed ANER. For our experiments, we use Java JDK 1.7 with Madamira Morphological Analyzer. This section is subdivided into three sections (dataset description, evaluation metrics and comparative study). We compare the performance of our proposed with some previous works in terms of Precision, Recall, and F-measure.

8.5.1 Dataset Description

In this thesis, we have built our own NER corpus to train our system in order to identify certain types of NEs that were not covered previously. In particular, our NER corpus enables the identification of the following named entities:

- Detailed location, including: Country, city, district, street, and building name
- Date and time
- Phone Number

Arabic NER has gained more attention recently due to the increased availability of annotated Arabic datasets. Arabic NER systems, as in other languages, are domain dependent and mainly trained on news corpora or other well-structured data that uses the Modern Standard Arabic (MSA) variety of the language. In this thesis, the training dataset has been prepared and transformed (i.e. tagged) using our tag schema, in XML format. The total number of NEs/keywords covered in our corpus is 55,760 keywords. A Named Entity in Arabic language exists in many types.

NER Types	Illustrating Example	Translation
Location (street)	شارع الامارات	Emirates Street
Location (place)	برج خليفة	Burj Khalifa (Tower)
Time	10 صباحا	10 Am
Date	جمعة	Friday
Number	20 ,عشرين	Twenty

Table 23: Examples of NER types

Table 23 illustrates some of the NER types implemented in our system, along with illustrating examples. In order to build our Arabic NER corpus, we used the silver standard corpus creation approach proposed in (Hahm et al. 2014). In this approach, the NER corpus is built using Wikipedia as raw corpus, and NE candidate terms identification and annotation is performed automatically using entity classification and disambiguation techniques. Based on (Hahm et al., 2014), each Wikipedia linked term is tagged as a NE,

according to the equation5, in which: NE is a named entity and T is a named entity candidate term (linked term in Wikipedia).

Algorithm 1 was used to build a training dataset for our NER system *(Hahm et al. 2014)*, based on Wikipedia data. Where t is a linked term list, $n_i$is an NE list, and A is a Wikipedia article. Moreover, we built a dictionary i of terms related to locations organizations, hospitals, hotels names, for instance Dubai city, we included all streets and districts, for accurate recognition of locations in the city.

Algorithm 1: Building dataset for NER

```
BEGIN
initialize an empty list Li, S, t
if t in ni, then
   put t in Li
end
for each Li do
    if entry of Li is in A then
       extract sentences from A;
       annotate i as a linked term;
       put sentences in S;
    end
 end
END
```

8.5.2 Testing and Training Data Corpus

KALIMAT corpus that have been built by (El-Haj and Koulali 2013). A publicly available multipurpose Arabic corpus that aims to be a gold standard or baseline corpus. KALIMAT consists of 20,291 Arabic articles collected from an Omani newspaper along with automatically generated extractive summaries (20,291 single and 2057 multi-document summaries) of theses and articles by previously developed summarizers.

KALIMAT consists of: 1) 20,291 Arabic articles collected from the Omani newspaper Alwatan by (Abbas et al. 2011) 2) 20,291 extractive single document system summaries, 3) 2,057 multi-documentsystem, 4) 20,291 Named Entity Recognised articles, 5) 20,291 part of speech tagged articles, and 6) 20,291 morphologically analysed articles. The data collection articles fall into six categories: culture, economy, local-news, international-news, religion, and sports.

The Arabic Corpora[9], which itself consists of two corpora. The first is the Khaleej-corpus which was

[9] https://sourceforge.net/projects/arabiccorpus/files/

acquired from thousands of articles downloaded from the Akhbar Al Khaleej newspaper's website (Murad Abbas and Smaili 2005). It contains 5690 documents which correspond to nearly 3 million words. It is divided into four topics (categories). The second is the Watan-2004 corpus (M Abbas, Smaili, and Berkani 2011a), which contains 20291 documents correspond to nearly 10 million words organized in six topics (categories): Culture, Religion, Economy, Local News, International News and sports. Punctuation was omitted intentionally in this corpus to make it more useful for language modeling.

The EASC [10](Essex Arabic Summaries Corpus) (Attia et al. 2016), a natural Arabic language resource divided into ten categories: Art and Music, Education, Environment, Finance, Health, Politics, Religion, Science and Technology, Sport, and Tourism. The EASC contains 58611 words acquired from 153 articles.

The BBC Arabic corpus from BBC Arabic collected from website bbcarabic.com, the corpus includes 4,763 text documents. Each text document refers to 1 of 7 categories (Middle East News 2356, World News 1489, Business & Economy 296, Sports 219, International Press 49, Science & Technology 232, Art & Culture 122). The corpus contains 1,860,786 (1.8M) words and 106,733 distinct keywords after stop words removal.

The CNN Arabic (El-Khair 2016), which is collected from the CNN Arabic website cnnarabic.com, the corpus includes 5,070 text documents. Each text document belongs to 1 of 6 categories (Business 836, Entertainments 474, Middle East News 1462, Science & Technology 526, Sports 762, and World News 1010). The corpus contains 2,241,348 (2.2M) words and 144,460 distinct keywords after stop words removal.

The last corpus is OSAC (Saad 2010). OSAC Arabic corpus collected from multiple websites, the corpus includes 22,429 text documents. Each text document belongs to 1 of 10 categories (Economics, 6 History, Entertainments, Education & Family, Religious and Fatwas, Sports, Heath, Astronomy, Low, Stories, Cooking Recipes). The corpus contains about 18,183,511 (18M) words and 449,600 distinct keywords after

The ANERcorp corpus developed by (Benajiba, Rosso, and BenedíRuiz 2007) had been used by several

[10] 1 https://sourceforge.net/projects/easc-corpus/

systems. The ANERCORP dataset is a manually annotated corpus which is freely available for research purpose created by Yassine Benajiba in ANER. The corpus was annotated by one person in order to guarantee the coherence of the annotation, and it has 4901 sentences with 150286 tokens. There are more than 150K tokens in the corpus, and 11% of them are named entities. ANERcorp is easy to parse as each line contains a single word. Each token in this corpus is tagged with one of the followings: person, location, company or organization, and others. ANERcorp was annotated into various classes that follow:

- B-PERS: The beginning of the person name
- I-PERS: The inside of the person name
- B-LOC: The beginning of the location name
- B-ORG: The beginning of the organization name
- I-ORG: The inside of the organization name
- B-MISC: The beginning of the miscellaneous word
- I-MISC: The inside of the miscellaneous word
- O: The word is not a named entity. It refers to some other named entitie

8.5.3 Evaluation Metrics

Four common measures are undertaken for evaluation such as Precision, Recall, the Harmonic Mean of F-measure, and Accuracy. The possibilities of finding the Arabic-named entities.

- *True Positive (TP):* The number of the correctly-found named entity.
- *False Negative (FN):* The number of named entities that are not found by the system
- *False Positive (FP):* The number of found named entities which do not exist in the corpus.
- *True Negative (TN):* The number of entities not correctly found by the system

Precision

Precision is defined as the correct number of entities found divided by the total number of found.

$$p = \frac{\text{Number of correct entities found}}{\text{total number of entities}}.$$

$$p = \frac{TP}{TP + FP}$$

Recall

Recall is defined as the number of correct entities found divided by the total number of correct entities.

$$R = \frac{Number\ of\ correct\ entities\ found}{Total\ number\ of\ correct\ entities}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

F-measure

Assurance of the system performance is not concluded by precision and recall alone. When using precision and recall, some problems may arise in some cases. To overcome this problem, the mean of both precision and recall can be measured. It is known as ―F-measure‖, which is calculated below:

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$

9.5.2.1 Main Results

We first run experiments to dissect the effectiveness of each component (layer) of our neural network architecture. We compare the performance with three baseline systems the bidirection LSTM, and BLSTM-CNNs, the combination of BLSTM with CNN to model character level information. We used our own corpus as a training corpus, and Kalimat Corpus for testing our model. All these models are run using the same hyper-parameters as shown in Table 24. According to the results shown in Table 25, CNN-BLSTM models significantly outperform the BLSTM model, showing that character level representations are important for linguistic sequence labeling tasks. This is consistent with Finally, by adding CRF layer for joint decoding we achieve significant improvements over BLSTMCNN models for NER on all metrics. This demonstrates that jointly decoding label sequences can significantly benefit the final performance of neural network models.

Layer	**Hyper Parameter**	**NER**
CNN	Window size	3
	Number of filters	30
LSTM	State size	200
	Initial state	0.0
Dropout	Dropout rate	.5

Table 24: hyper-parameters

	Person			**Location**			**Organization**		
Model	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1

CNN	86.3	84.6	85.4	87.8	86.9	87.3	85.4	85.3	85.3
CNN-BLSM	87.5	86.9	87.2	88.2	88.6	88.4	87.3	86.9	87.1
CNN-BLSTM-CRF	94.6	93.8	94.2	95.5	95.1	95.3	93.5	92.7	93.1

Table 25: Performance of our model together with three baseline systems

9.5.2.2 Comparative Study

In this section, a number of experiments are conducted to evaluate the performance of the proposed system with the best previous works using four different performance metrics such as Precision, Recall, F-measure and Accuracy.

we examined the system performance against other related works. Hence, we need to use the same corpus used by other's work. Based on our survey we found that the "ANERcorp" corpus developed by Benajiba and Rosso had been used by several systems. Therefore, we chose "ANERcorp" to evaluate the system's results with those performed by others' such as ANERsys 2.0 (Benajiba and Rosso), RENAR (Zaghouani,) and Rule Based Approach(Al-Ahmari) .

The dataset distribution is follows: Person: 39%, Location: 30.4%, Organization: 20.6%, and Miscellaneous: 10%.

In addition, we present the results for proposed CNN-Bi-LSTM-CRF for better comparison.

