Dependency-based Bilingual Word Embeddings and Neural Machine Translation

Taghreed Alqaisi

Doctor of Philosophy

University of York
Computer Science

February 2023

Abstract

Bilingual word embeddings, which represent lexicons from various languages in a common embedding space, are critical for facilitating semantic and knowledge transfers in a wide range of cross-lingual NLP applications. The significance of learning bilingual word embedding representations in many Natural Language Processing (NLP) tasks motivates us to investigate the effect of many factors, including syntactical information, on the learning process for different languages with varying levels of structural complexity. By analysing the components that influence the learning process of bilingual word embeddings (BWEs), this thesis examines some factors for learning bilingual word embeddings effectively. Our findings in this thesis demonstrate that increasing the embedding size for language pairs has a positive impact on the learning process for BWEs. While sentence length depends on the language. Short sentences perform better than long ones in the En-ES experiment. However, by increasing the sentence, En-Ar and En-De experiment achieve improved model accuracy. Arabic segmentation, according to En-Ar experiments, is essential to the learning process for BWEs and can boost model accuracy by up to 10%.

Incorporating dependency features into the learning process enhances the trained models performance and results in more improved BWEs in all language pairs. Finally, we investigated how the dependancy-based pretrained BWEs affected the neural machine translation (NMT) model. The findings indicate that in various MT evaluation matrices, the trained dependancy-based NMT models outperform the baseline NMT model.

Glossary

- (ARPA) Advanced Research Project Agency
- (MSA) Modern Standard Arabic
- (GloVe) Global Vectors
- (CFG) Context-free Grammar
- (NN) Neural Networks
- (NLP) Natural Language Processing
- (MT) Machine Translation
- (CBOW) Continuous Bag-of-Words
- (SG) Skip-Gram
- (RAE) Recursive Autoencoder
- (BRAE) Bilingually-constrained Recursive Autoencoder
- (BCorrRAE) Bilingual Correspondence Recursive Autoencoder
- (BattRAE) Bidimensional Attention-based Recursive Autoencoder
- (BilBOWA) Bilingual Word Embeddings Without Word Alignments
- (BRAVE) Bilingual paRAgraph VEctors
- (RBMT) Rule−based Machine Translation
- (EBMT) Example−based Machine Translation
- (CBMT) Context−based Machine Translation
- (SMT) Statistical Machine Translation
- (PBM) Phrase-based Model

- (FPBMT) Factored Phrase-based Model
- (HPBM) Hierarchical Phrase-Based Model
- (PoS) Part-of-Speech
- (NMT) Neural Machine Translation
- (RNN) Recurrent Neural Networks
- (PER) Position Error Rate
- (WER) Word Error Rate
- (BLEU) Bilingual Evaluation Understudy
- (PSD) Population Standard Deviation
- (VSO) Verb-Subject-Object
- (SVO) Subject-Verb-Object
- (UD) Universal Dependencies
- (MT) Machine Translation
- (TER) Translation Error Rate

Contents

List of Tables

List of Figures

Acknowledgements

First praise is to Allah, the Almighty, on whom ultimately we depend for sustenance and guidance. I would like to thank my supervisor Simon O'Keefe for supporting and motivating me through the whole period of the PhD. It has been a pleasure working with him. I would like to thank my children for their support and being part of this journey. I would never have been able to reach where I am now without them. I thank all my colleagues in the Artificial Intelligence group for providing a friendly environment for sharing ideas and receiving constructive criticism. In particular, I thank Alexandros Komnnios, Nils Moenning, Chaitanya Kaul and Mudita Sharma for all the hours we have been together during our studies in York. Last but not least, I want to thank my parents for their endless motivation and support throughout all the years of my studies.

Taghreed Alqaisi
June 2022

Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of the University of York. This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references.

I hereby give consent for my thesis, if accepted, to be made available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed
Taghreed Alqaisi

Date
15/06/2022

Some of the material contained in this thesis has appeared in the following published paper:

- T. Alqaisi and S. O'Keefe, "En-ar bilingual word embeddings without word alignment: Factors effects," in Proceedings of the Fourth Arabic Natural Language Processing Workshop. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 97–107. [Online]. Available: https://aclanthology.org/W19-4611

- T. Alqaisi, A. Komninos, and S. O'Keefe, "Dependency based bilingual word embeddings without word alignment," in 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–6. [Online]. Available: https://eprints.whiterose.ac.uk/180866/

Chapter 1

Introduction

Semantic representations play an important role in many Natural Language Processing (NLP) tasks: Machine Translation (MT), information extraction and document classifications. Word embeddings learned with Neural Networks (NN) have drawn attention of many researchers due to their superior performance in many downstream tasks compared to traditional count-based distribution models [6]. They are capable of capturing a words' semantic and syntactic information. Driven by these successes, many models have been developed, such as Word2Vec and GloVe.

Many Neural Network-based models have been presented in the context of learning cross-lingual embeddings, and they have proven to be effective at a variety of NLP tasks (Chapter 2), including document classification, named entity recognition, MT, and others. Despite their success, the structural complexity of languages have not been taken into account in many of these areas. In this research, we investigate the effect of many factors including syntactical information on the process of learning bilingual word embeddings for different languages with different levels of structural complexity. Also, we studied the effect of infusing dependency features on learning bilingual word embeddings. We investigated how the pre-trained dependency-based bilingual word embeddings perform on Neural Machine Translation (NMT) tasks.

1.1 Motivation

In this thesis, we use a variety of language pairs that span from comparable to dissimilar language sentence structures: English-Spanish (En-Es), English-German (En-De), and English-Arabic (En-Ar) language pairs. Arabic is still considered a challenging language for automatic translation. In considering English-Arabic language pairs, this thesis aims to address a known gap for Machine Translation. Arabic is widely spoken and highly varied. Further, the structure of the language differs significantly from that of English, Spanish, and German. In this thesis, we apply Modern Standard Arabic (MSA) as it is the most accessible version of the Arabic language. We are motivated to investigate this topic because there is a dearth of study in this field. As a result, the primary research question is: How can we train NN-based bilingual word embeddings that will allow us to enhance Machine Translation for languages of varying complexity?. In this thesis, we investigate factors that affect the learning process of bilingual word embeddings on languages with different morphological complexity. Therefore, we pose the following research questions are:

- Which hyperparameters are most influential in affecting the learning process of BWEs? For example: sentence length, embedding size and morphological segmentation.

- How does incorporating additional features, like syntactic features and POS, into the learning process of bilingual embeddings affect the model performance?

- What is the effect of including these features in training word embeddings on Neural Machine Translation as application?

1.2 Aims and Objectives

Despite the enormous variety of natural languages spoken throughout the world, the majority of works are focused on language pairs such as English-European or English-Chinese languages. Within Machine Translation, the main purpose is to investigate the effect of different factors on the performance of learning bilingual word embeddings for different language complexity pairs. This investigation can improve the process of learning bilingual word embeddings. Learning better bilingual word embedding results in minimising the distance (increasing semantic similarity) between sentences from different languages. When doing this, it is important to consider language complexity differentiation, as translating from a simple language into a more complex one is still a challenging task, even for Statistical Machine Translation (SMT). In order to achieve this goal, we investigate the effect of several factors on training performance of the bilingual embeddings model. We specifically consider: the size of the embeddings, the length of the sentences, and the process of word segmentation (for Arabic). In addition, we examine the integration of syntax features into the learning process.

To answer the research questions formulated in the previous section, the project aim and objectives as follows:

- Identify the effect of factors such as sentence length, embedding size, and Arabic pre-processing settings (namely, segmentation schemes) on the learning of bilingual word embeddings models.
- At the monolingual level, a significant amount of effort has been made on learning of syntax-based distributed representations of individual words. However, there is a lack of research on syntax-based multilingual representations. Consequently, we aim to improve the learning of bilingual embeddings for diverse language pairs. This will be done through the following:
  - Including a variety of knowledge-based techniques. For example: Learning dependency-based bilingual word embeddings. Research shows that utilising a parsed datasets help the learning process for monolingual word embeddings [7]. Thus, we aim to extend the work of [7] in order to investigate these features effect on learning bilingual word embeddings.
- Investigating the effect of using dependency-based pretrained BWEs on NMT task.

1.3 Contributions

This thesis contributes to a better understanding of MT translation tasks and, as a result, leads to learning better word representation models. To our knowledge, this thesis is one of the first to investigate the effectiveness of learning models in dealing with these language translation challenges. Therefore, this thesis makes the following contributions:

- A word-word dictionary for En-Es, En-De and En-Ar. This dataset has been created to be used in Chapters 4 and 5 and evaluated on Cross Language Dictionary Induction (CLDI) task. We have created a test dataset for all used language pairs: Es-En, De-En and Ar-En as has been explained in Chapter 4.
- Trained Bilingual Word Embeddings (BWEs) with different hyper-parameters: namely sentence length and embedding size for En-Es, En-De and En-Ar language pairs [1].
- Trained BWEs with different segmentation schemes for En-Ar language pair.
- Training different dependency-based bilingual word embeddings for En-Es, En-De and En-Ar language pairs that will inform further research [2].

[1] https://github.com/totyqa/Dependency-Based-BWEs/blob/main/PRETRAINED-BWEs

[2] https://github.com/totyqa/Dependency-Based-BWEs/blob/main/PRETRAINED-BWEs

- NMT systems that have been trained on the pre-trained dependency-based BWEs are available upon request [3].

1.4 Chapter Overview

The remainder of the thesis is organised as follows. Chapter 2 provides an overview of word embeddings from word-based to phrase-based. Also it presents Bilingual word embedding methods in more detail. In addition, Word Embeddings for Machine Translation are presented in more details. Next, we present the history of machine translation and how it has developed over time, including demonstrating previous work on Statistical Machine Translation (SMT) and current approaches and methods. We explain the existing evaluation methods for Machine Translation. Additionally, Arabic language literature is presented.

Chapter 3 is the first experimental chapter, which investigates the effect of factors including: sentence length, embedding size for English, Spanish, German and Arabic languages in addition to word segmentation for Arabic language. It presents the related work in this research area. We explain the used data sets, trained model BilBOWA and evaluation method in greater detail.

Chapter 4 improves BWEs by incorporating syntax features on the learning process. The BilBOWA model has been trained using different language pairs that vary in sentence structures. Evaluation shows that the model performs better than the baseline.

Chapter 5 presents NMT on knowledge bases. The NMT systems have been trained using the pretrained syntax-based bilingual word embeddings from Chapter 5. The results show that using dependency features improves the MT quality comparing to the baseline.

Chapter 6 presents a summary of the finding of the thesis and future work.

[3] https://github.com/totyqa/Dependency-Based-BWEs/blob/main/PRETRAINED-BWEs

Chapter 2

Background

This chapter present a background on the area of research in general as well as will discuss approaches and methods of related research work. Firstly, we give an overview on history of translation systems and review the challenges that are specific to the Arabic language. We then present existing methods and discuss their limitations. Also, word embedding methods will be reviewed at monolingual and bilingual levels.

2.1 Word Embeddings

An increasing number of researchers are studying neural networks, and they have produced highly promising findings in a variety of applications of Natural Language Processing (NLP). In recent years, a range of models, such as semantics and question answering models [8, 9, 10], Machine Translation [11, 12], parsing [13, 14], have been introduced. This chapter introduces a background on word embeddings. Word embeddings are one of the most important NLP applications since they are capable of capturing semantic similarities between words.