Source	**Technique**	**Person**	**Location**	**Organization**
Benajiba and Rosso	ANERsys	*56.3*	*91.7*	*48*
Zaghouani	RENAR	*71.2*	85.2	47
Al-Ahmari	Rule Based Approach	*80.7*	93.2	75.4
Al Thobaiti	Maha Althobaiti	*85.9*	88	84.3
Our System	CNN-Bi-LSTM- CRF	*95.2*	95.8	95.4

Table 26 Comparative Results for Precision

9.5.2.3 Precision

The percentage of correctly classified entities in relation to the total number of entities in corpus is called precision. The higher precision shows the system has produced better performance for three classes such as precision, location and organization. Table 26 shows the comparison results of precision for person, location and organization.

Our proposed system is evaluated on a corpus of 150K tokens and the overall performance obtained for

various categories such as a person, location, and organization for a precision of *95.2%, 95.8%, and 95.4%, respectively*. Precision in recognition of location and organization is highest compared to entity precision. The reason behind this is because location names are less ambiguous compared with person or organization names. Multi-feature extraction part has improved the precision rate is high. Existing works on ANER are based on the basic features extraction and poor to classify Arabic texts. In our proposed ANER, we attained high value of precision due to the multi-feature extraction and deep learning based classification.

9.5.2.4 Recall

Recall is one of the important metric similar to precision, which must be higher to prove the performance of the proposed ANER is correctly classified the named entities (P, L, and O). Table 27 shows the performance of the proposed as well as the previous ANER. In previous ANER systems, recall value relatively low degraded the system performance compared to precision, and f measures. Since its training sets and features may not cover some NE occurrences from documents. In this research, we combined two classification algorithms that are boosting the performance of our proposed system. However, the performance of the proposed ANER system is great rise in recall due to an accurate of recognition of the given Arabic texts. The proposed ANER for recall performance obtained for various categories such as a person, location, and organization is 92.3%, 94.6%, and 95.2%, respectively.

Source	Technique	Person	Location	Organization
Benajiba and Rosso	ANERsys	48.6	82.2	45
Zaghouani	RENAR	45.2	85.2	47
Al-Ahmari	Rule Based Approach	52.2	85.4	45.2
Al Thobaiti	Maha Althobaiti	51.2	62.5	40.3
Our System	CNN-Bi-LSTM- CRF	92.3	94.6	95.2

Table 27: Comparative results for Recall

9.5.2.5 F-measure

F-measure is the most significant metric which is the harmonic mean of precision and recall. When the feature set achieved high precision and recall then the system will have obtained higher F-measure. Table 28 shows the performance of the results of proposed ANER with the previous works. If the trained

feature set is worsened, then precision and recall will be low which will result in the reduction of the overall mean value of the F-measure of the system. By using more number of feature set, the performance of the proposed ANER is highly improved. Table 6 shows the performance F measure. The obtained result shows that our proposed classifiers outperform than others. The overall performance obtained for various categories such as a person, location, organization, and miscellaneous types was an *F-measure of 93.7%,95.2%,95.3%* respectively.

When compare our proposed ANER with previous works, our proposed ANER has obtained better performance due to the following: Arabic text pre-processing comprised of tokenization of words, tokenized words normalization, elimination of predefined words i.e. stops words, morphological analysis, POS tagging and term weighting (TFIDF). TF-IDF and SMO objective is to indicate Arabic text in a high quality. Based on the weights of terms (words), SMO is applied to select the optimum number of features.

Source	Technique	Person	Location	Organization
Benajiba and Rosso	ANERsys	52.2	86.7	46.5
Zaghouani	RENAR	55.3	85.2	47.0
Al-Ahmari	Rule Based Approach	63.4	89.1	56.5
Al Thobaiti	Maha Althobaiti	64.2	73.1	54.5
Our System	CNN-Bi-LSTM- CRF	93.7	95.2	95.3

Table 28: Comparative Results for F-Measure

In this chapter, a new approach is proposed to tackle the problem of ANER in an innovative way. The key difference between this work and the previous ones is that our model uses a hybrid scheme that combines of three algorithms one of them is a CNN which gives us the insight to move fully to neural network approaches, connected Bi-LSTM and the last one is CRF To evaluate the performance of our model in comparison to other common state-of-the-art architectures, a well-known dataset is used, the ANERcorp dataset. The proposed model outperforms the current state-of-the-art models by a considerable margin.

8.6 Chapter Summary

Named entities generates serious problems for automatic machine translation systems and often cause translation failures beyond the local context, affecting both the overall morphosyntactic well-formedness of sentences and word sense disambiguation in the source text. In this chapter, we newly introduced a multi-feature based CNN with Bi-LSTM and CRF for ANER system. This combination is reached high level of performance than the others. Without the extra steps such as pre-processing, feature extraction, and feature selection, NER leads to very low precision and recall for Arabic language. So in our proposed ANER work, several steps are presented to boost the system performance. In the text pre-processing stage, we used Madamira tool since most of the described tools here require little or manual effort. However, Madamira has the great potential to pre-process Arabic documents. In this toll, we implemented four steps: tokenization, stop words removal, morphological analysis and POS tagging. Also, multi-feature extraction and feature selection is further presented to improve the system performance by VSO model and SMO algorithm. Finally, the optimum sets of features are given as an input to CNN, Bi-LSTM and CRF. The resulting system is fast and robust and can be easily applied to large datasets. Experimental results on ANERcorp dataset shows Precision, Recall, and F- measure values of 94.5%, 95.7%, and 96.3%, respectively.

Chapter Nine: Machine translation for Arabic Word Sense Disambiguation

9.1 Introduction

Word sense disambiguation (WSD) is the issue of distinguishing the sense or the meaning of a word in a particular context. WSD, in Natural Language Processing (NLP), instantly determines the sense of a word by taking into account the related context (Alkhatib and Shaalan 2018). Word Sense Disambiguation (Navigli, 2009) has recently received significant attention, being part of the deep- rooted challenges in Arabic NLP. In fact, by addressing lexical ambiguity, effectual WSD model brings many advantages to various downstream implementations, from Information Retrieval and Extraction to Machine Translation.

There are various types of ambiguity in Arabic language; a large number of words are based on specific contexts in different characteristics. To illustrate, دين is an Arabic word that has two meanings: one m e a n s religion and the other means rent money. A person using common sense can easily distinguish such ambiguity, whereas the difference is not identified by machine translation. As an alternative, MT needs a further complicated analysis and calculation to accurately distinguish the meaning, a process referred to as WSD (Alkhatib and Shaalan 2018).

WSD method reveals that the two words, prior to and following a word that is ambiguous, are adequate for its disambiguation in nearly every language (Mohamed and Tiun 2015). The details taken from the local context are not enough in all occasions, for the Arabic language. For the sake of solving this issue, the Arabic WSD system, which is not just based on the local context, but based on the global context taken out as well, has been presented.

All WSD methods make use of words in a phrase to collaboratively disambiguate one another (Agirre et al. 2010; Ponzetto and Navigli 2010). The differentiation between several propositions resides in the origin and kind of knowledge in a sentence using lexical units. Consequently, each approach is categorized into corpus-based method or knowledge-based method. Corpus-based methods utilize machine-learning techniques to produce word usage models from great number of text samples. Monolingual, bilingual, raw, or sense-tagged statistical information is obtained from corpora. Instead, knowledge-based methods utilize exterior knowledge resources defining precise sense distinctions to

assign a words right sense in a context. Machine-Readable Dictionaries (MRDs), thesauri, and computational lexicons, such as WordNet (WN), are used by Dagan and Itai, 1994, and (Gale, Church, and Yarowsky 1992). Statistical data of the monolingual corpus of different language was first used by (Dagan and Itai 1994), to solve lexical ambiguities in a language. This approach utilizes the differences between words mappings to senses in different languages.

Our focus in this article is supervised WSD. However, we deviate from prior methods and take another distinct view of the task: we aim to model target text joint disambiguation as a whole with regard to sequence labeling issue rather than framing a separate classification problem for each given word. Taking this viewpoint into account, a sequence of words is translated into a sequence of possibly sense-tagged tokens by WSD.

Taking this into account, we plan, examine, and then experimentally compare different neural architectures of various complications, which range between a single bidirectional Long Short-Term Memory (Alex Graves, Jaitly, and Mohamed 2013) and a sequence-to-sequence model (Sutskever, Vinyals, and Le 2014). Each of the architectures reflects a certain way to model the problem of disambiguation. However, all of them possess prominent parts that distinguish them from previous WSD methods: their training was end-to-end from text that is sense-annotated to sense labels and they learn from the training data a single all-words model, with no fine-tuning and no local features' explicit engineering.

Many studies show that in order to improve translation quality, one must correctly identify the most likely senses of source-side ambiguous words when selecting target translation (Tang and Xiong 2016;Cho et al. 2013; Zou et al. 2013). Our contributions here are doubled. It is shown, firstly, that a new and effectual alternative to the standard method of supervised Word Sense Disambiguation modeling is represented by neural sequence learning, allowing single all-words approach to challenge a group of word experts and establish state-of-the-art outcomes, at the same time making it not difficult to train, possibly more flexible to utilize in downstream applications, and instantly adjustable to other languages with no need for further sense-annotated data (see Section 6.2); secondly, we conduct an thorough exploratory assessment in which different neural architectures devised for the task were compared (and by some means remained under investigation in preceding literature), various configurations and

training strategies were explored, and their points of weakness and points of strength on every standard benchmark for all-words Word Sense Disambiguation were analyzed.

9.2 Related Work

Nonetheless, the number of Arabic word sense disambiguation (AWSD) researches is restricted and we will propose the latest work in AWSD.

In (Diab 2003) established the unsupervised method, Sense Annotations Leveraging Alignments and Multilinguality (SALAAM), that annotates Arabic words with their meanings from the English WordNet by utilizing aligned Arabic-English corpus based on translation correspondence between both Arabic and English words. SALAAM shows 56.9% total accuracy on the task of AWSD.

In (Zouaghi, Merhbene, and Zrigui 2011), used one of the knowledge-based approaches; they evaluated the Lesk algorithm variants to disambiguate Arabic words and used dictionary as a resource. Then they did modifications on Lesk algorithm. Different similarity measures were utilized to show how two ideas in Arabic WordNet can be relatedly alike. Modifications on the Lesk algorithm gave only a precision of 67%.

In (Awajan 2016), a novel Arabic word disambiguation method was presented; that is, Wikipedia and vector space model are utilized and applied, respectively, as a mathematical representation for documents. They implemented the suggested approach by applying cosine similarity measure on three experiments. In the first experiment, one sentence was retrieved for the meanings from Wikipedia and the right meaning was given, while, the second experiment used Tf-Idf vector space model to match the correct sense. Finally, in the third experiment a paragraph was brought from Wikipedia for every meaning. There were no evaluation and accuracy results.

(Menai and Alsaeedan 2012), used Genetic Algorithm (GA) and called it GAWSD. They tested their method by utilizing an Arabic text sample; after that, a comparison was made using naïve Bayes classifier. It was shown from the results that this method shows bigger achievement than naïve Bayes classifier.

Other techniques were used in AWSD like hybrid techniques and fuzzy logic; for example, (Anis Zouaghi, Merhbène, and Zrigui 2012), showed a hybrid technique for AWSD, which combines between methods that are unsupervised and knowledge-based for the sake of giving the right sense of a word,

which is called WSD-AL. By comparison, the hybrid approach surpasses other approaches with regard to accuracy of 79%.

However, (NayerEl-Gedawy 2013), used an approach to solve Arabic WSD after the supervised approach. With WordNet, two fuzzy logic classifiers were employed and contrasted with naive Bayes classifier. These classifiers distinguish the most probable meanings to an ambiguous word and a list comprising ten ambiguous words, which are collected from different researches that handle similar issues. It is asserted that fuzzy logic regards overlapping over different meanings and that it handles lack of clarity and certainty. Their results of the experiments present more precise solutions that were given by fuzzy classifier. This approach accomplished 83% accuracy.

Although, in (Hadni, Alaoui, and Lachkar 2016), English WordNet and Arabic WordNet are used, while relying on machine translation for terms and, for an ambiguity of a word, the nearest notion is chosen, utilizing the association between the word that is ambiguous and the various notions in local context. Different machine-learning and feature selection techniques in their experiments are used in assessing this approach. Their systems' results indicate better performance in the suggested method than in the other AWSD techniques.

(Bakhouche et al. 2015), used Ant Colony Algorithm (Schwab and Guillaume 2011) that can use the Lesk similarity measure for Arabic WSD. We established dictionary through plotting the AWN to the English WN to overpower synset definitions absence in AWN. Here, there was an 80% achievement. (Merhbene, Zouaghi, and Zrigui 2013), presented semisupervised method that integrates a corpus containing the candidate's ambiguous words and Arabic WordNet to establish ambiguous words glosses. For disambiguation process, the right meaning was obtained by usage of weighted directed graph. Recall and accuracy of only 83% were achieved.

The approach of (Bouhriz, Benabbou, and Habib 2016), is different from prior ones, where it is not restricted only to the use of ambiguous words local context features. Ambiguous words' context features that are global and take out from corpus are taken into account. To determine the candidate sense, we utilize local and context vectors for WSD process. We also utilized AWN as a lexical resource for meanings. Here the precision is only 74%.

Nevertheless, when we deal with AWN we face one serious issue, which is the absence of numerous

ideas due to insufficiency of prior AWN and its lack of coverage for all terms. Accordingly, we look for corresponding ideas on WordNet for words that do not exist in Arabic WordNet. In accordance with this trend, we pay particular attention to the (up till now not explored) supervised word sense disambiguation context and conduct an investigation on all-words state-of-the-art methods built upon neural sequence learning and able to disambiguate all target content words in an input text, an important attribute in multiple knowledge-based methods using the Arabic WordNet.

9.3. System Architecture for WSD

WSD is defined with regard to a sequence learning problem in this section. We regard here a variable-length sequence of input symbols and aim to predict a sequence of output symbols, while WSD, in its classical formulation (Navigli, 2009), is shown as an issue of classification for a specified word w in context, where word meanings of w are the class labels.

Input symbols represent word tokens that are extracted from a specific vocabulary. The output symbols are extracted from either a predefined sense inventory (in case of corresponding input symbols being open-class content words, e.g., adjectives, nouns, adverbs, or verbs) or similar input vocabulary (in case of corresponding input symbols being function words, such as determiners or prepositions). A WSD model can, therefore, be defined as a function mapping symbols sequences into symbols sequences.

All-words word sense disambiguation here is not any longer divided into a set of distinguishable and different classification tasks (one for each target word) despite being dealt with right away at sequence level, using a model that deals with every disambiguation choice.

Three separate methods were represented for achieving the above: word embedding (Section 4.1), a traditional LSTM model (Section 4.2), a variant incorporating an attention system (Section 4.2), and a sequence-to-sequence architecture (Section 4.4).

Preprocessing Steps

Firstly, sentences moved towards cleaning and pre-processing stage, for the sake of removing unnecessary tokens and symbols. This stage's objective is maximizing the amount of terms whose embeddings may be detected in pretrained word embedding model. We follow the following procedures for cleaning Arabic text:

- Removing time, dates, URLs, numbers (Arabic and English), and special symbols.

• Text segmentation, stop words removal, and stemming processes to eliminate affixes of words (suffixes and prefixes).

9.3.1 Word Embedding

We used (Laatar, Aloulou, and Belghuith 2018) to define the method for sense disambiguation of Arabic word based on words embedding under three main steps as illustrated in figure 18.

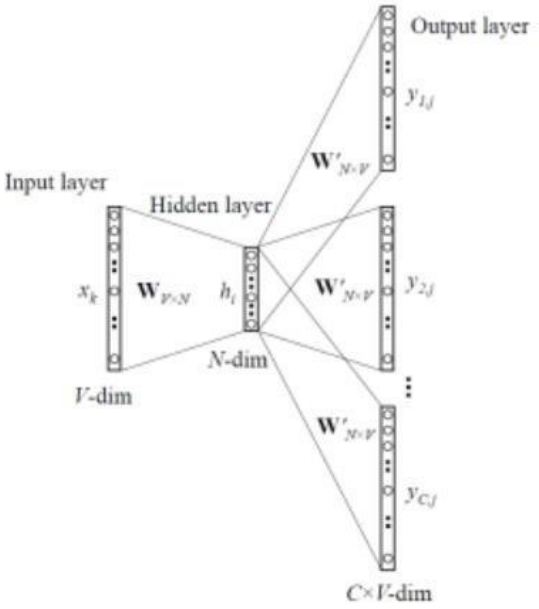


Figure 18: Word Embedding Archetecture

The first step of the suggested model is training Arabic corpus. For our training corpus, we opted Arabic WordNet dictionary (Alkhatib, Monem, and Shaalan 2017c) which is originally designed to solve the WSD issue. The dataset comprises around 8500 words. Skip gram is used to train the word vectors from large amounts of text data. Being simple and effective, we choose Skip gram. Skip gram's training goal is to predict the surrounding words given the present word (Mikolov et al., 2013). The second step is to allocate vector representations for the context of use that has a word which is ambiguous and its senses based on their definitions (glosses extracted from dictionaries). Subsequently, we create context vector and sense vectors. Our strategy of creating context vector has summing of words vectors all around a target word as a main feature. Similar to context vector generation, the sum of the entire content word vectors is used as the generation of vectors of senses in every sense description of the word that is ambiguous. The final step is measuring the resemblance between all ambiguous word glosses and the current context by calculating cosine similarities between context vector and sense vectors of the

ambiguous word. Then, we select the sense which gives the maximum cosine similarity as an appropriate sense for the ambiguous word.

9.3.2 Bidirectional LSTM Model

Section 10.3 shows the most direct method created to model WSD, that is, taking into consideration a sequence labeling architecture which tags every symbol in the input sequence with a label. LSTMs are specifically devised to steer clear of the issue of long-term dependency which faced the standard RNN. Our LSTM architecture follows a similar line and uses a LSTM architecture that is bidirectional in (Raganato, Delli Bovi, and Navigli 2017); as a matter of fact, clues of great significance could be anywhere in the context (not necessarily before the target) for disambiguating a target word and in order to ensure that a model is effective, utilizing details from all input sequences at each time step is essential.

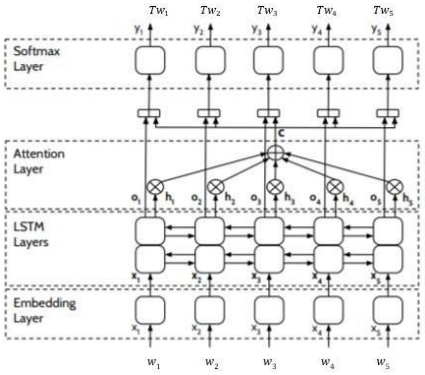


Figure 19: Our bidirectional LSTM sequence labeling architecture for WSD (2 hidden layers)

Architecture. The figure 19shows a drawing of the bidirectional LSTM tagger consisting of the following:

- An embedding layer converting each word through the embedding matrix into a real-valued d-dimensional vector.
- One LSTM retains previous words' context and another retains context of the following words.

Hence, firstly the sentences are transferred to every LSTM; every LSTM has a hidden size h.

- Each LSTM's final output is concatenated to give a vector of $2h$. A layer that is completely connected with softmax activation transforms the output vector at each of the time steps into a

probability distribution over the output vocabulary. They were obtained as forward and

backward pass concatenations.

Training. We train the tagger on a dataset of orders that are labeled and obtained directly from the phrases of a corpus which is sense-annotated, but it is different for the case of numerous real-world datasets, in which just content words subgroup is annotated; thus, the architecture can handle sentences that are annotated partially and fully. Other than tokenization and splitting of sentence, there is no need for preprocessing on the training information.

9.3.3 Attentive Layer

We used the attention mechanism that has already proven to be effective in different NLP tasks (Bahdanau et al., 2015; Vinyals et al., 2015). The resulting attentive tagger of bidirectional LSTM model increases the original architecture with attention layer, in which the context vector is calculated from every hidden state of both taggers. The attentive tagger reads the whole input sequence first to build context vector and utilize it for the sake of prediction of output label at each of the time steps, by its concatenation with bidirectional LSTM model output vector as Figure 2 shows.