Learning word embeddings derives from the principle that words can be transformed from a discrete space of features to a continuous vector space of features that captures their syntactic and semantic information. In other words, words with similar meanings are more likely to have vectors that are comparable. This similarity can be measured using different distance methods such as cosine similarity and euclidean distance. Many neural-based word embeddings models have been introduced recently, and they show a significant improvement in a variety of NLP applications, including language modelling [15, 16, 17], machine translation [18, 12, 19], named entity recognition [20], document classification and sentiment analysis [21, 22, 23].

Word embeddings can be classified, based on the objective function that needs to be learnt, into two main categories: Monolingual word embeddings - which is the process of learning similar word representations for similar word meaning within the same language, and Bilingual/cross-lingual approaches - which is the process of learning similar words between languages.

- Monolingual approach

  Recently, various monolingual word embedding models have been introduced which have demonstrated great performance across a range of natural language processing applications. The majority utilise Word2Vec, which is a log-linear, continuous Bag-of-Words (CBOW) or skip-gram (SG) model with negative sampling (Section 2.1.1), that has been introduced by [24]. Word2Vec models show a very efficient performance in terms of presenting words in continuous space and capturing their semantic information. A large number of works are introduced that are based on Word2Vec with various modifications to it, for example: GloVe model by[25] and the works of [26]. Ling et al. [26] have introduced the Wang2Vec model, which utilises structural skip-grams and a continuous window that adapted to be more sensitive to word positioning. The Wang2Vec model outperforms the Word2Vec model in terms of performance and generalisation ability in noisy conditions and tasks involving syntax. Therefore Wang2vec is effective at modelling intricate semantic and syntactic word relationships.

  Moreover, Trask et al. [27] introduce Sense2Vec model that is considered as a state-of-the-art multi-sense embedding model due to its ability in capturing more nuanced senses. In addition, dependency-based word embedding were introduced by [28] and [7]. Omer and Yoav [28] produce dependency-base word embedding models, which include arbitrary contexts with a skip-gram model. Komninos and Manandhar [7] extended these works in [28], by considering co-occurrences in a dependency graph between word and dependency context features. At phrase-level, Socher et al. [29] introduces a recursive auto-encoder (RAE) model, which is a phrase-based word embedding model that learn phrase representations as explained in more details later. Despite the success that these models have achieved, word ambiguity continues to be a challenge in natural language processing since it manifests itself at all levels of language, particularly at the phrase and sentence level.

- Bilingual/Cross-lingual approach:

  Bilingual or cross-lingual word embedding is the process of learning word embeddings in two or more languages using two or more corpora. The majority of bilingual/cross-lingual word embedding models are simply extensions of monolingual word embedding models, with some exceptions. Many successful models for learning bilingual word embeddings have been developed, each of which makes use of a different corpus with a different level of alignment. Firstly, at word-level alignment, Luong et al. [30] extend the skip-gram model to learn efficient bilingual word embeddings. Also, at phrase-level, a bilingually-constrained phrase embeddings (BRAE) model learns source-target phrase embeddings by minimising the semantic distance between translation equivalents and maximising the semantic distance between non-translation equivalents [31]. Su et al. [32] extend BRAE model by introducing a bilingual correspondence recursive auto-encoder (BCorrRAE) model, which incorporates word alignment to learn bilingual phrase embeddings by capturing different levels of their semantic relations. Zhang et al. [3] introduce a bi-dimensional attention-based recursive auto-encoder (BattRAE) model to learn bilingual phrase embeddings by integrating source-target interactions at different levels of granularity using attention-based models. Using sentence-aligned corpus, Gouws et al. [33] and Coulmance et al. [34] introduce BilBOWA and Trans-gram methods to learn and align word embeddings without word alignment. At document level aligned corpus, Vulic and Moens [35] present a model that learns bilingual word embeddings from non-parallel document-aligned data without using translation pairs. In addition, Mogadala and Rettinger [36] introduce a Bilingual paRAgraph VEctors (BRAVE) model that learns bilingual embeddings from either a sentence-aligned parallel corpus or a label-aligned non-parallel document corpus. Vulic and Moens [35] introduce a model that learns multilingual (two or more languages) words embeddings using document-aligned comparable data.

2.1.1 Methods

NN-based word embedding methods improve many NLP tasks and outperform the traditional word representation in many aspects by saving time and memory as well as producing powerful word representations. Many word embedding methods have been introduced: language models, Continuous Bag of Word (CBOW), and skip-gram models. As these models are NN-based models, they learn continuous word vectors using back-propagation and stochastic gradient descent algorithms which we explain further in next sections.

Skip-Grams

Skip-gram was introduced by [24]. In this model, for a given sentence, the model uses word w_t to predict the context words (words before and after word w_t). In other words, the input is a single word w_t and the output is a set of context words $\{w_{t-C}, \ldots, w_{t-2}, w_{t-1}, w_{t+1}, w_t{+}2, \ldots, w_{t+C}\}$ (See Figure 2.1). C is the word window

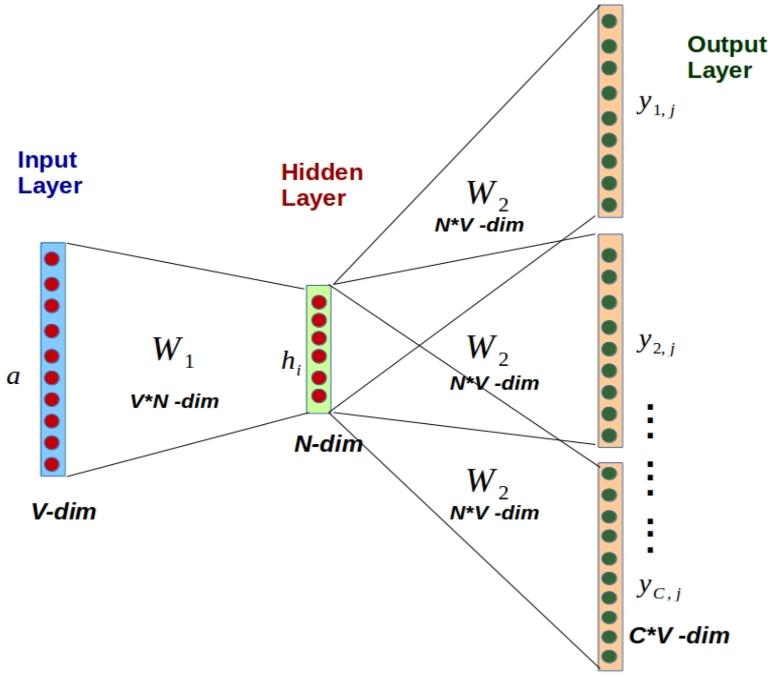


Figure 2.1: The skip-gram model

size that defines the number of skipped words before and after the input word. For example, in the sentence (*The black fox jumps*), considering the word (*fox*) as an input word and C=2. Then the output words (context words) are [*the*, *black*, *jumps*] and if C=1, the context words are [*black*, *jumps*]. To feed these data into the model, all words (input and output) need to be encoded to one-hot vectors that have the value of 1 at the index corresponding to the word in the vocabulary and *zeros* on other indexes. As a NN-based model, the learning process can be divided into two main phases: feed-forward propagation and back-propagation.

- Feed-forward Propagation:

  As shown in (Figure 2.1), the NN consists of input layer, one hidden layer , output layer and two weights matrices W_1 and W_2. The goal of the model building process is to learn these weights (words' vectors). Here, W_1 produces word vectors for the target words and W_2 produces the context word's vectors.

  In this phase, the input vector a, which is a one-hot vector, is multiplied by W_1 as:

  $$h = a^T.W_1 \tag{2.1}$$

  as a has only one value of 1 in its correspondence index and the others are *zeros*, h will be a vector of a row of W_1 that corresponds to the index of a. There is no activation function at this layer. So, h is the input to the next layer.

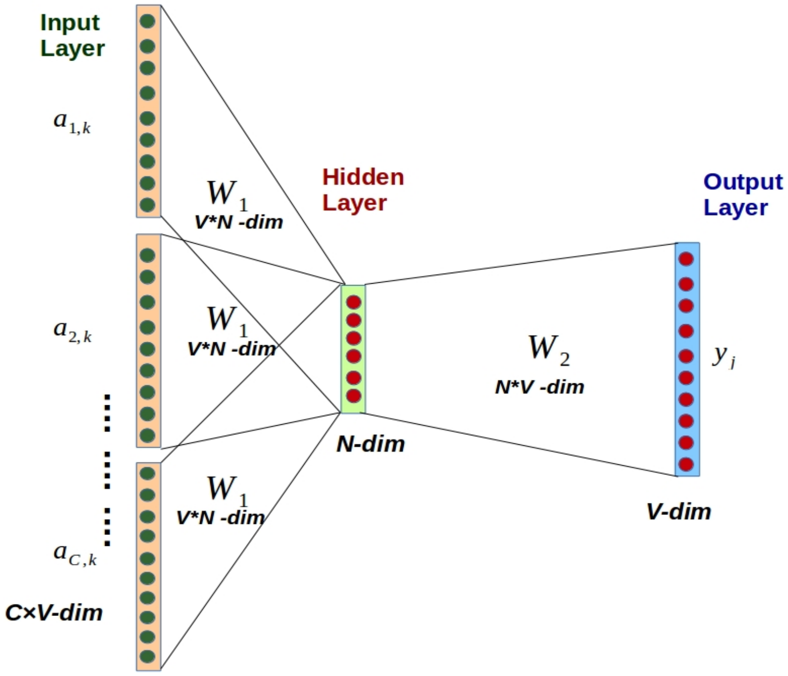


Figure 2.2: Continuous Bag-of-Word

After that, from the hidden layer to the output layer, h is multiplied by W_2 for each context word:

$$u = h.{W_2}^T \tag{2.2}$$

Finally, softmax activation function, which is a combination of multiple sigmoid functions, is applied to the output layer to produce the multinomial distribution. In another words, it computes the probability distribution of a word occurrence with a given size context window [37]. Softmax is defined as:

$$y = p(w_{c,j} = w|w) = \frac{\exp(u_j)}{\sum \exp u'_j} \tag{2.3}$$

- Back-propagation:
  After producing the softmax's output, the back-propagation algorithm is applied to learn both W_1 and W_2 that minimises the loss function.

Continuous Bag-of-Words

Continuous Bag-of-Words (CBOW) is the opposite of the skip-gram (SG) model, as it is a language model that learns word embeddings by predicting word w_t from given context words (previous and followed words) as inputs (Figure 2.2).

Optimisation Techniques

The computation of both the SG and CBOW models is extremely time-consuming because it is performed across the whole vocabulary. As a result, several strategies are employed by researchers in order to make them more efficient and faster. These techniques are: Hierarchical softmax and Negative Sampling. Hierarchical Softmax is a model that representing all words in the vocabulary using a binary tree[38].

Negative sampling is a process of modifying the optimisation objective of the skip-gram model to make it faster and improve its performance. It aims to update only a small percentage of the weights of the training sample instead of updating them across the whole vocabulary. The idea behind this technique is to randomly sample a few words as negative examples. Depending on the amount of training data available, the number of negative samples can range from 5 to 20 words (with more data, fewer negative samples are needed).