Architecture. Figure 1 presents a sketch of our models consisting of the following:

- An embedding layer converting each word through the embedding matrix into a real-valued d-dimensional vector.
- One or more stacked layers of bidirectional LSTM (Alex Graves and Schmidhuber 2005). After that, the output and hidden state vectors at the time step are acquired as forward and backward pass vectors concatenations.
- The layer that is completely connected to softmax activation turning output vector into probability distribution over output vocabulary at each time step.

9.3.4 Sequence-to-Sequence Model

In Section 3.2, attentive tagger carries out a two-pass approach through reading the input to build context vector first and then estimating each element's output label. Taking the previous into consideration, we can constructively regard the attentive architecture to be an encoder. Then, an

additional generalization of the model is an absolute encoder-decoder architecture (Sutskever, Vinyals,

and Le 2014) in which word sense disambiguation is considered sequence-to-sequence mapping (sequence-to-sequence WSD), that is, “translating” word sequences into sequences of tokens which are potentially sense-tagged.

In the WSD context that is formulated as sequence learning issue, labeled sequences training sets are taken as input by sequence-to-sequence model (see Section 3.1) and while substituting every content word with its most appropriate sense, sequence-to-sequence model duplicates an input. That is to say, we consider sequence-to-sequence WSD a mixture of two subtasks:

- The memorization task, in which the method duplicates input sequence token by token

at time of decoding.

- The real disambiguation task, in which the method substitutes content words over the input sequence with the most appropriate meanings from sense inventory.
- Multiword expressions (like phrasal verbs or nominal entity mentions) are substituted by sense identifiers in the latter stage, thus providing output sequence with a length that is probably not the same.

Architecture. In Sections 4.3 and 4.4, encoder-decoder architecture forms a general principle over the two methods. Specifically, in the middle of encoder and decoder modules one bidirectional LSTM layer or more were covered. Encoder uses embedding layer (see Section 4.4) as a way of turning input symbols into embedded representations, then gives it to bidirectional LSTM layer, and finally builds context vector (e.g., hidden state of bidirectional LSTM layer following the reading of the entire input sequence) or computes the weighted sum discussed in Section 4.4 (in case of attention mechanism being utilized). Context vector is sent to decoder in either case that sequentially creates output symbols based on c and present hidden state, utilizing one bidirectional LSTM layer or more as in encoder module.

Alternative to giving c to decoder at first time step only (Sutskever, Vinyals, and Le 2014), we condition each output which allows the decoder to peek at every step into the input, as shown in (Cho et al. 2014b). Lastly, the present output vector of final LSTM layer is converted into probability distribution over the output vocabulary by a fully connected layer with softmax activation. Figure 3 presents full encoder-decoder architecture (i.e., attention mechanism).

9.4 Multitask Learning

Several recent contributions in English language (Alonso and Plank 2016; Bjerva, Plank, and Bos 2016; Raganato, Delli Bovi, and Navigli 2017) presented multitask learning efficacy (Caruana, 1997, MTL) in case of sequence learning, but as per our knowledge it has not been applied yet.

For each additional task we describe an auxiliary loss function as in (Raganato, Delli Bovi, and Navigli 2017). The total loss is calculated by finding the sum of major loss (i.e., that related to labels of word sense) and entire auxiliary losses that are considered.

With regard to the architecture, by including two softmax layers besides that in the real architecture, we consider the model described above and modify it. This is illustrated in Figure 4 for the attentive tagger of Section 3.3, taking into consideration POS (Pasha et al., 2014) and coarse-grained semantic labels (LEX) which are based on the WordNet (Alkhatib, Monem, and Shaalan 2017c) lexicographer files and coarse-grained semantic categories that are related manually to entire synsets in WN based on logical and syntactic groupings.

Lastly, present output vector of the final LSTM layer is converted into probability distribution over output vocabulary by a fully connected layer with softmax activation. Figure 20 shows full encoder-decoder architecture (i.e., attention mechanism).

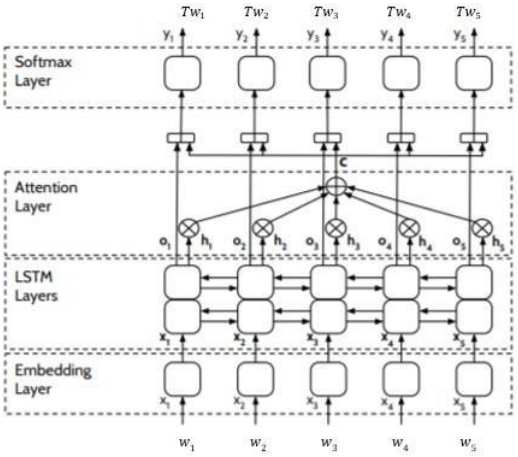


Figure 20: Full encoder-decoder architecture for WSD

9.5 Experiment's Setup

We show the setup of our experimental evaluation all through this section. First, we represent training corpus and entire standard benchmarks for all-words word sense disambiguation; then we describe technical elements on architecture and training procedure for the whole methods that are presented in Section 3 and their multitask augmentations in Section 4.

Benchmarks. The dataset KALIMAT was divided into training, validation, and testing groups with respective percentages of 70%, 10%, and 20%. The dataset was utilized to evaluate our models on Arabic all-words word sense disambiguation task (Section 6.1) and the multilingual all-words WSD (Section 6.2); we examined the multilingual all-words WSD, using two distinct configurations of embedding layer: pretrained bilingual embeddings for the language pairs of interest (Arabic-Arabic and Arabic-English).

Architecture Details. We used a layer of pretrained word embeddings for the sake of setting a level playing field in contrast to methods on Arabic all-words WSD. We used the Skip gram model with dimension of 300 (i.e., d = 300) and a window size of 10 as initialization and throughout the training process we kept them fixed. After that, 2 bidirectional LSTM layers comprising 2048 hidden units (1024 units in each direction) for entire architectures were used.

9.6 Results of Experiments

Here we abbreviate the methods that are based on LSTM tagger (Sections 4.2-4.3) as BLSTM and the sequence-to-sequence methods (Section 4.5) as Seq-to-Seq.

	F-score %
BLSTM	85.4
BLSTM + att.	86.7
BLSTM + att. + LEX	88.33
BLSTM + att. + LEX + POS	89.7
Seq2Seq	83.1
Seq2Seq + att.	83.9
Seq2Seq + att. + LEX	85.5
Seq2Seq + att. + LEX + POS	87.2

Table 29: F-scores (%) for English all-words fine-grained WSD.

9.6.1 Arabic All-Words WSD

Our models' performance for all-words word sense disambiguation that is fine-grained is presented in Table 29. F1-score on each individual test set is described, in addition to F1-score acquired on concatenation of total of four test sets, split by part-of-speech tag.

The final performance through part of speech was compatible with the prior analysis, showing that our models performed better than the entire knowledge-based methods, at the same time acquiring results that are higher-ranking or similar to the top models that are supervised.

	F-scores %
BLSTM + att. + LEX	89.2
BLSTM + att. + LEX + POS	89.6
Seq2Seq + att. + LEX	88.1
Seq2Seq + att. + LEX + POS	88.7

Table 30: F-scores (%) for coarse-grained WSD.

Performance on coarse-grained word sense disambiguation followed similar tendency (Table 30). BLSTM outperformed Seq2Seq.

	F-scores %
BLSTM (bilingual)	86.7
BLSTM (multilingual)	88.3

Table 31: F-scores (%) for multilingual WSD.

Table 31 shows F-score that despite being trained only on Arabic data, bilingual and multilingual models accomplished competitive results. We also observe that in spite of the increased number of target languages treated simultaneously, the total F-score performance significantly stayed the same (and somewhat enhanced) while going from bilingual to multilingual models.

9.7 Chapter Summary

Despite growing interest machine translation, little work has been done along the same lines to train bilingual distribution word embedding to improve machine translation. We embraced a new perspective on supervised Arabic WSD in this article that is, so far, mostly seen as a classification issue at word level and formulated it by utilizing neural sequence learning. For this reason, experimentally different end-to-end models of varying complexities, that is, augmentations that are based on attention mechanism and multitask learning were analyzed, compared, and defined.

In contrast to prior supervised methods, in which a dedicated method should be trained for each content word and every disambiguation target is treated separately, sequence learning methods learn one model in

one move from the training data; after that, they disambiguate together whole target words in input text. Derived systems consistently established state-of-the-art figures in the entire benchmarks for all- words word sense disambiguation, both fine-one of the largest corpora annotated manually with word meanings and coarse-grained, efficaciously showing that the undoubted and well-established word expert assumption of supervised WSD can be overcome, all the time maintaining the precision of supervised word experts.

Chapter Ten: Conclusion and Future work

Translating the Arabic Language into other languages engenders multiple linguistic problems, as no two languages can match, either in the meaning given to the conforming symbols or in the ways in which such symbols are arranged in phrases and sentences. Lexical, syntactic and semantic problems arise when translating the meaning of Arabic words into English. Machine translation (MT) into morphologically rich languages (MRL) poses many challenges, from handling a complex and rich vocabulary, to designing adequate MT metrics that take morphology into consideration.

In this thesis the main platform to perform MT is NNs. We reviewed the fundamentals of deep learning. The procedure for training a neural network was explained and we introduced a number of well-known neural architectures. Both Chapter 1 and Chapter 2 provide background knowledge about the research carried out in the thesis, but the core research is explained in Chapters 3 to 10. Each chapter studies a dedicated research question which is introduced at the beginning of the chapter. The history of the research question and related models are discussed thereafter. The last part of each chapter covers the problem itself and a potential solution.

Chapter 2 – Introduction walks the readers through the history and fundamentals of Arabic language and challenges of Natural Language processing.

Chapter 3 - Background, I provide readers with all the necessary knowledge to fully understand and build a vanilla NMT, which covers details of language model and recurrent neural network, a basic building block for NMT. Several key highlights in this chapter include:

a) Arabic Metaphor in machine translation together with drawbacks of existing approaches, leading to the development of NMT.

b) Arabic NER in machine translation together with drawbacks of existing approaches, leading to the development of NMT.

c) Arabic WSD in machine translation together with drawbacks of existing approaches, leading to the development of NMT.

Chapter 4 – Background: I provide readers with all the necessary knowledge to fully understand and build a vanilla NMT, which covers details of language model and recurrent neural network, a basic building block

for NMT.

Chapter 5 - tools: We introduced an approach that investigates employing deep neural network technology for error detection in Arabic Text. We have developed a systematic framework for spelling and grammar error detection as well as correction at the word level based on a Bidirectional Long- Short-Term Memory (Bi-LSTM) mechanism and Word embedding, in which a Polynomial Network (PN) classifier is at the top of the system. In order to get conclusive results, we have developed the most significant gold standard annotated corpus to date, containing 15 million fully-inflected Arabic words. This data was collected from diverse text sources and genres, in which any erroneous and ill-formed words have been annotated, validated and manually revised by Arabic specialists. The experimental results confirmed that our proposed system significantly outperforms the performance of Microsoft Word 2013 and Open Office Ayaspell 3.4 that have been used in the literature for evaluating similar research.

Chapter 6- tools: we developed Classical Arabic dictionaries and Al-Hadith ontology. Al-Hadith WordNet has demonstrated its capability in a text classification task that we developed for evaluation proposes. The classifier has been applied on around 8500 synsets that include 6126 nominal, 1990 verbal, 310 adjectival, and 71 adverbial expressions. We used the Wordnet as a dictionary resource to translate from Arabic to English and from English to Arabic.

Chapter 7: we presented a well-established machine translation approach for automatically extracting paraphrases, leverages bilingual corpora to find the equivalent meaning of phrases in a single language, is performed by “pivoting” over a shared translation in another language. we revisit bilingual pivoting in the context of neural machine translation and present a paraphrasing model based mainly on neural networks. Our model described paraphrases in a continuous space and generates candidate paraphrases for an Arabic source input. Experimental results across datasets confirm that neural paraphrases significantly outperform those obtained with statistical machine translation, in particular the Google

translator, and indicate high similarity correlation between our model and human translation, making our model attractive for real-world deployment.

Chapter 8: we developed a system based on Hybrid Deep Learning with Evolutionary Algorithm which also known as Convolutional Neural Networks (CNN) with Bi-LSTM and CRF at the top of the model. The proposed hybrid mechanism is tested on ANERCorp. In this paper, three stages are involved: the first stage

is preprocessing where we clean the dataset by several steps (Tokenization, Stop Word Removal, Morphological Analysis and POS Tagging), the second involves multi-features extraction and selection using Vector Space Model (VSO) and Spider Monkey Optimization (SMO) respectively, and the final stage applies the algorithm to classify the data. Experimental results show that our proposed system can find four types of NEs: Person, Location, Organization, and Miscellaneous and achieves high Precision, Recall, F-Measure and Accuracy.

Chapter 9: we solved the WSD challenge in Machine translation, we followed a different perspective and depend on sequence learning to frame the disambiguation issue: we presented and studied thoroughly a series of end-to-end neural architectures directly tailored to the task, from bidirectional Long Short-Term Memory to encoder-decoder models. Our extensive assessment of standard benchmarks and in multiple languages shows that sequence learning allows for more versatile all-words models, consistently leading to state-of-the-art results, even against word experts with engineered features.

In Chapter 10: we tried to summarize all other chapters. We explained the core idea of each chapter and discussed what questions the chapter tried to solve. We reviewed the proposed solutions and enumerated their shortcomings (and also their advantages). For each chapter we mentioned that what questions were solved. In this thesis, we believe that we could contribute to our field as we collected data corpus such as NER corpus, and metaphor corpus, and provided a bilingual wordnet corpus. We designed and implemented different NLP tools such as a Wordnet, and error detection and correction tool. Our solutions introduced new models for embedding learning, and machine translation. We also enhanced the NMT framework to work better for Arabic NLP.

Future Work

Our future work will be a machine translation system for Dialect Arabic language, and to build our data resources that improve the translation from Dialect Arabic to English. Since the use of Dialectal Arabic language has traditionally been restricted to informal personal speech, while writing has been done almost only with using MSA (or its ancestor Classical Arabic). But this situation has been quickly changed, however, with the rapid propagation of social media in the Arabic-speaking part of the world, where much of the communication is composed in dialect (speaking and writing). Also, the focus of the Arabic NLP research community, is turning towards to start dealing with Dialect Arabic. This new focus presents new

challenges, the most obvious of which is the lack of dialectal linguistic resources. Dialectal text, which is usually user-generated, is also noisy, and the lack of standardized orthography means that users often improvise spelling. Dialectal data also includes a wider range of topics than formal data genres, such as newswire, due to its informal natureArabic Dialects present many challenges for machine translation, not least of which is the lack of data resources.

Chapter Eleven :References

A. Alsayadi, Hamzah, and Abeer M. ElKorany. 2016. “Integrating Semantic Features for Enhancing Arabic Named Entity Recognition.” *International Journal of Advanced Computer Science and Applications* 7 (3). https://doi.org/10.14569/IJACSA.2016.070318.

9 Abandah, Gheith A., Alex Graves, Balkees Al-Shagoor, Alaa Arabiyat, Fuad Jamour, and Majid Al- Taee. 2015. “Automatic Diacritization of Arabic Text Using Recurrent Neural Networks.” *International Journal on Document Analysis and Recognition (IJDAR)* 18 (2): 183–97. https://doi.org/10.1007/s10032-015-0242-2.

10 Abbas, M, K Smaili, and D Berkani. 2011a. “Evaluation of Topic Identi Fi Cation Methods on Arabic Corpora.” *JDIM* 9 (5): 185–92.

11 Abbas, Murad, and Kamel Smaili. 2005. “Comparison of Topic Identification Methods for Arabic Language.” *In International Conference on Recent Advances in Natural Language Processing- RANLP* 14 (September): 14–17.

12 Abdallah, Sherief, Khaled Shaalan, and Muhammad Shoaib. 2012. “Integrating Rule-Based System with Classification for Arabic Named Entity Recognition.” In *Computational Linguistics and Intelligent Text Processing*, edited by Alexander Gelbukh, 7181:311–22. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-28604-9_26.

13 Abdel Monem, Azza, Khaled Shaalan, Ahmed Rafea, and Hoda Baraka. 2008. “Generating Arabic Text in Multilingual Speech-to-Speech Machine Translation Framework.” *Machine Translation* 22 (4): 205–58. https://doi.org/10.1007/s10590-009-9054-9.

14 Abdul Karim, Nor Norzelatun ;, and Norzelatun Rodhiah Hazmi. 2005. “ASSESSING ISLAMIC INFORMATION QUALITY ON THE INTERNET: A CASE OF INFORMATION ABOUT HADITH” 10 (2): 51–66.

15 Agirre, Eneko, Oier Lopez de Lacalle, Aitor Soroa, and Informatika Fakultatea. 2010. “Knowledge- Based WSD on Specific Domains: Performing Better than Generic Supervised WSD,” no. 2010: 6.

16 Al-Ahmari, S. Saad, and B. Abdullatif Al-Johar. 2016. “Cross Domains Arabic Named Entity Recognition System.” In , edited by Xudong Jiang, Guojian Chen, Genci Capi, and Chiharu Ishll,

100111I. Tokyo, Japan. https://doi.org/10.1117/12.2240887.

17 Al-Arfaj, and Al-Salman. 2015. “Arabic NLP Tools for Ontology Construction from Arabic Text: An Overview.” *International Conference on Electrical and Information Technologies (ICEIT). IEEE,*.

18 Alex Graves, Navdeep Jaitly and Abdel-rahman Mohamed. 2013. “HYBRID SPEECH RECOGNITION WITH DEEP BIDIRECTIONAL LSTM Alex Graves , Navdeep Jaitly and Abdel-Rahman Mohamed University of Toronto Department of Computer Science 6 King ’ s College Rd . Toronto , M5S 3G4 , Canada,” 273–78.

19 Alfaifi, Abdullah Yahya G. 2015. “Building the Arabic Learner Corpus and a System for Arabic Error Annotation,” 390.

20 Ali, Mohammed, Guanzheng Tan, and Aamir Hussain. 2018. “Bidirectional Recurrent Neural Network Approach for Arabic Named Entity Recognition.” *Future Internet* 10 (12): 123. https://doi.org/10.3390/fi10120123.

21 Al-Jefri, Majed M., and Sabri A. Mahmoud. 2015. “Context-Sensitive Arabic Spell Checker Using Context Words and N-Gram Language Models.” *Proceedings - 2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences, NOORIC 2013*, 258–63. https://doi.org/10.1109/NOORIC.2013.59.

22 Alkhatib, Manar. 2010. “Classification of Al-Hadith Al-Shareef Using Data Mining Algorithm.” *European, Mediterranean & Middle Eastern Conference on Information Systems*, no. January 2010: 1–23.

23 Alkhatib, Manar, Azza Abdel Monem, and Khaled Shaalan. 2017a. “A Rich Arabic WordNet Resource for Al-Hadith Al-Shareef.” *Procedia Computer Science* 117: 101–10. https://doi.org/10.1016/j.procs.2017.10.098.

24 Alkhatib, Manar, and Khaled Shaalan. 2018. “The Key Challenges for Arabic Machine Translation.” In *Intelligent Natural Language Processing: Trends and Applications*, edited by Khaled Shaalan, Aboul Ella Hassanien, and Fahmy Tolba, 740:139–56. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-67056-0_8.

25 Allam, Ali, Mohamed Kouta, and Mohamed Sakre. 2015. “Automated Construction of Arabic-

English Parallel Corpus." Unpublished. https://doi.org/10.13140/rg.2.1.2135.0880.

26 Almaayah, Manal, Majdi Sawalha, and Mohammad A M Abushariah. 2015. "A Proposed Model for Quranic Arabic WordNet."

27 Alonso, Héctor Martínez, and Barbara Plank. 2016. "When Is Multitask Learning Effective? Semantic Sequence Prediction under Varying Data Conditions." *ArXiv:1612.02251 [Cs]*, December. http://arxiv.org/abs/1612.02251.