Using negative sampling modifies the loss objective:

$$E = -\log \sigma((-w_{2w_o})^T.h) - \sum_{j=1}^{k} \log \sigma((-w_{2_{w_j}})^T.h) \tag{2.4}$$

where w_{2wo} is the positive sample word vector, $h = v$, and w_j is a word vector negative sample. Therefore, the derivative of E with respect to u changes to:

$$\frac{dE}{du_{c,j}} = y_{c,j} - t_{c,j} \tag{2.5}$$

where $t = 1$ in positive samples and $t = 0$ in negative samples.

GloVe Embeddings

Global vectors (GloVe) were introduced by [25]. This model learns the word embeddings via matrix factorisation by minimising the difference between the dot product of the embeddings of a word x_{w_i} and its context. GloVe is a new global log-bilinear regression model for unsupervised word representation learning that outperforms previous models on tasks such as word analogies, word similarity, and named entity recognition. The training is based on a corpus's aggregated global word-word co-occurrence information. Glove has the advantage of incorporating both global and local information into its vectors, whereas Skip-gram and CBOW techniques place a greater emphasis on local information [39]. However, the main drawback of the Glove model is how time-consuming it is.

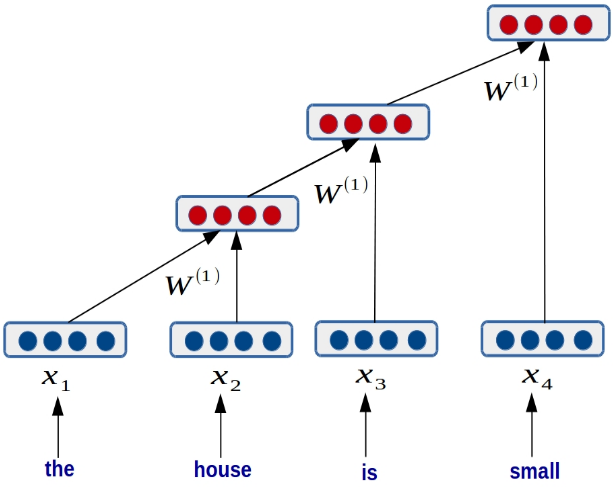


Figure 2.3: RAE: compute the parent node

Recursive Autoencoder

Recursive Auto-encoders (RAE) were introduced in [29]. RAEs capture the meaning of phrases by computes the parent vector y for two children as shown in Figure 2.3. Then it reconstructs the original children nodes to measure how well y presents its children (See Figure 2.4).

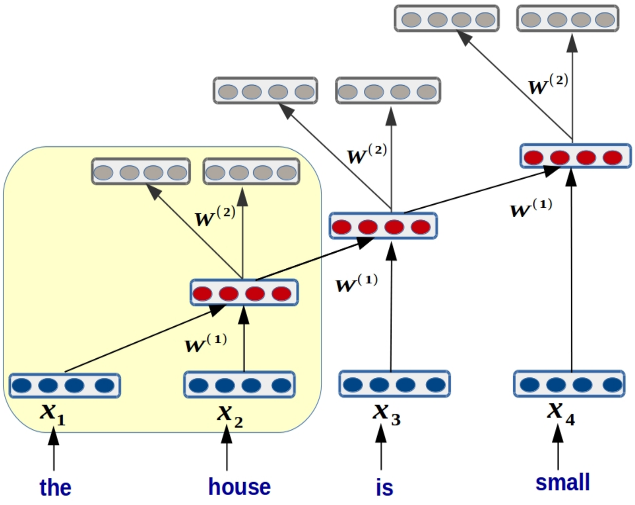


Figure 2.4: RAE: reconstruct the children nodes

2.2 Bilingual Word Embeddings

2.2.1 Methods

The process of learning the semantic similarity across two languages creates bilingual word embeddings. Many learning methods have been introduced using different learning algorithms. These algorithms differ in many aspects:model architecture, the data-sets used, and learning algorithms etc.,. In terms of the bilingual learning step, research presents three different bilingual embedding approaches: monolingual mapping, parallel corpus and joint optimisation approaches. Each methos is presented below and for comprision see Table 2.1.

Monolingual Mapping

In this method, monolingual word representations are learnt separately for each language using a large monolingual corpus. Then, using word translation pairs, the model learns a transformation matrix that maps word representation from one language to another. The main limitation of this method is that the performance of mapping methods is dependent on the language pair, the comparability of the training corpora, and the word embedding algorithm parameters [40]. Also it is essential that embedding spaces in many languages have a similar structure for mapping method to work.

Table 2.1: Bilingual word embeddings methods' advantages vs disadvantages

	Monolingual Mapping	**Parallel**	**Joint learning**
Advantages	Very fast	Use an efficient noise-contrasting training	Train on any available monolingual data
Disadvantages	Ignore the multi-sense polysemy	Train on limited parallel data	Slow in training

Parallel Corpora and Cross-Lingual Training

This approach tries to optimise the cross lingual objective and can be categorised into two main methods: bilingual lexicon and sentence alignments methods. The bilingual lexicon method requires word-level alignments and aims to ensure the translated pairs of words have the same vector representation [41, 33]. In contrast, the sentence aligned methods aim to minimise the distance between two sentence representations [42, 43]. Many models have been introduced in the literature using both approaches. We will discuss these models in later chapters.

Joint Learning

In this approach, the monolingual and cross-lingual objectives are optimised jointly. This type of training is also often referred to as Joint Optimisation. Gouws et al. [33] propose a bilingual bag-of-words without word alignment model that uses the skip-gram model as a monolingual objective and jointly learns the bilingual embeddings by minimising the distance between aligned sentences and through assuming that each word in the source sentence is aligned to all words in the target sentence. Coulmance et al. [34] introduce a bilingual skip-gram without word alignment model, which assumes each source word is aligned to every word in the target sentence. This model implements a skip-gram model in both monolingual and bilingual objectives.

2.2.2 Alignment-Level

Bilingual word embeddings methods can be categorised in terms of the alignment-level of the used parallel data: word, phrase, sentence, document and non-document alignment. This section highlights the relevant literature in these areas.

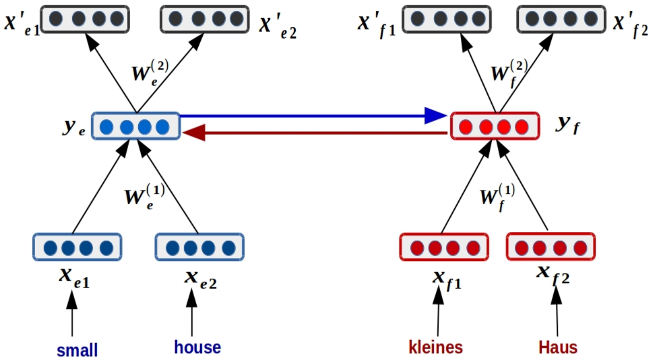


Figure 2.5: Bilingual-constrained phrase embeddings

Word Level

Zou et al. [44] use MT alignments to learn bilingual word embeddings for unlabelled data. Their model shows a good performance in different NLP tasks such as: semantic similarity, phrase-based MT and named entity recognition. Moreover, Luong et al. [30] introduce an effective joint model that learns a high quality bilingual representation by extending the skip-gram model.

Phrase Level

As mentioned above, research introduces many RAE-based Bilingual phrase embeddings models:

- Bilingual-Constrained Recursive Auto-encoder (BREA):
  Zhang et al. [31] present BRAE, which learns the bilingual phrase-based embeddings by, firstly, learning two RAEs (Recursive Auto-encoders) for source and target languages. Secondly, it fine-tunes the phrase embeddings to capture different levels of semantic relations within the bilingual phrases. It does so by minimizing the euclidean distance between translation equivalents and maximizing the euclidean distance between non-equivalents at different levels of granularity (words, sub-phrases or phrases) (See Figure 2.5). Through learning and employing these semantic representations, it leads to significant improvements in machine translation performance.

- Bilingual Correspondence RAE (BCorrRAE):
  BCorrRAE [32] is similar to BRAE in terms of learning two RAEs and ac-

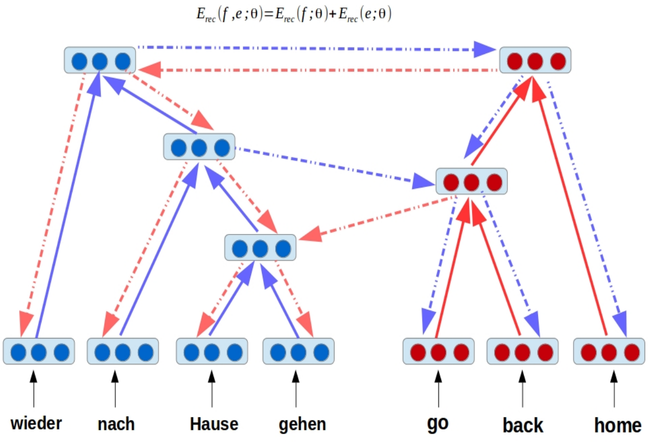


Figure 2.6: BCorrRAE

cessing bilingual constraints at different levels (phrase, sub-phrase, word) thus incorporating word alignments into their model. BCorrRAE minimises the joint objective on the combination of a RAE reconstruction error, structural alignment consistency error and cross-lingual reconstruction error by using the max-semantic-margin error, which is used to minimise the semantic difference between translation equivalents and maximise the semantic distance between non-translation pairings (See Figure 2.6).

- Bi-dimensional attention-based RAE (BattRAE):
  Similar to BCorrRAE, BattRAE [3] employs two RAEs, one for source and one for target, to generate embeddings using tree structures of a phrase at different levels (words, sub-phrases, and phrase). The main difference in this model is that it introduces a bi-dimensional attention network to learn interactions. So, after learning the two RAEs (for source and target), [3] use a bi-dimensional attention network to project embeddings into a common attention space as shown in Figure 2.7.

2.3 Embeddings Used in Machine Translation

The introduced phrase models capture different levels of semantic relations within bilingual phrases by minimizing the Euclidean distance between translation equivalents and maximizing the euclidean distance between non-equivalents at different levels of granularity (words, sub-phrases or phrases). Learning and employing these semantic representations leads to significant improvements in machine translation

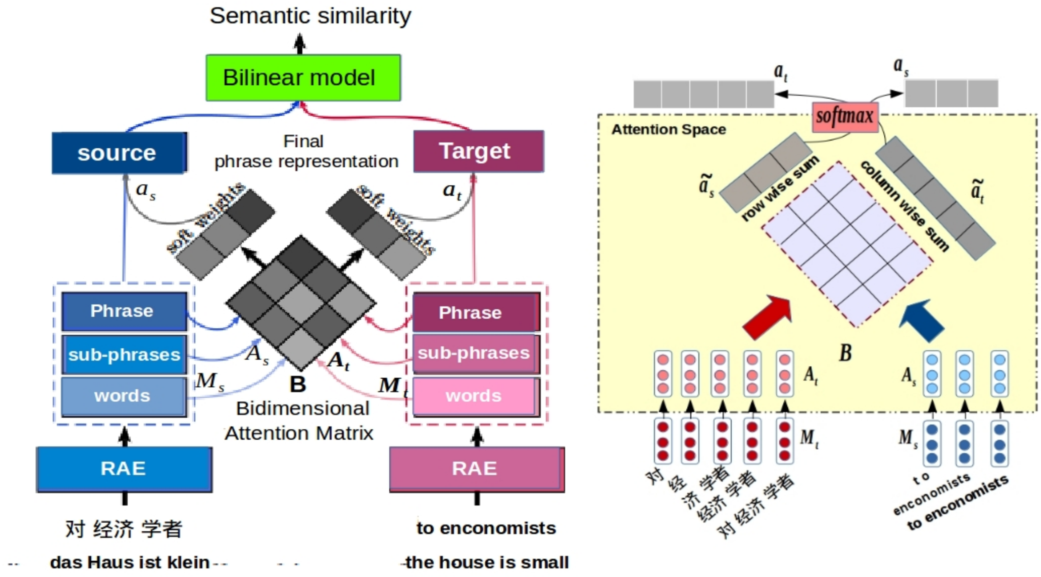


Figure 2.7: BattRAE Model (on the left the whole model and on right the attention computation process) [3]

performance.

2.3.1 Bilingually-constrained Recursive Autoencoder (BRAE)

BRAE is trained to minimise the semantic distance of translation equivalents while concurrently maximising the semantic distance of non-translation pairings. The model learns how to semantically embed each phrase in two languages after training, as well as how to change semantic embedding space from one language to the other. The model learns two RAEs jointly (Recursive Auto-Encoders): one for source language and the other for target language. Two types of errors are involved for phrase pairs (f, e). Firstly, 'reconstruction error' to show how well the learned vectors represent their phrases f and e.

$$E_{rec}\left(f,e;\theta\right) = E_{rec}\left(f;\theta\right) + E_{rec}\left(e;\theta\right) \tag{2.6}$$

Secondly, 'semantic error':

$$E_{sem}\left(f,e;\theta\right) = E_{sem}\left(f|e,\theta\right) + E_{sem}\left(e|f,\theta\right) \tag{2.7}$$

where

$$E_{sem}\left(f|e,\theta\right) = 1/2\|p_e - f\left(W^{(3)}p_f + b^{(3)}\right)\|^2 \tag{2.8}$$

Compute the joint error with

$$E\left(f, e; \theta\right) = \alpha E_{rec}\left(f, e; \theta\right) + \left(1 - \alpha\right) E_{sem}\left(f, e; \theta\right) \tag{2.9}$$

and the objective function

$$E_{sem}\left(f|e, \theta\right) = 1/N \sum_{s,tS,T} E\left(s, t; \theta\right) + \lambda/2\|\theta\|^2 \tag{2.10}$$

2.3.2 Bilingual Correspondence Recursive Autoencoder (BCorrRAE)

BCorrRAE [32], incorporating word alignments into their model to access bilingual constraints at different levels (phrase, sub-phrase, word).

Minimises a joint objective on the combination of a RAE reconstruction error, structural alignment consistency error and a cross-lingual reconstruction error using the max-semantic-margin error. Learning two RAEs for source and target allows the model to computes RAE reconstruction error and the consistency error as below:

$$E_{con}\left(f, e; \theta\right) = E_{con}\left(T_f; \theta\right) + E_{con}\left(T_e; \theta\right) \tag{2.11}$$

Then to assess whether n_f is a structural alignment consistent (SAC) node or not. the consistency inconsistency score is computed for each n_f

$$s\left(n_{\bar{f}}\right) = W^{score} p_{n_{\bar{f}}} \tag{2.12}$$

$$s\left(T_f\right) = \sum s\left(n_{\bar{f}}\right) \tag{2.13}$$

$$E_{con}\left(T_f; \theta\right) = s\left(T_f\right)_{cns} + s\left(T_f\right)_{ins} \tag{2.14}$$

Cross-lingual reconstruction error:

$$E_{clre}\left(f, e; \theta\right) = E_{(f2e.rec)}\left(T_f, T_e; \theta\right) + E_{(e2f.rec)}\left(T_e, T_f; \theta\right) \tag{2.15}$$

where

$$E_{(f2e.rec)}\left(T_f, T_e; \theta\right) = 1/2 \sum_{\langle n_{\bar{f}}, n_{\bar{e}}\rangle \in S} \sum_{n \in T'_e} \|p_n - p'_n\|^2 \tag{2.16}$$

Compute the final objective functions.

2.3.3 Bi-dimensional Attention-based Recursive Autoencoder

Similar to BCorrRAE, Bidimensional Attention-based Recursive Autoencoder (BattRAE) employ two RAEs for source and target in order to generate embeddings and tree structures of a phrase at different levels (words, sub-phrases, phrase). Then they introduce a bi-dimensional attention network to learn their interactions.

Form matrix Ms and Mt from extracted word embeddings. Project embeddings into a common attention space:

$$A_s = f\left(W^{(3)} M_s + b_{[:]}{}^{A}\right) \tag{2.17}$$

$$A_t = f\left(W^{(4)} M_t + b_{[:]}^{A}\right) \tag{2.18}$$

Compute semantic matching score:

$$B_{i},_{j} = g\left(A_{s,i}^{T} A_{t,j}\right) \tag{2.19}$$

Compute the matching score vectors:

$$\tilde{a}_{s},_{i} = \sum_{j} B_{i,j} \tag{2.20}$$

$$\tilde{a}_{s},_{j} = \sum_{i} B_{i,j} \tag{2.21}$$

Apply softmax on matching score vectors to keep their value at the same magnitude.

$$a_s = softmax\left(\tilde{a}_s\right) \tag{2.22}$$

$$a_t = softmax\left(\tilde{a}_t\right) \tag{2.23}$$

Compute the final phrase representation:

$$p_s = \sum a_{s,i} M_{s,i} \tag{2.24}$$

$$p_t = \sum a_{t,i} M_{t,i} \tag{2.25}$$

Semantic similarity Transform P_s and P_t into a common semantic space:

$$s_s = f\left(W^{(5)} p_s + b^{s}\right) \tag{2.26}$$

$$s_t = f\left(W^{(6)} p_t + b^{s}\right) \tag{2.27}$$

Compute semantic similarity score:

$$s\left(f, e\right) = s^{T} S s_t \tag{2.28}$$

Objective function Two errors involved Reconstruction error

Semantic error function:

$$J(\theta) = 1/N \sum_{j=1} N\alpha E_{rec}(f_j, e_j) + \beta E_{sem}(f_j, e_j) + R(\theta) \tag{2.29}$$

To conclude, the three bilingual RAEs based models: BRAE, BcorrRAE and BattRAE, trained to minimise the distance between translation equivalents and maximise the distance between nonequivalents pairs. However, BcorrRAE model enhances the learning process by incorporating word alignment at different levels (word, sub-phrase and phrase). In contrast, BattREA model learns the interaction between these RAEs by incorporating a bilingual attention network.

2.4 Machine Translation Systems

Rapid development of technologies in the field of Machine Translation have enabled application across many areas. Machine Translation systems have been implemented across different organisations, businesses, governments and industry and are involved in a multitude of tasks such as learning, entertainment, security, multimedia, and many more. In multimedia, machine translation have been applied to TV programmes including the news, movies, and live TV broadcasts to translate the spoken language into the written form of another language. However, there are still many aspects which present challenges to MT in terms of matching human translation capabilities.

After inventing electronic computers in the 1950s, many researchers have since been involved in developing automatic Machine Translation systems. They define MT as the use of a computer to translate text from one natural language into another [45] [46].

Machine Translation started with two categories: direct translation, as word-to-word translation, and indirect translation [47][48][49]. The first MT system was installed in 1959 and throughout the 1960s many MT systems were introduced, however, their performance (in terms of results) was quite poor. In 1970, a Russian-English MT system was installed in the US Air Force [50] and, in the 1980s, Example-Based translation systems were built in Japan as well as Statistical Machine Translation (SMT) introduced by IBM's labs in the 1990s [1]. Example-Based MT and Statistical MT are Corpus-Based MT in addition to context-base MT. Since then, statistical machine translation systems have drawn the attention of many researchers. Recently Neural-based MT have been introduced and are showing very promising results.

2.4.1 Approaches

Many MT models have been introduced in the last few decades; Rule-Based, Example-Based, Transfer-Based, Statistical MT, to name but a few. Each of these models classifies into one of two main Machine Translation approaches; Rule-Based Machine Translation and Corpus-Based Machine Translation according to their core methodology. Some models combine more than one approach such as hybird machine translation or use more than one model in the same approach, such as in Hierarchical Phrase-Based Models.

- Rule-Based Machine Translation:
  In the early 1970s, the first Rule-Based Machine Translation (RBMT) system was developed. The basic idea of RBMT is that it relys on linguistic information in both source and target languages. Consequently, RBMT is also known as knowledge-Based Machine Translation. There are three types of RBMT model: Direct Approach, Transfer-Based Approach, and Inter-lingual RBMT.
  - Direct Approach, which is considered as the oldest approach, is a word level translation [51].
  - Transfer-Based Approach can use knowledge of both the source and target languages. This method involves three steps: analysing the source language text to establish its grammatical structure, transferring the resulting structure to a structure for generating text in the target language, and finally generating this text [52].
  - Interlingual RBMT, which is transforming the input sentence into abstract representation and mapping it to the final output [53].
- Corpus-Based Machine Translation:
  Over the last three decades, corpus-based machine translation methods have become one of the most widely explored areas in machine translation. This is due to the use of parallel corpora in machine translation which has enabled a high level of accuracy to be achieved and has improved the translation performance. Therefore, many corpus-based approaches have been introduced; example-based, context-based, and statistical-based machine translation.
  - Example-Based Machine Translation:
    Example-based Machine Translation (EBMT) was suggested by [54] and is based on the idea of translating by analogy. It is, therefore, also known as the analogy-based, and memory-based approach. EBMT is defined as the process of matching an input sentence with already translated examples of a corpus or database in order to extract suitable examples.

  The extracted examples are then recombined in an analogical manner to determine the correct translation [55].

  - Context-based Machine Translation:
    Context-based Machine Translation (CBMT) is another type of corpus-based MT model. However, this model requires no parallel corpora. Instead, it utilises an extensive monolingual target text corpus and a full-form bilingual dictionary as core requirements, with a smaller monolingual source-text corpus as an optional requirement (which enables further improvement of the translation performance) [56, 57].

  - Statistical Machine Translation:
    In 1949 Warren Weaver introduced Statistical Machine Translation (SMT) [58]. Since then, numerous examples of SMT have been developed which we will explore in more detail later.

- Hybrid-Based Machine Translation:
  Hybrid-Based MT is defined as the use of multiple machine translation approaches within a single machine translation system. Many works have been conducted using this combination of two or more machine translation approaches. The most popular combinations fall into the following three approaches; rule-based, example-based, and statistical MT. Examples of the hybrid approach include [59] and [60] who apply the rule-base hybrid approach, and [61] who uses the rule/statistical-based hybrid approach.

To conclude, in the field of MT, numerous studies have been published, and different approaches have been developed. The early approach, the Rule-based MT method, is very expensive, time-consuming, and labor-intensive because it depends on significant linguistic resources and requires linguistic knowledge. SMT approach is much faster than the rule-based approach. However, it is costly to create the compositions and does not perform well for languages with various sentence structures.

Due to the success of SMT approach, the next section reviews this approach in more details.