28 Al-Raisi, Fatima, Abdelwahab Bourai, and Weijian Lin. 2018. "Neural Symbolic Arabic Paraphrasing with Automatic Evaluation." In *Computer Science & Information Technology*, 01–13. Academy & Industry Research Collaboration Center (AIRCC). https://doi.org/10.5121/csit.2018.80601.

29 Al-tahrawi, Mayy M, and Sumaya N Al-khatib. 2015. "Arabic Text Classification Using Polynomial Networks." *Journal of King Saud University - Computer and Information Sciences* 27 (4): 437– 49. https://doi.org/10.1016/j.jksuci.2015.02.003.

30 Al-Tahrawi, Mayy M., and Sumaya N. Al-Khatib. 2015. "Arabic Text Classification Using Polynomial Networks." *Journal of King Saud University - Computer and Information Sciences* 27 (4): 437– 49. https://doi.org/10.1016/j.jksuci.2015.02.003.

31 Althobaiti, Maha, Udo Kruschwitz, and Massimo Poesio. 2015. "Combining Minimally-Supervised Methods for Arabic Named Entity Recognition." *Transactions of the Association for Computational Linguistics* 3 (December): 243–55. https://doi.org/10.1162/tacl_a_00136.

32 Al-zoghby, Aya M, and Khaled Shaalan. 2015. "Computational Linguistics and Intelligent Text Processing" 9042: 405–16. https://doi.org/10.1007/978-3-319-18117-2.

33 Attia, Mohammed, Pavel Pecina, Younes Samih, and Khaled Shaalan. 2015. "Improved Spelling Error Detection and Correction for Arabic," 10.

34 Attia, Mohammed, Pavel Pecina, Younes Samih, Khaled Shaalan, and Josef Van Genabith. 2016. "Arabic Spelling Error Detection and Correction." *Natural Language Engineering* 22 (5): 751–73. https://doi.org/10.1017/S1351324915000030.

35 Awad, Dana. 2015. "The Evolution of Arabic Writing Due to European Influence: The Case of Punctuation," 20.

36 Awajan, Arafat Awajan. 2016. “Arabic Word Sense Disambiguation Using Wikipedia.” *International Journal of Computing and Information Sciences* 12 (1): 61–66. https://doi.org/10.21700/ijcis.2016.108.

37 Ayedh, Abdullah, Guanzheng Tan, and Hamdi Rajeh. 2016. “The Impact of Feature Reduction Techniques on Arabic Document Classification.” *International Journal of Database Theory and Application* 9 (6): 67–80. https://doi.org/10.14257/ijdta.2016.9.6.07.

38 Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. “Neural Machine Translation by Jointly Learning to Align and Translate.” *ArXiv:1409.0473 [Cs, Stat]*, September. http://arxiv.org/abs/1409.0473.

39 “Bahdanau et al. - 2014 - Neural Machine Translation by Jointly Learning to .Pdf.” 2014.

40 Bakhouche, Abdelaali, Tlili Yamina, Didier Schwab, and Andon Tchechmedjiev. 2015. “Ant Colony Algorithm for Arabic Word Sense Disambiguation through English Lexical Information.” *International Journal of Metadata, Semantics and Ontologies* 10 (3): 202. https://doi.org/10.1504/IJMSO.2015.073880.

41 Banerjee, Satanjeev, and Alon Lavie. 2005. “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,” 8.

42 Bannard, Colin, and Chris Callison-Burch. 2005. “Paraphrasing with Bilingual Parallel Corpora.” In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL ’05*, 597–604. Ann Arbor, Michigan: Association for Computational Linguistics. https://doi.org/10.3115/1219840.1219914.

43 Baraka, Rebhi S, and Yehya M Dalloul. 2014. “Building Hadith Ontology to Support the Authenticity of Isnad” 2 (1): 16.

44 Bassil, Youssef, and Mohammad Alwani. 2012. “OCR Post-Processing Error Correction Algorithm Using Google ’ s Online Spelling Suggestion.” *Journal of Emerging Trends in Computing and Information Sciences* 3 (1): 90–99.

45 Benajiba, Yassine, Mona Diab, and Paolo Rosso. 2008. “Arabic Named Entity Recognition Using Optimized Feature Sets.” In *Proceedings of the Conference on Empirical Methods in Natural*

46 *Language Processing - EMNLP '08*, 284. Honolulu, Hawaii: Association for Computational Linguistics. https://doi.org/10.3115/1613715.1613755.

47 Benajiba, Yassine, and Paolo Rosso. 2007. "ANERsys 2.0: Conquering the NER Task for the Arabic Language by Combining the Maximum Entropy with POS-Tag Information," 10.

48 Benajiba, Yassine, Paolo Rosso, and José Miguel BenedíRuiz. 2007. "ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy." In *Computational Linguistics and Intelligent Text Processing*, edited by Alexander Gelbukh, 4394:143–53. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-70939-8_13.

49 Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2009. "A Neural Probabilistic Language Model," 19.

50 Bentivogli, Luisa, and Marcello Federico. 2015. "Neural versus Phrase-Based Machine Translation Quality : A Case Study" 2.

51 Bjerva, Johannes, Barbara Plank, and Johan Bos. 2016. "Semantic Tagging with Deep Residual Networks." *ArXiv:1609.07053 [Cs]*, September. http://arxiv.org/abs/1609.07053.

52 Black, William, Sabri Elkateb, Sackville Street, Horacio Rodriguez, Musa Alkhalifa, Adam Pease, and Christiane Fellbaum. 2006. "Introducing the Arabic WordNet Project," no. Tufis 2004: 295– 99.

53 Boudchiche, Mohamed, Azzeddine Mazroui, Mohamed Ould Abdallahi Ould Bebah, Abdelhak Lakhouaja, and Abderrahim Boudlal. 2017a. "AlKhalil Morpho Sys 2: A Robust Arabic Morpho-Syntactic Analyzer." *Journal of King Saud University - Computer and Information Sciences* 29 (2): 141–46. https://doi.org/10.1016/j.jksuci.2016.05.002.

54 ———. 2017b. "AlKhalil Morpho Sys 2: A Robust Arabic Morpho-Syntactic Analyzer." *Journal of King Saud University - Computer and Information Sciences* 29 (2): 141–46. https://doi.org/10.1016/j.jksuci.2016.05.002.

55 Bouhriz, Nadia, Faouzia Benabbou, and El Habib. 2016. "Word Sense Disambiguation Approach for Arabic Text." *International Journal of Advanced Computer Science and Applications* 7 (4). https://doi.org/10.14569/IJACSA.2016.070451.

56 Boujelben, Ines, Salma Jamoussi, and Abdelmajid Ben Hamadou. 2014. "A Hybrid Method for

Extracting Relations between Arabic Named Entities." *Journal of King Saud University - Computer and Information Sciences* 26 (4): 425–40. https://doi.org/10.1016/j.jksuci.2014.06.004.

57 "Building Hadith Ontology to Support the Authenticity of Isnad Building Hadith Ontology to Support." 2016, no. December 2014.

58 Campbell, W M, K T Assaleh, and C C Broun. 2001. "A NOVEL ALGORITHM FOR TRAINING POLYNOMIAL NETWORKS," 5.

59 Cer, Daniel, Christopher D Manning, and Daniel Jurafsky. 2010. "The Best Lexical Metric for Phrase- Based Statistical MT System Optimization," 9.

60 Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014a. "Learning Phrase Representations Using RNN Encoder- Decoder for Statistical Machine Translation." https://doi.org/10.3115/v1/D14-1179.

61 ———. 2014b. "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation." *ArXiv:1406.1078 [Cs, Stat]*, June. http://arxiv.org/abs/1406.1078.

62 Dagan, Ido, and Alon Itai. 1994. "Word Sense Disambiguation Using a Second Language Monolingual Corpus." *Computational Linguistics* 20 (4): 34.

63 Darwish, Kareem. 2014. "Arabic Information Retrieval." *Foundations and Trends® in Information Retrieval* 7 (4): 239–342. https://doi.org/10.1561/1500000031.

64 David O'Steen, and David Breeden. 2009. "Named Entity Recognition in Arabic: A Combined Approach."

65 Denkowski, Michael, Hassan Al-Haj, and Alon Lavie. 2010. "Turker-Assisted Paraphrasing for English-Arabic Machine Translation," 5.

66 Devlin, Jacob, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. "Fast and Robust Neural Network Joint Models for Statistical Machine Translation." In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1370–80. Baltimore, Maryland: Association for Computational Linguistics. https://doi.org/10.3115/v1/P14-1129.

67 Diab, Mona T. 2004. "Feasibility of Bootstrapping an Arabic WordNet Leveraging Parallel

Corpora and an English WordNet," 8.

68 Do, Thanh Nghi, and Fran??ois Poulet. 2006. "Classifying One Billion Data with a New Distributed SVM Algorithm." *Proceedings of the 4th IEEE International Conference on Research, Innovation and Vision for the Future, RIVF'06*, 59–66. https://doi.org/10.1109/RIVF.2006.1696420.

69 El-Haj, Mahmoud, and Rim Koulali. 2013. "KALIMAT a Multipurpose Arabic Corpus," 4.

70 El-Haj, Mahmoud, Udo Kruschwitz, and Chris Fox. 2011. "Multi-Document Arabic Text Summarisation." *2011 3rd Computer Science and Electronic Engineering Conference, CEEC'11*, 40–44. https://doi.org/10.1109/CEEC.2011.5995822.

71 Elkateb, Sabri, William Black, Horacio Rodríguez, Adam Pease, and Christiane Fellbaum. 2002. "Building a WordNet for Arabic," 29–34.

72 El-Khair, Ibrahim Abu. 2016. "Abu El-Khair Corpus: A Modern Standard Arabic Corpus" 02 (11): 10. Ellman, Jeremy. 2003. "EuroWordNet: A Multilingual Database with Lexical Semantic Networks: Edited by Piek Vossen. Kluwer Academic Publishers. 1998. ISBN 0792352955, {£}58/{$}92.