2.4.2 Statistical Machine Translation

Warren Weaver's SMT approach, devised in the late 1940s, [58] has now been widely applied and developed within the field. Firstly, IBM's Candide Project developed the the word-based model in the 1980s [62]. Following this, extensive research was conducted to improve the quality of SMT systems. This led to numerous Phrase-Based models being introduced which demonstrate better performance than word-based models. The main weakness in word-based models was that one word in the source language (S) can be translated into many words in the target (T) language and vice versa [1, 63]. As such, many researchers agree that the main problem is finding the best translation in terms of fluency and adequacy [64, 1]. This has resulted in a wealth of research in this area.

Word Models

The IBM Candide Project[62] developed the first word-based model in the late 1980s (known as IBM1). This was a simple machine translation model based on lexical translation. The main idea of this model was to map words from the source language to the target language using a dictionary. The IBM1 model is defined as the process of generating a number of different translations for a given sentence using lexical translation probabilities and the notion of alignment [65], [1] and [62].

Numerous word-based models have subsequently been introduce by IBM, namely; IBM2, IBM3, IBM4, and IBM5 [1]. Each of these models has improved the MT performance. Their importance is relay on the alignment and probability distribution [1].

The main disadvantage in the word-based model is that the textual information is not taken into account and the lexicon probabilities are based on single words only. Therefore, the language model is not capable of solving ambiguity as translation depends on the surrounding words. As a result, this model is very weak at solving re-ordering problems [66].

Phrase Models

The Phrase-Based Model (PBM) is an example of a noisy-channel approach (which was introduced for translating French into English in 1993). Examples of PBM that have been introduced include; [67], [66] and [63]. All these model have the same basic PBM architecture; phrase segmentation (or generation), phrase reordering, and phrase translation, which is defined as:

$$\arg\max_{e} P\left(e|f\right) = \arg\max_{e} P\left(e,f\right) = \arg\max_{e} \left(P\left(e\right) \times P\left(f|e\right)\right) \tag{2.30}$$

where $P(f|e)$ is the translation model that encodes e into f by: segmenting e, which is a sentence of target language into phrases $\overline{e}_1 \ldots \overline{e}_I$. Then reordering $\overline{e}_i$ based on distortion model and finally translating each of the $\overline{e}_i$ into the target language using the estimation of $P(\overline{f}|\overline{e})$ from the training data [68].

Many researchers agree that phrases of more than three words long increase the performance of the translation for training a corpus of up to 20 million words

[67],[63], [66]. Nowadays, the most used PBMT models are; Koehn's Phrase-Based Model, Factored Phrase-Based Model, and Hierarchical Phrase-Based Model.

- Koehn's Phrase-Based Model:

  In this research, [1] is the main reference for MT system. Koehn [1] has pointed out that their phrase-based model has obtained the highest levels of performance through the heuristic learning of phrase translation from word-based alignments and through the lexical weighting of phrase translations to find the best translation [63]. They apply the Bayes rule to invert the translation direction and integrate a pre-trained language model (PLM). The best target (e) translation for a source (f) is defined as:

  $$e_{best} = \arg\max_{e} p\left(e|f\right) = \arg\max_{e} p(f|e)PLM(e) \tag{2.31}$$

  Despite defining the best target translation exactly as the same reformulation in word-based models, Koehn et al. have decomposed $P(f|e)$ into:

  $$p\left(\bar{f}_1^I|\bar{e}_1^I\right) = \Pi_{i=1}^{I}\phi\left(\bar{f}_i|\bar{e}_i\right) d\left(start_i - end_{i-1} - 1\right) \tag{2.32}$$

  where $\phi\left(f_i|e_i\right)$ is phrase translation probability, d is a reordering model and $(start_i - end_i - 1)$ is a reordering distance. As such, they break up the source sentence **f** into I phrases and each source phrase of $\bar{f}_i$ is translated into a target phrase $\bar{e}_i$. The phrase translation probability is modelled as translation from target to source. This is the standard model for phrase-based statistical machine translation.

  After this stage, phrases may be reordered. Many researchers agree that dealing with word reordering is a big challenge in MT, as phrase order can vary from one language to another [69, 1]. However, they agree that a reordering model that reorders translated words to give a good translation should be used at some points to solve the various language structures [46, 70]. There are many lexicalised reordering models that can predict the orientation (monotone (M), swap (S), and discontinuity (D)) of phrase pairs such as LRMs, HRM [70]. However, Koehn has handled the reordering phase using a distance-based reordering model, which has been defined as the number of skipped words as shown in (2.32) [1]. He points out that a limited number of these words produce better translations than large reordering, which can result in worse translation [1].

Then, Koehn [1] extends the standard model to improve the translation quality by giving the language model more weights $\lambda_\phi, \lambda_d, \lambda_{\mathbf{LM}}$ for scaling the contributions of the three components as:

$$e_{\text{best}} = \arg\max_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)^{\lambda_\phi} d(\text{start} - \text{end}_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|e|} P\text{LM}(e_i|e_1...e_{i-1})^{\lambda_{\text{LM}}} \tag{2.33}$$

while $\phi(\bar{f}|\bar{e})$ phrase translation table, d reordering model , and $P\text{LM}(e)$ language model. By adding weights, he creates a long-linear model, which is used widely in the machine learning community. The long-linear formulae is:

$$p\,(e|f) = \exp \sum_{i=1}^{n} \prod_{i} \lambda_i h_i\,(e|f) \tag{2.34}$$

where e is the translation, f the input sentence, h_i an evaluation of each feature function, and λ_i feature weight [71].

Finding the best scoring translation according to the model is still a complicated problem in MT. The process of identifying the best scoring translation is called decoding [1]. In SMT, a phrase-translation table is used to test the translation options for a given sentence. Finding the best translation, in terms of language influence, is the 'search problem' in this task. Koehn [1] employs heuristic search methods in their work. However, heuristic searches are not guaranteed to find the best translation and this can lead to search error. Highest-probability translation also fail to provide a good translation (referred to as model error). Therefore, decoding algorithms are used in the SMT model to identify the best translation, such as: stack decoding [72]: beam search and A^* search, Greedy hill climbing decoding and Finite State Transducers Decoding (FSTD) [1]. Currently, there are many popular decoding toolkits for building MT such as Moses and FSTD. These are efficient and help save time.

- Factored Phrase-Based Model:
  A Factored Phrase-Based Model (FPBMT) is an extension to Phrase-Based MT. The main difference is the added linguistic factors at the word level in both source and target training data. The linguistic factors are usually; surface form, POS, lemma, and morphological features such as gender, count, case etc.. These factors are added to each token to become a vector of factors. Each factor presents a different level of annotation [71]. This also uses a log-linear approach as shown in equation 2.34.

Mapping lemmas	haus $- >$ house, home, building, shell
Mapping morphology	$NN \vert plural - nominative - neutral- > NN \vert plural, NN \vert singular$
Generating surface forms	$house \vert NN \vert singular - > houses$

Table 2.2: Translate the German one-word phrase *hauser* into English [1]

Input factored representations will translate into output factored representations. This process is broken into three mapping processes; 1) Translate input lemmas into output lemmas, 2) translate morphological and POS factors, and 3) generate a surface from a given lemma and linguistic factors [71]. In this model, all translation steps operate on phrase level and generation steps on the word level [71]. For example, when translating the German one-word phrase $Huser$ into English. The input representation of $Huser$ is:
surface-form $Huserr$ | lemma Hus | POS NN | count $plural$ case | $nominative$ | gender $neutral$. The three mapping steps are shown in table 2.2.

Each step will be given multiple choices. The number of given choices is reflective of the translation ambiguity. The factored model shows improvement in SMT. Adding linguistic factors plays a core role in solving some of the morphological and ambiguity problems and leads to better phrase mapping from source to target languages [73]. As in PBMT models, factored models use a combination of the same components (e.g., LM, reordering model, translation and generation steps).

- Hierarchical Phrase-Based Model:
  The Hierarchical Phrase-Based Model (HPBM) [74] takes the fundamental ideas of syntax-based modelling (i.e., the hierarchical structure of the language) and integrates it into a PBM. The name 'hierarchical' comes form using hierarchical phrases (phrases that consist of sub-phrases). Chiang [68] points out that the main motivation for proposing HPBM is solving reordering problems in PBMT. As in PBMT, learning reordering is done at the phrase level. In other words, learning reordering of words can be efficiently achieved using phrases. However, when it comes to reordering phrases, it fails. Therefore, HPBM uses hierarchical phrases (consisting of words and sub-phrases) to learn the reordering of phrases [74].

  HPBM uses a synchronous context-free grammar (CFG) rules, which is a formalism consisting of terminals and non-terminals as symbols in addition to rewritten rules. Terminals are words while non-terminals are POS tags and phrase categories. The rewritten rules in CFG align to a single non-terminal symbol on the left-hand side with at least one either terminal or non-terminal symbol on the right-hand side [1]. The CFC formula is:

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle \tag{2.35}$$

where X is a non-terminal symbol, γ and α are strings of terminals and non-terminals, and $\sim$ is an one-one correspondence between non-terminals occurrences in γ and α [74].

The weight of each rule in HPBM has been computed from the use of both the noisy-channel approach and log-linear model as:

$$w\left(X \rightarrow \langle \gamma, \alpha \rangle\right) = \prod_i \phi_i \left(X \rightarrow \langle \gamma, \alpha \rangle\right)^{\lambda_i} \tag{2.36}$$

where ϕ_i are defined as features by rules. In spite of the use of CFG in HPBM, it can be considered as a move toward syntax-based MT. The main difference between HPBM and syntax-based models is that HPBM is learned from bitext without any syntactic information. So, HPBM is formally syntax-based (uses CFG) not linguistically-based [74]. HPBM uses minimum-error-rate training of log-linear models and uses n-gram language models the same as PBMT. However, HPBM has displayed better performance in terms of the reordering model compared to PBMT models.

Syntax or Structure Models

In spite of the similarity between the PBMT model and the syntax-based model in the training pipeline, syntax-based MT models require adding linguistic annotation to the translation rules. These annotations can be added to the source or the target training data, or to both of them [75]. Capturing the language differences is the main advantage of using linguistic annotations in this model which leads to better MT performance specially for language pairs with different morphology [75, 76]. However, the availability of the parser is a limitation for this approach.

2.4.3 Statistical Machine Translation Tools

Despite improvements in this field, MT still faces big challenges due to the complexity of natural languages. Individual words may have alternative meanings and many possible translations. Moreover, natural languages from different language families vary significantly in their morphological or syntactical complexity. Therefore, researchers agree that language complexity at the morphological and syntactical levels is the core challenge. However, a significant body of research has been published that shows performance increases. The achieved improvements target the three main components of statistical methods: (Cross) Language Modelling, Translation Model and Decoder. As a result, many tools have been introduced across each of these components.

Statistical Decoder

Decoder is a SMT system toolkit that is used to find the highest scoring sentence in the target language for a given source sentence. SMT decoder requires Parallel data for training the translation model automatically. Sentences in the Parallel are aligned so as to have the same numbers of lines in both source and target languages. Nowadays, many software tools are freely available to build SMT systems. The Center for Language and Speech Processing at Johns-Hopkins University (CLSP/JHU) has implemented GIZA++ as an extension of the program GIZA (word-based models) [77], which is commonly used in word alignment. Also, the University of Edinburgh has implemented phrase-based MT decoders; **Pharaoh**(a beam search decoder) and **Moses** (which is the most popular decoder for SMT). In addition to, **SAMT** (a tree-based model), **Joshua** (for Parsing-based Machine Translation) [78], **MARIE** (n-gram-based SMT Decoder), and several more.