73 179 Pages." *Natural Language Engineering* 9 (4): 427–30. https://doi.org/10.1017/S1351324903223299.

74 Elsebai, Ali, and Farid Meziane. 2011. "Extracting Person Names from Arabic Newspapers." In *2011 International Conference on Innovations in Information Technology*, 87–89. Abu Dhabi, United Arab Emirates: IEEE. https://doi.org/10.1109/INNOVATIONS.2011.5893875.

75 Faidi, Kaouther, Raja Ayed, Ibrahim Bounhas, and Bilel Elayeb. 2015. "Comparing Arabic NLP Tools for Hadith Classification" 3 (3): 1–12.

76 Farwaneh, Samera, and Mohamed Tamimi. 2012. "Farwaneh, S. and Tamimi, M., 2012. Arabic Learners Written Corpus: A Resource for Research and Learning." *The Center for Educational Resources in Culture, Language and Literacy.*, September.

77 Firat, Orhan, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. "Zero-Resource Translation with Multi-Lingual Neural Machine Translation." *ArXiv:1606.04164 [Cs]*, June. http://arxiv.org/abs/1606.04164.

78 Freihat, Abed Alhakim, Gabor Bella, Fausto Giunchiglia, and Hamdy Mubarak,. 2018. "A Single- Model Approach for Arabic Segmentation, POS Tagging, and Named Entity Recognition." *2nd International Conference on Natural Language and Speech Processing (ICNLSP). IEEE, 2018.*

79 G.A., Miller, R.~Beckwith, C.~Fellbaum, D.~Gross, and K.~Miller. 1993. "Introduction to {WordNet}: On-Line," no. August.

80 Gale, William A., Kenneth W. Church, and David Yarowsky. 1992. "A Method for Disambiguating Word Senses in a Large Corpus." *Computers and the Humanities* 26 (5–6): 415–39. https://doi.org/10.1007/BF00136984.

81 Graves, A., and N. Jaitly. 2014. "Towards End-to-End Speech Recognition with Recurrent Neural Networks." *Proceedings of the 31st International Conference on Machine Learning (ICML-14)* 32: 1764–72.

82 Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. "Hybrid Speech Recognition with Deep Bidirectional LSTM." In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 273–78. Olomouc, Czech Republic: IEEE. https://doi.org/10.1109/ASRU.2013.6707742.

83 Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. "Speech Recognition with Deep Recurrent Neural Networks," no. 3. https://doi.org/10.1109/ICASSP.2013.6638947.

84 Graves, Alex, and Jürgen Schmidhuber. 2005. "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures." *Neural Networks* 18 (5–6): 602–10. https://doi.org/10.1016/j.neunet.2005.06.042.

85 Greenstein, Eric, and Daniel Penner. 2017. "Japanese-to-English Machine Translation Using Recurrent Neural Networks," 1–7.

86 Gridach, Mourad, and Hatem Haddad. 2018. "Arabic Named Entity Recognition: A Bidirectional GRU- CRF Approach." In *Computational Linguistics and Intelligent Text Processing*, edited by Alexander Gelbukh, 10761:264–75. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-77113-7_21.

87 Haddad, Bassam, and Mustafa Yaseen. 2007. "Detection and Correction of Non-Words in

Arabic: A Hybrid Approach." *International Journal of Computer Processing Of Languages* 20 (04): 237. https://doi.org/10.1142/S0219427907001706.

88 Hadni, Meryeme, Said El Alaoui, and Abdelmonaime Lachkar. 2016. "Word Sense Disambiguation for Arabic Text Categorization" 13 (1): 9.

89 Hadni, Meryeme, Said Alaoui Ouatik, and Abdelmonaime Lachkar. 2013. "Effective Arabic Stemmer Based Hybrid Approach for Arabic Text Categorization." *International Journal of Data Mining & Knowledge Management Process* 3 (4): 1–14. https://doi.org/10.5121/ijdkp.2013.3401.

90 Hahm, Younggyun, Jungyeul Park, Kyungtae Lim, Youngsik Kim, Dosam Hwang, and Key-Sun Choi.

91 2014. "Named Entity Corpus Construction Using Wikipedia and DBpedia Ontology," 5.

92 Hamadou, Abdelmajid Ben, Odile Piton, and Héla Fehri. 2010. "Multilingual Extraction of Functional Relations between Arabic Named Entities Using NooJ Platform," 11.

93 Hamza, Bakkali, Yousfi Abdellah, Gueddah Hicham, and Belkasmi Mostafa. 2014. "For an Independent Spell-Checking System from the Arabic Language Vocabulary." *International Journal of Advanced Computer Science and Applications* 5 (1). https://doi.org/10.14569/IJACSA.2014.050115.

94 Harrag, Fouzi, Eyas El-Qawasmah, and Abdul Malik S. Al-Salman. 2011. "Stemming as a Feature Reduction Technique for Arabic Text Categorization." *Proceedings of the 10th International Symposium on Programming and Systems, ISPS' 2011*, 128–33. https://doi.org/10.1109/ISPS.2011.5898874.

95 Helwe, Chadi, and Shady Elbassuoni. 2019. "Arabic Named Entity Recognition via Deep Co-Learning." *Artificial Intelligence Review* 52 (1): 197–215. https://doi.org/10.1007/s10462-019-

96 09688-6.

97 Hinton, Geoffrey, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, et al. 2012. "Deep Neural Networks for Acoustic Modeling in Speech Recognition." *Ieee Signal Processing Magazine*, no. November: 82–97. https://doi.org/10.1109/MSP.2012.2205597.

98 Huang, Zhiheng, Wei Xu, and Kai Yu. 2015. "Bidirectional LSTM-CRF Models for Sequence

Tagging.” *ArXiv:1508.01991 [Cs]*, August. http://arxiv.org/abs/1508.01991.

99 Hutchison, David, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, et al. 2010. “Comparison of Paraphrase Acquisition Techniques on Sentential Paraphrases.” In *Advances in Natural Language Processing*, edited by Hrafn Loftsson, Eiríkur Rögnvaldsson, and Sigrún Helgadóttir, 6233:67–78. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-14770-8_9.

100 Isabelle, Pierre, Colin Cherry, and George Foster. 2017. “A Challenge Set Approach to Evaluating Machine Translation A Challenge Set Approach to Evaluating Machine Translation.”

101 Jaafar, Amir Hamzah, and Noraini Che Pa. 2017. “HADITH COMMENTARY REPOSITORY : AN ONTOLOGICAL,” no. 167: 191–98.

102 Ju, Meizhi, Makoto Miwa, and Sophia Ananiadou. 2018. “A Neural Layered Model for Nested Named Entity Recognition.” In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume*

103 *1 (Long Papers)*, 1446–59. New Orleans, Louisiana: Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1131.

104 Kalchbrenner, Nal, and Phil Blunsom. 2013. “Recurrent Continuous Translation Models.” *Emnlp*, no.

105 October: 1700–1709. https://doi.org/10.1146/annurev.neuro.26.041002.131047.

106 Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. 2014. “A Convolutional Neural Network for Modelling Sentences.” In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 655–65. Baltimore, Maryland: Association for Computational Linguistics. https://doi.org/10.3115/v1/P14-1062.

107 Khalifa, and Shaalan. 2019. “Character Convolutions for Arabic Named Entity Recognition with Long Short-Term Memory Networks.” *Computer Speech & Language*.

108 Kholy, Ahmed El, and Nizar Habash. 2010. “Orthographic and Morphological Processing for English- Arabic Statistical Machine Translation Mots-Clés : Keywords :,” 19–23.

109 Kholy, Ahmed El, and Nizar Habash. 2010. “Techniques for Arabic Morphological

Detokenization and Orthographic Denormalization," 7.

110 Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. "Statistical Phrase-Based Translation," no. June: 48–54.

111 Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. "ImageNet Classification with Deep Convolutional Neural Networks." *Communications of the ACM* 60 (6): 84–90. https://doi.org/10.1145/3065386.

112 Laatar, Rim, Chafik Aloulou, and Lamia Hadrich Belghuith. 2018. "Word Embedding for Arabic Word Sense Disambiguation to Create a Historical Dictionary for Arabic Language." In *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, 131–35. Amman: IEEE. https://doi.org/10.1109/CSIT.2018.8486159.

113 Lafferty, John, Andrew McCallum, and Fernando C N Pereira. 2001. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," 10.

114 Lebret, Remi, David Grangier, and Michael Auli. 2016. "Neural Text Generation from Structured Data with Application to the Biography Domain." *ArXiv:1603.07771 [Cs]*, March. http://arxiv.org/abs/1603.07771.

115 Lee, Young-Suk. 2004. "Morphological Analysis for Statistical Machine Translation," 5.

116 Liu, Zhuo-Ran, and Yang Liu. 2017a. "Exploiting Unlabeled Data for Neural Grammatical Error Detection." *Journal of Computer Science and Technology* 32 (4): 758–67. https://doi.org/10.1007/s11390-017-1757-4.

117 Lu, Liang, and Steve Renals. 2017. "Small-Footprint Highway Deep Neural Networks for Speech Recognition." *Ieee-Acm Transactions on Audio Speech and Language Processing* 25 (7): 1502– 11. https://doi.org/10.1109/TASLP.2017.2698723.

118 Luong, Minh-thang, and Christopher D. Manning. 2015. "Stanford Neural Machine Translation Systems for Spoken Language Domains." *Iwslt-2015*, 76–79.

119 Luong, Minh-Thang, and Christopher D. Manning. 2016. "Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models."

120 Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. 2015a. "Effective Approaches to Attention-Based Neural Machine Translation." https://doi.org/10.18653/v1/D15-1166.

121 M. ASHAREF, N. OMAR, and M. ALBARED. 2012. “ARABIC NAMED ENTITY RECOGNITION IN CRIME DOCUMENTS.” *Journal of Theoretical and Applied Information Technology*.

122 Mallinson, Jonathan, Rico Sennrich, and Mirella Lapata. 2017. “Paraphrasing Revisited with Neural Machine Translation,” no. 2017.