- Pharaoh:
  Pharaoh decoder was first introduced by [79] in 2003. It implements a beam search algorithm for phrase-based MT. In 2004 the decoder was improved by recombining hypothesis in order to reduce search space and has demonstrated better performance through faster processing [80].

- Moses:
  Moses is an open-source toolkit for statistical MT, it is used for building an Arabic-English SMT system. It involves taking the rules of each language to transfer the grammatical structure of the source language into the target language [45]. Moreover, the toolkit (Moses) also includes a wide range of tools for building a statistical MT system, such as a word alignment tool, language model tool and a reordering tool, as well as an automatic machine translation evaluation tool [81, 1].

- Joshua:
  Joshua also is an open source toolkit for SMT [78]. It is implements all synchronous context-free grammar (SCFGs) algorithms. Li et al. [78] state that

great effort has been applied to building this toolkit to enable it to be used and extended easily.

To reiterate: SMT decoder can produce SMT systems for any pairs of languages using the parallel corpus, which is a collection of aligned sentences in two different languages. Many corpus are freely available through multiple sources and across a wide range of language pairs.

Language Models

Language models (LM) have been used in speech recognition tasks since their introduction in 1983 [82]. They were initially proposed for NLP tasks such as: MT [83] and automatic spelling correction [84]. As in NLP, SMT uses large scale n-gram language models. The original framework of n-gram LM was developed in 1999 [85]. There are many proposed LM algorithm that have been implemented and tested for SMT and other NLP tasks: SRILM [86], IRSTLM [85, 87], RandLM , KenLM [88]. Each of these LMs has shown good performance over many NLP tasks. Researchers have focused their studies on the differences between these LM's performances. Federico and Cettolo [85] state that when running Moses with SRILM and IRSTLM , in terms of memory size, IRSTLM requires less memory (about half that of SRILM) during decoding. While SRILM outperforms IRSTLM in terms of speed (as IRSLM is 44% slower). KenLM uses a two data structures library (PROBING, and TRIE models) for language modelling. It has outperformed both SRILM and IRSTLM as it uses less memory and is 2.4 times faster than SRILM [88] and significantly faster than IRSTLM. However, in MT, to truly compare the performance of LMs, building a baseline MT is required.

Word Alignment

Word alignment is one of several steps in training a phrase-based SMT system. This process helps to extract the phrase pairs that are needed in translation system. Word alignment is a time consuming process and requires the majority of the time taken to build a SMT system [89]. The statistical alignment models: IBM1, IBM2, IBM3, IBM4, IBM5 [65] IBM6 [77] and GIZA++ are available, with Giza++ currently being the most popular word alignment tool for bilingual parallel training corpora [89].

PGIZA++ and MGiza++ have recently been implemented to improve Giza++ (in terms of speeding-up the alignment process). Gao and Vogel [89] studied the performance of PGIZA++ and MGiza++ and their experiments show a significant improvement in terms of speeding-up the training process compared to Giza++.

2.4.4 Neural Machine Translation

In many tasks neural models have shown significant improvements: semantics and question answering [8, 9, 10], Machine Translation (MT) [11, 12], parsing [13, 14] and lexical representations, which will be explained in detail in Chapter 5. Within the last decade, research has achieved a great success following phrase-based approaches, like log-linear models. However, there are still many limitations. Particular weaknesses are; word alignment, reordering and Out-Of-Vocabulary Handling (OOV). Most of these problem are caused by data sparsity.

Therefore, many researchers have introduced different neural networks to improve machine translation quality. The proposed models can be categorised into two main categories. Firstly, hybrid approaches which are apply neural models to capture statistical sub problems. Secondly, end-to-end NN MT approaches, which encode the source sentence into the target sentence.

End-to-end approaches using Neural Machine Translation contain only a single component: A neural network trained to maximise the conditional likelihood on bilingual training data. Basic Architecture includes: an encoder (to encode the variable-length source sentence into a real-valued vector), and a decoder (to decode the real-valued vector into a variable-length target sentence) [90]. Liu et al. [91] propose a semi-supervised method that models the end-to-end decoding process for SMT using a Recursive Recurrent Neural Network (RRNN), which is a combination of recursive neural networks and recurrent neural networks.

Despite the success of NMT, it still has many limitations. One of these limitations is producing high OOV rate due to using a fixed sized of vocabulary to reduce training and encoding time. Moreover, end-to-end NMT encodes the source sentence into a fixed-length vectors that are unable to capture all the necessary features of long sentences. Finally, NMT cannot benefit from the large monolingual data needed to train good LMs for the source language. These limitations encourage researchers to incorporate NMT with SMT models in order to produce more powerful MT models. Recently, the Attention-based encoder-decoder model was introduced [92]. This model uses attention mechanism to connect the encoder with the decoder. Attention function is defined as the process of mapping a query and a set of key-value pairs into output (the weighted sum of the values) [92].

2.4.5 Hybrid Methods

In the past decade, various neural architectures have been introduced. From the simple feed-forward neural networks (FNN), through recurrent neural networks (RNN), to autoencoders. The rapid development in NN-based NLP models, namely, word embeddings (learning syntactic and semantic word representations) models has improved different NLP tasks and led to better context modelling. Therefore, many NN-based models have been proposed to solve different SMT problems [93, 94, 95]. He et al. [94] incorporate SMT models; namely language modelling and translation models with NMT in order to get the power of training these models using a large monolingual corpus which is not possible with a NMT model alone. Their model improves the translation quality and outperforms the state-of-the-art NMT system over the 2 BLEU score. Moreover, Luong et al. [96] translate the OOV words using a dictionary as a post-processing step.

Xiong et al. [97] propose a reordering model using a maximum entropy (MaxEnt) based on ITG (Max-Ent ITG) using the two reordering rules (straight and inverted). As fore-mentioned, learning word embeddings motivates researchers to investigate the effect of learning vector space representations for words/phrases on modelling word order. Thus, Li et al. [98], use the ITG reordering clarifier to introduce a NN-based reordering model based on recursive auto-encoders. RNN-based models have also been used for language and translation modelling. Sundermeyer et al. [99] use RNN to propose two translation models: word-based and phrase-based. Both models rely on in-source and target-word vocabulary to build word/phrase representations. For the phrase-based translation model, they introduce a bi-directional NN-based architecture to model unlimited past and future dependencies.

2.5 Machine Translation Evaluation

Due to the rapid development of MT systems and their importance within the field, evaluating each system's performance is a vital step in understanding which system to use. Koehn [1] points out that MT is not as easy to evaluate as other natural language tasks such as speech recognition, which gives one correct answer to match. In contrast, different MT systems can translate one sentence in multiple ways. Therefore, Machine Translation Evaluation (MTE) becomes a very active field of research [1]. There are two ways of achieving MT evaluation: manual/human evaluation, and automatic evaluation. Each of which, have outputs based on fluency and adequacy criteria.

2.5.1 Human Evaluation

In 1990s, the Advanced Research Projects Agency (ARPA) introduce a MT evaluation methodology using fluency, adequacy, and comprehension manual evaluation for measuring MT quality [100]. Adequacy is calculated sentence by sentence using evaluators based on human judgement [1] and its correctness relies on two criteria: fluency and accuracy [1]. All manual evaluators use scores around the same average

$\bar{x}$, which consists of a set of judgements $(x_1, ..., x_n)$ and is defined as:

$$\bar{x} = \frac{1}{n}\Sigma_{i=1}^{n} x_i \tag{2.37}$$

However, there are several disadvantages of manual evaluation metrics. These issues are:

- **Speed**: it takes a lot of time.

- **Size**: it is not suitable for long documents.

The MT manual evaluation assesses and measures the translation quality (accuracy and fluency). In translation quality, **accuracy** is understood by whether, or not, the target text reflects the source text accurately. While **fluency** is whether features are presented correctly such as grammar, spelling, typography, etc. Multidimensional Quality Metrics (MQM) are one of the most effective translation quality assessment tools. MQM categorises issues into 5 error analysis types: fluency, accuracy, verity, design, and internationalization. Each of these are further broken-down into many branches. **Design** is a physical design or presentation include character, paragraph, and UI element formatting and markup, text integration with graphics, and page or window layout. While **verity** is the text contains statements that contradict its setting. Finally, **internationalization** error, which is an issue about the internationalisation of contents. The first three issues are considered the MQM 'core issues' [101].

2.5.2 Automatic Evaluation Metrics

An automatic evaluation method for MT has recently attracted researchers' attention. The main objective of this method is to evaluate MT system quality in a cheap and rapid way [102, 1]. The big challenge in automatic machine translation is how quickly it produces a good similarity measure. Recently, automatic machine translation evaluation has become an active research field in MT, with many metrics being proposed:

- Precision and Recall compare each translation system against one or more reference translations (human translations) of the same sentence.

- (Balanced) F1-Measure (F1-Score) is a a combination of recall and precision.

- Word Error Rate (WER) is the rate of translation errors for each word. This metric requires the translation to have the correct order. For a perfect WER the word order in the hypothesis and reference must be identical.

- Position-Independent Word Rate Error (PER) is an error rate metric introduced by [103], where word order is fully ignored. Instead it calculates the difference between the number of words found in hypotheses and references. The resulting value is divided by the reference's word count.

- Translation Error Rate (TER) was introduced by [104]. The TER is an evaluation method that avoids the labor-intensiveness of human judgments and the knowledge-intensiveness of more meaning-based approaches. Machine Translation specialists utilise the Translation Error Rate (TER) to estimate how much post-editing is required for machine translation tasks. The automatic metric counts how many steps it takes to change a translated segment in a given line with one of the reference translations. It is simple to use, language agnostic, and aligns with post-editing effort.

- Bilingual Evaluation Understudy (BLEU) Score: BLEU scores are widely used as an automatic evaluation metric [1, 102, 105]. BLEU was introduced by Papineni et al. [106]. It is similar to PER in terms of mismatch measuring but considers matches of large n-grams against reference translations. Despite the main use of BLEU being MT evaluation, it is also used as a loss function for discriminative training [81, 78, 102]. The BLEU method is based on N-gram models as well as a set of human reference translations [105] and is defined as:

$$BLEU - n = brevity - penalty \exp \Sigma_{i=1}^{n} \lambda_i \log precision_i \tag{2.38}$$

while $brevity - penalty$ defined as:

$$\text{brevity-penalty} = \min(1, \frac{\text{output-length}}{\text{reference-length}}) \tag{2.39}$$

However, the maximum order n for n-grams to be matched is set to 4 and the weights λ_i is set to 1. Therefore, the metric is called BLEU-4 and simplifies as:

$$BLEU - 4 = \min(1, \frac{\text{output-length}}{\text{reference-length}}) \Pi_{i=1}^{4} precision_i \tag{2.40}$$

In spite of BLEU methods giving a score from 0 to 100, [1] believes that the actual score is meaningless as there are many factors which can affect the results, such as the number of reference translations, language pair and the tokenization scheme. NIST works similar to BLEU and has been developed by Edinburgh University Statistical Machine Translation Group [107].

- METEOR is an automatic machine translation metrics based on a concept of uni-gram (word-word) matching between the output of a machine translation and one or more human reference translations [108]. For more than one human reference translation, the output translation is scored by matching each reference independently and then taking the best scoring pairs [109]. It also incorporates a stronger emphasis on recall and functions in a similar way to BLEU by ignoring the near matches [1]. However, it has been developed to address BLEU's weaknesses [108].

- (Rouge-L) Recall-Oriented Understudy for Gisting Evaluation has tools for automatically evaluating a summary's quality by contrasting it with other ideal summaries written by humans. Between the computer-generated summaries being evaluated and the ideal summaries created by humans, the measurements count the amount of overlapping units, such as n-grams, word sequences, and word pairs. Rouge-L has an impact on MT despite being successful in automatic evaluation for summeries [110].

- CIDEr (Consensus-based Image De-scription Evaluation) a novel paradigm for evaluating image descriptions based on human consensus introduced by [111]. The CIDEr metric compares the similarity of a generated sentence to a set of human-written ground truth sentences. This metric demonstrates high agreement with consensus as determined by humans. The notions of grammaticality, saliency, importance, and accuracy (precision and recall) are inherently captured by this metric when using sentence similarity.

2.6 Syntactic and Semantic Parsing

The syntactic and semantic parsing process, from the point of view of linguistics, is to disclose how words are combined to form sentences and calculate the relations between these words [112]. In other words, the purpose of parsing is to reveal how words are combined to make sentences and the rules that control their construction. This process is useful for a variety of tasks in NLP applications, including machine translation, question answering, information extraction, sentiment analysis, and information extraction [113, 114, 115]. The performance of parsing is critical, as such, it has received extensive research attention in the past decades.

Syntactic parsing is the process of extracting syntax information from sentences, such as subjects, objects, modifiers, and topics. Parsing frequency uses grammars to refine the output structures of syntax and semantics. Many advanced grammars are available for accurately expressing syntactic and semantic information at the sentence level. For example: Context-free Grammar (CFG) [116], which is known as constituent parsing (or phrase-structure parsing), and Dependency Grammar for syntactic and semantic parsing, where words are linked directly by dependency connections, and labels indicate their syntactic or semantic meaning. There have been several successful solutions posed with large-scale corpora for a variety of languages are already available.

With regard to machine translations, syntactic and semantic parsing are important because as research state that integrating syntax features improve many NLP tasks [117]. This thesis investigates the impact of such features in learning BWEs and NMT as shown in Chapter 5.and 6.

2.7 Summary

This chapter presents a background on the field of research of word embeddings. It explain this approach at monolingual and bilingual levels. Also, we discuss approaches and methods of related research work in more details. In addition, this chapter provides an overview of the history of translation systems and its evaluation methods. In this thesis, in Chapter 6, a syntax-based NMT models is trained to investigate the effect of syntax features on the quality of MT task at the phrase level.

Chapter 3

Challenges for Machine Translation of Arabic Language

Arabic language structure is described in this section, including its morphology and semantics. From the Gulf to Morocco, 22 countries use Arabic as their official language. Countries and even areas within the same country have their own unique versions. Classical Arabic, Modern Standard Arabic (MSA), and Arabic dialects are only a few of the various varieties of Arabic. When it comes to news broadcasts, MSA, a form of Arabic based on classical Arabic syntax and morphology and phonology, is written and spoken. However, Arabic dialects are the genuine native language forms for everyday communication. The Arabic language is still a problem in MT because of its complexity. This thesis focuses on MSA.

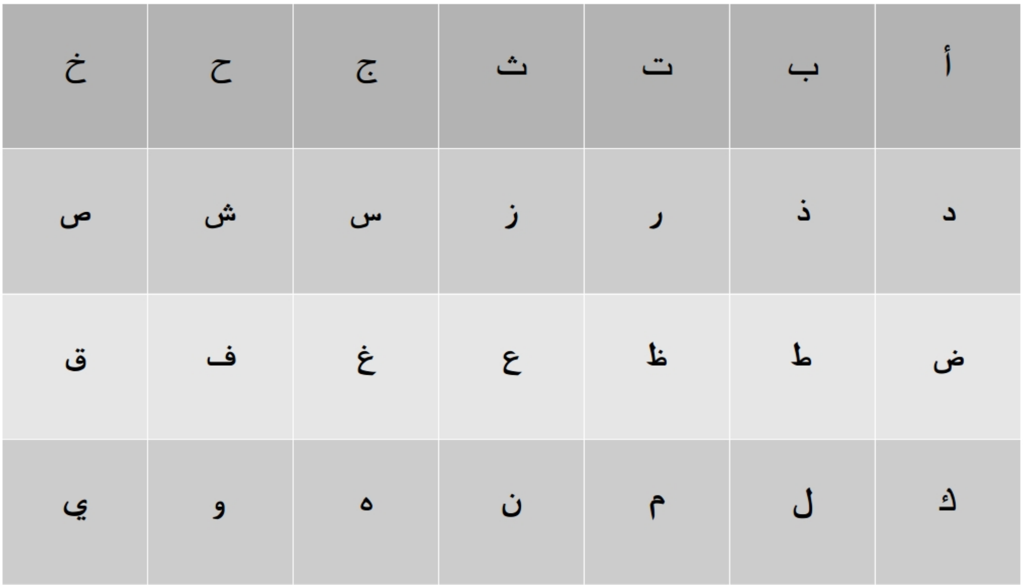
3.1 Arabic Alphabets

As with any language, the Arabic language has vowels and consonants. It consists of many types of letters written from right to left [118, 119]; the basic 28 letters (see Figure 3.1), the Hamzated Alif letters (see Figure 3.2), the Ta-Marbuta letter (see Figure 3.3), and the Alif-Maqsura letter (see Figure 3.4) [118].

Arabic letter formats can vary depending on the letter's position in the word [119] as some letters can take up to three different shapes. For examples see Figure 3.5.

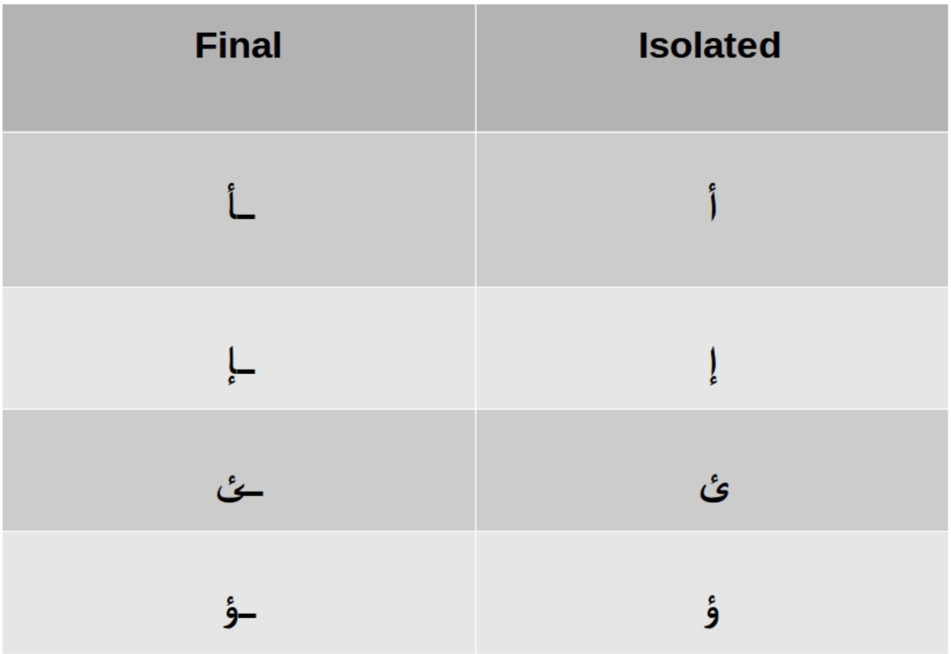
3.2 Arabic Words

Words in Arabic, like those in many other languages, can take on different meanings depending on the prefix or suffix they're attached to. It is common for prefixes to express the tense and gender, whereas suffixes are used to specify the number of



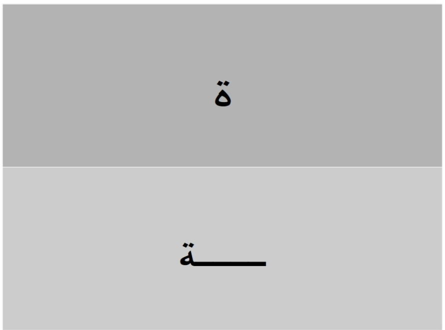
خ	ح	ج	ث	ت	ب	أ
ص	ش	س	ز	ر	ذ	د
ق	ف	غ	ع	ظ	ط	ض
ي	و	ه	ن	م	ل	ك

Figure 3.1: The Basic 28 letters in Arabic language



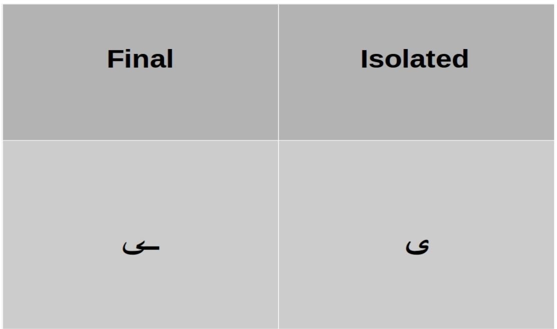
Final	Isolated
ـأ	أ
ـإ	إ
ـئ	ئ
ـؤ	ؤ

Figure 3.2: Hamzated Alif letters



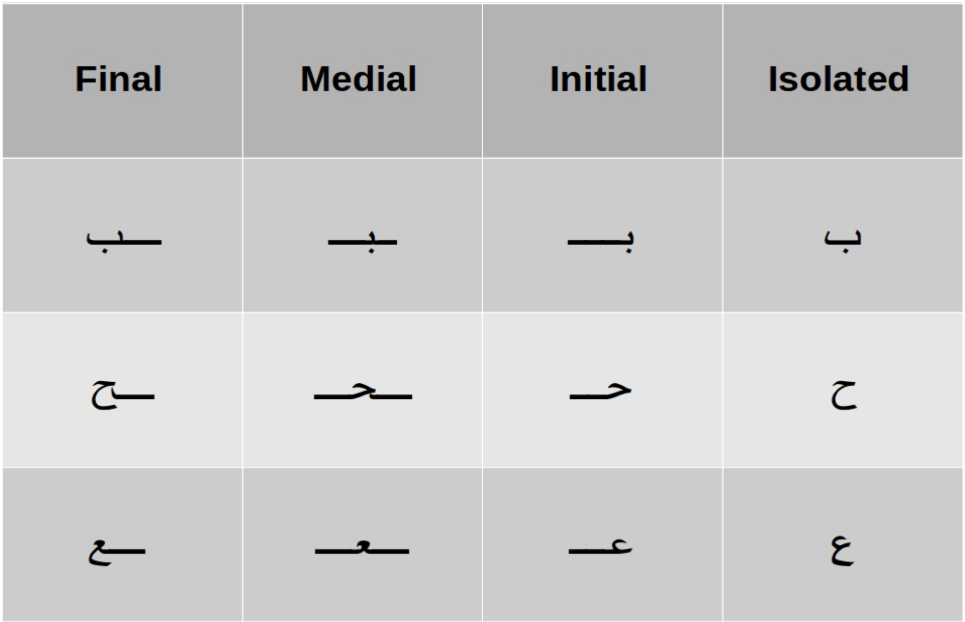
ة
ـــة

Figure 3.3: Ta-Marbuta letter in its different forms



Final	Isolated
ـى	ى

Figure 3.4: Alif-Maqsura letter



Final	Medial	Initial	Isolated
ـــب	ـــبـــ	بـــ	ب
ـــح	ـــحـــ	حـــ	ح
ـــع	ـــعـــ	عـــ	ع

Figure 3.5: Variation in Arabic letters shapes sample

people and the gender [69]. As a result, a single Arabic word can be translated into as many as three different English words. Thus, an Arabic word can be interpreted in several ways by modifying its affixation. Affixation is a problem in SMT within the Arabic language. Researchers have used morphological analysis to handle Arabic affixes in order to improve the Arabic-English MT System [120].