123 Manning, Christopher, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. “The Stanford CoreNLP Natural Language Processing Toolkit.” In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60. Baltimore, Maryland: Association for Computational Linguistics. https://doi.org/10.3115/v1/P14-5010.

124 Menai, Mohamed El Bachir, and Wojdan Alsaeedan. 2012. “Genetic Algorithm for Arabic Word Sense Disambiguation.” In *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 195–200. Kyoto, Japan: IEEE. https://doi.org/10.1109/SNPD.2012.38.

125 Merhbene, Laroussi, Anis Zouaghi, and Mounir Zrigui. 2013. “A Semi-Supervised Method for Arabic Word Sense Disambiguation Using a Weighted Directed Graph,” 5.

126 Mesfar. 2007. “Named Entity Recognition for Arabic Using Syntactic Grammars.” *International Conference on Application of Natural Language to Information Systems. Springer, Berlin, Heidelberg*.

127 Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. “Distributed Representations of Words and Phrases and Their Compositionality,” 9.

128 Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. “Introduction to WordNet: An On-Line Lexical Database *.” *International Journal of Lexicography* 3 (4): 235–44. https://doi.org/10.1093/ijl/3.4.235.

129 Mouhcine, Rabi, Amrouch Mustapha, and Mahani Zouhir. 2018. “Recognition of Cursive Arabic Handwritten Text Using Embedded Training Based on HMMs.” *Journal of Electrical Systems and Information Technology* 5 (2): 245–51. https://doi.org/10.1016/j.jesit.2017.02.001.

130 Mussa, Sarah Abdul-Ameer, and Sabrina Tiun. 2015. “Word Sense Disambiguation on English

Translation of Holy Quran." *Bulletin of Electrical Engineering and Informatics* 4 (3). https://doi.org/10.11591/eei.v4i3.507.

131 NayerEl-Gedawy, Madeeh. 2013. "Using Fuzzifiers to Solve Word Sense Ambiguation in Arabic Language." *International Journal of Computer Applications* 79 (2): 1–8. https://doi.org/10.5120/13710-1465.

132 Ng, Hwee Tou, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. "The CoNLL-2014 Shared Task on Grammatical Error Correction," 14.

133 Nirenburg, Sergei, Harold L. Somers, and Yorick A. Wilks, eds. 2003. "The Proper Place of Men and Machines in Language Translation." In *Readings in Machine Translation*. The MIT Press. https://doi.org/10.7551/mitpress/5779.003.0022.

134 Och, F. J., Tillmann, C., and Ney, H. 1999. "Improved Alignment Models for Statistical Machine Translation."

135 Oudah, Mai, and Khaled Shaalan. 2012. "A Pipeline Arabic Named Entity Recognition Using a Hybrid Approach," 18.

136 Pasha, Arfath, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M Roth. 2014. "MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic," 8.

137 Ponzetto, Simone Paolo, and Roberto Navigli. 2010. "Knowledge-Rich Word Sense Disambiguation Rivaling Supervised Systems," 10.

138 Rachidi, T, M Bouzoubaa, L Elmortaji, B Boussouab, and A Bensaid. 2012. "Arabic User Search Query Correction and Expansion."

139 Raganato, Alessandro, Claudio Delli Bovi, and Roberto Navigli. 2017. "Neural Sequence Learning Models for Word Sense Disambiguation." In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1156–67. Copenhagen, Denmark: Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1120.

140 Rei, Marek, and Helen Yannakoudakis. 2016. "Compositional Sequence Labeling Models for Error Detection in Learner Writing." https://doi.org/10.18653/v1/P16-1112.

141 Rodr, Horacio, David Farwell, Javi Farreres, Manuel Bertran, M Antonia Mart, William Black, Sabri Elkateb, James Kirk, Piek Vossen, and Christiane Fellbaum. 2008. “Arabic WordNet : Current State and Future Extensions,” 1–21.

142 Saad, Motaz K. 2010. “OSAC : Open Source Arabic Corpora.”

143 Sardinha, Tony Berber. 2011. “Metaphor and Corpus Linguistics” 11 (2): 32.

144 Schmidhuber, Juergen. 2015. “Deep Learning in Neural Networks: An Overview.” *Neural Networks* 61 (January): 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.

145 Schwab, Didier, and Nathan Guillaume. 2011. “A Global Ant Colony Algorithm for Word Sense Disambiguation Based on Semantic Relatedness.” In *Highlights in Practical Applications of Agents and Multiagent Systems*, edited by Javier Bajo Pérez, Juan M. Corchado, María N. Moreno, Vicente Julián, Philippe Mathieu, Joaquin Canada-Bago, Alfonso Ortega, and Antonio Fernández Caballero, 89:257–64. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-19917-2_31.

146 Setiawan, Hendra, Zhongqiang Huang, Jacob Devlin, Thomas Lamar, Rabih Zbib, Richard Schwartz, and John Makhoul. 2015. “Statistical Machine Translation Features with Multitask Tensor Networks.” *ArXiv:1506.00698 [Cs]*, June. http://arxiv.org/abs/1506.00698.

147 Shaalan, Khaled. 2003. “Development of Computer Assisted Language Learning System for Arabic Using Natural Language Processing Techniques,” December, 25.

148 Shaalan, Khaled F. 2005. “Arabic GramCheck: A Grammar Checker for Arabic.” *Software - Practice and Experience* 35 (7): 643–65. https://doi.org/10.1002/spe.653.

149 Shaalan, Khaled F, Marwa Magdy, and Aly Fahmy. 2010. “Morphological Analysis of Ill-Formed Arabic Verbs in Intelligent Language Tutoring Framework.” *The 23rd International Florida Artificial Intelligence Research Society Conference (FLAIRS-23)*, no. Flairs: 277{\textendash}282.

150 Shaalan, Khaled, Ashraf Hendam, and Ahmed Rafea. 2012. “Rapid Development and Deployment of Bi-Directional Expert Systems Using Machine Translation Technology.” *Expert Systems with Applications* 39 (1): 1375–80. https://doi.org/10.1016/j.eswa.2011.08.019.

151 Shaalan, Khaled, Marwa Magdy, and Aly Fahmy. 2015. “Analysis and Feedback of Erroneous

Arabic Verbs." *Natural Language Engineering* 21 (02): 271–323. https://doi.org/10.1017/S1351324913000223.

152 Shaalan, Khaled, and Mai Oudah. 2014. "A Hybrid Approach to Arabic Named Entity Recognition."

153 *Journal of Information Science* 40 (1): 67–87. https://doi.org/10.1177/0165551513502417.

154 Shaalan, Khaled, and Hafsa Raza. 2007. "Person Name Entity Recognition for Arabic." In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages Common Issues and Resources - Semitic '07*, 17. Prague, Czech Republic: Association for Computational Linguistics. https://doi.org/10.3115/1654576.1654581.

155 Shi, Dengliang. 2011. "A Study on Neural Network Language Modeling," no. 2003.

156 Soliman, Abu Bakr, Kareem Eissa, and Samhaa R. El-Beltagy. 2017. "AraVec: A Set of Arabic Word Embedding Models for Use in Arabic NLP." *Procedia Computer Science* 117: 256–65. https://doi.org/10.1016/j.procs.2017.10.117.

157 Sun, Koun-tem, Yueh-min Huang, and Ming-chi Liu. 2011. "A WordNet-Based Near-Synonyms and Similar-Looking Word Learning System" 14: 121–34.

158 Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. "Sequence to Sequence Learning with Neural Networks," 9.

159 Tang, Haiqing, and Deyi Xiong. 2016. "Improving Translation Selection with Supersenses," 10.

160 Tsai, Tzong-Han, Shih-Hung Wu, Cheng-Wei Lee, Cheng-Wei Shih, and Wen-Lian Hsu. 2004. "Mencius: A Chinese Named Entity Recognizer Using the Maximum Entropy-Based Hybrid Model," 18.

161 Utiyama, Masao, and Hitoshi Isahara. 2007. "A Comparison of Pivot Methods for Phrase-Based Statistical Machine Translation," 8.

162 Wang, Peilu, Zhongye Jia, and Hai Zhao. 2014. "Grammatical Error Detection and Correction Using a Single Maximum Entropy Model." *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, no. 13511500200: 74–82.

163 Wen, Tsung-Hsien, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young.

2015. “Semantically Conditioned LSTM-Based Natural Language Generation for Spoken Dialogue Systems,” no. 2005.

164 Wu, Hua, and Haifeng Wang. 2007. “Pivot Language Approach for Phrase-Based Statistical Machine Translation.” *Machine Translation* 21 (3): 165–81. https://doi.org/10.1007/s10590-008-9041-

165 6.

166 Young, Tom, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2016. “Recent Trends in Deep Learning Based Natural Language Processing,” 1–22.

167 Yusuf, Ali. 2000. . “. "The Holy Quran: Translation by Abdullah Yusuf Ali.”

168 Zaghouani, Wajdi, Taha Zerrouki, and Amar Balla. 2015. “SAHSOH$@$QALB-2015 Shared Task: A Rule-Based Correction Method of Common Arabic Native and Non-Native Speakers’ Errors.” In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, 155–60. Beijing, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/W15-3219.

169 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,” 11.

170 Zeyer, Albert, Patrick Doetsch, Paul Voigtlaender, Ralf Schlüter, and Hermann Ney. 2016. “A Comprehensive Study of Deep Bidirectional LSTM RNNs for Acoustic Modeling in Speech Recognition,” 3–7.

171 Zou, Will Y, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. “Bilingual Word Embeddings for Phrase-Based Machine Translation,” 6.

172 Zouaghi, A, L Merhbene, and M Zrigui. 2011. “Word Sense Disambiguation for Arabic Language Using the Variants of the Lesk Algorithm,” 8.

173 Zouaghi, Anis, Laroussi Merhbène, and Mounir Zrigui. 2012. “A Hybrid Approach for Arabic Word Sense Disambiguation.” *International Journal of Computer Processing of Languages* 24 (02): 133–51. https://doi.org/10.1142/S1793840612400090.