3.3 Arabic Morphology

In the Arabic language, the morphology and syntax are difficult to understand [121]. Despite the fact that numerous publications focus on translating Arabic to English, their authors are frequently constrained by the complexity of the morphology they encounter.

Firstly, verb tenses in the Arabic Language are: **Madi** (past), **Modare** (present) and **Amr** (order). The future tense is indicated by adding a word

سوف

or the letter

س

before the present tense verb. Regardless of the verb tense, verb structures vary in Arabic and rely on many factors: gender, single or plural.

For example, the verb:

تذهب

in English means ’she goes’. The first letter

ت

indicates the female gender, while the verb

يذهبون

means ’they go’. The letter (prefix):

ي

indicates the tense, which is present in this case,
while the (suffix):

ون

indicates plural and gender (male).

3.4 Lexical Semantics

Getting the same meaning over from one language to another is the primary goal of translation. Researchers, on the other hand, agree that determining what a word means depends on the word that comes after it in many languages. This is in addition to whether it is a noun, proper noun, verb, or particle [1, 69].

The lack of regular use of diacritization (marking to denote short vowels) creates uncertainty, particularly in Arabic. Although they share the same spelling, these two terms mean very distinct things, are called **homonymy** words or **word sence**. It has been defined as the task of determining the right word sense for a given word as **word sense**, **disambiguation** [1].

3.5 Sentence Structure

The Arabic language has two types of sentences: nominal (starts with a name) and verbal (starts with a verb). The Arabic and English languages are very different from a structural point of view. One of the main differences between Arabic and English is the order of words. As with other languages, Arabic sentences are built using a verb, subject and object. Usually, an Arabic sentence is post-verbal (VSO) so the verb comes first and then the subject is followed by the object. However, it is possible to be pre-verbal (SVO) as in the English language, however, this is not always preferable [122]. In both cases, VSO or SVO, an Arabic sentence is flexible with its verb position. However, the subject must come before the object (with the exception of passive sentences, which can be either before their subject or without their subject). Secondly, in Arabic, the adjective always comes after its noun, which is not the case in English. So a reordering rule should move the object of an Arabic sentence to the right of the adjective. Finally, indicating possession and compounding in Arabic is called Idafa. Idafa consists of one or more nouns that have been defined by the following noun [122].

3.6 Phonemes

Phonology has been defined as the study of how phonemes are organised in natural languages by [118]. Phonemes are defined as small linguistic units used to present speech [123]. Sibawayh identified Arabic phonemes in the thirteenth century [124].

The Arabic language has six different vowels. These vowels are divided into two categories [124]: Short vowels and Long vowels

The long vowels are letters, so they are easily recognised automatically, even by a recognition system in speech data or a translation system in text data. On the other hand, there are no special letters for short vowels. Instead, special marks above and beneath the consonants are used [119]. These marks are called diacritisation (Fatha, Kasrah and Dammah (See Figure 4)) [120, 69]. Consonants can be un-vowelled in the Arabic language by placing **Skoon** above the consonant.

Un-vowelled and short vowels are still big challenges in speech recognition systems and machine translation systems when using the Arabic alphabets corpus as they are not written letters . However, such issues can be managed using the English alphabets Arabic corpus. Moreover, in the Arabic language there are **Sun** and **Moon** laams, which means *the* in English. They are written exactly the same but they are pronounced differently depending on the following consonant. The Moon laam is pronounced as written */al/* while the Sun laam is pronounced with silent l.

3.7 Challenges

Compared to English, Arabic is considered a very rich morphological language. This morphological complexity increases the challenge when translating from and into Arabic language in MT, specifically SMT. Word order and word agreement can be considered as main issues for Arabic language in SMT in addition to some other issues. These issues can be explained briefly as follow:

- Arabic Word-Order and MT:
  Similar to many other languages, Arabic has different types of sentence structures which can create issues for MT. One of the main problems is deciding the position in which the verb and subject should be placed. An Arabic sentence can be either post-verbal (VSO) or pre-verbal (SVO) meaning the verb can take either position. We note that the VSO order is the more common (which is not the case in English word order) and in many cases the word order is fixed. Noun-Adjective positions are also a problem. In Arabic, adjectives follow their noun while in English the noun follows its adjective. This word order differentiation between these two language pairs makes SMT task more challenging.

- Arabic Word-Agreement and MT:
  Word-agreement is the most challenging issue in MT for Arabic language as there are many word-agreement rules. These can be broadly classified into three main word-agreement issues in Arabic language MT; verb-subject, noun-adjectives, and numbers quantification.

  Firstly, in verb-subject agreement, the verb must agree with its subject across many aspects; gender, number, and person. These rules can vary depending on the rationality status and can change depending on the sentence structure. For example; in verbal sentence, the verb must agree with its noun in gender and person but not in number, while in nominal sentences the verb must agree with its noun in gender, number and person.

Secondly, there is variance in noun-adjective agreement between the two languages. In Arabic an adjective must agree with its noun in both number and gender while this is not the case in English. In English, the adjective has one form regardless its noun case.

Finally, there is the problem of numbers quantification as numbers have unique agreement rules [125]. These rules can be classified into many main cases:

- The numbers 1 and 2 in Arabic must follow the gender of the noun they refer to and are positioned after the noun. For example;

  كتاب واحد

- Numbers 3 to 10 always take the opposite gender to the noun they represent and are positioned before the noun. For example;

  ثلاثة طلاب

  "Three students"
  the word

  ثلاثة

  which means "Three" is in female gender form while the word

  طلاب

  which means" Students" is indicates male gender.

  أربع تفاحات

  "Four apple".
  the word

  أربع

  which means "Four" is in a male gender form while the word

  تفاحات

  which means" apples" is female.

- multiple digit numbers are considered as consisting of two or more words. For example 13 is considered as 3 and 10. Numbers that consist of two words (or two parts) have two rules as follow:
  - ∗ Numbers 11 and 12; always agree with the gender of the noun.
  - ∗ Numbers 13 to 19; the first part of these numbers, which is a number between 3-9, must take the opposite gender to its noun, while the second part, indicating the 10, must agree with the gender of its noun. For example; take the sentence

    خمس عشرة سيارة

which means "Fifteen cars". The word car is a female single noun. The number 5 must, therefore, take the male gender, whilst the number 10 must take the female gender.
the word

خمس

which means "Five" is in male gender form while the word

عشرة

which means "Ten" is in female form

سيارة

which means" car" is female so the first part of the number take contrary gender of its noun and in single form.

A contrasting example would be

خمسة عشر كتاب

which means "Fifteen books". In this case books is a male plural noun. The 5 must therefore be female whilst the 10 male.
the word

خمسة

which means "Five" is in female gender form while the word

عشر

which means "Ten" is in male form.
And the word

كتاب

which means" book" is in male and single form.

– Numbers $\{20, 30....90\}$; have one form regardless of their noun's gender.

– The 100s numbers always take the female form when they are written as one word but follow the 13-19 rules if written in two parts.

- Affix and Clitics in Arabic:
  In Arabic, affix and clitic have been defined as the small liguistic units that attach to the stem. The main difference between them is that clitic is grammatically independent [126]. Therefore, in Arabic, one word can be translated as upto four words in another language. The reason behind this ambiguity is that the word consists of an affix (prefix and suffix) in addition to its stem and sometimes a clitic as well. A clitic is considered as a word in other languages. Therefore, researchers have done a great deal to reduce such ambiguity by proposing preprocessing techniques such as Word Segmentation. In Arabic,

affixes can not always be translated into the target language. In a noun, the affix indicates the gender and number. For example; take the word

أميرات

which means "Princesses " in English. The suffix

ات

indicates the number(plural) and the gender (female) and it is not a word itself. In contrast in verbs the affix can be more than one word and indicates gender, number, person, and tense. For example; the word

سيدعونهم

which means "they will invite them". This word has a clitic:

س

which means "will" and also two suffixes

ن

which indicates the subject's number (plural) and gender (male).

هم

refers to the object and indicates its number(plural) and gender (male). Clitics increase the lexicon size and cause alignment and matching issues [126].

- The Absence of Short Vowels:
  Short vowels absence in Arabic language is also considered as one of the challenges in Arabic Natural Language Processing (ANLP). As one word can has many meanings depending on these unwritten vowels. Therefore, the correct translation can be only found depending on the context. Thus, in MT, predicting the correct translation (especially when translating from Arabic) is the main challenge caused by such ambiguity.

- Ta-Marbota and Haa:
  The letter Ta-Marbuta is usually written without its dots (see Table 2.3), this causes MT to treat it as

  ه

  , which is a different letter all together (the same shape without dots).

- Alif-Maqsura and Ya:
  Alif-Maqsura and Ya have similar issues. Alif-Maqsura is rarely written with its dots making it appear the same as Ya (which doesn't have dots).

These issues within the Arabic orthography, where multiple forms of the same word can be interpreted, lead to increased ambiguity [126]. In addition, Arabic is written from right to left (similar to Chinese and Korean) and has no capitalisation, both of which are considered complex problems in ANLP [127]. Also, Part of Speech (POS) are difficult to define as Arabic linguistics are unclear [46].

Despite improvements, the fore-mentioned issues still have not been solved. They can be clearly seen in the En-Ar MT except "short vowels absence" issue. Translating between two different morphological languages, specifically from English into Arabic, is still far from the optimal. Picking the best translation, in terms of matching meaning, morphology and sentence structure, is still a big challenge for En-Ar SMT. In order to solve these issues more work need to be done in ANLP [127][128].

3.8 Arabic Language Preprocessing

In pre-processing, extensive research has been conducted into the impact of morphological pre-processing techniques on statistical machine translation (SMT) quality. Researchers agree on the importance of morphological and syntactic pre-processing in MT in terms of reducing both sparsity and the number of "out of vocabulary" words (OOV) [69, 129]. At pre-processing level, current research focuses on two main pre-processing techniques: word segmentation and word pre-ordering. Many tools have been introduced: AMIRA [130], MADA [131], MADA+TOKAN [132], Farasa [133], AlKhalil Morpho [134] and MADAMIRA [135].

MADAMIRA is a tool for morphological analysis and the disambiguation of Arabic including normalisation, lemmatisation and tokenisation. It can tokenise the input text with 11 different tokenisation schemes and normalise Alif and Ya characters. MADAMIRA has been developed in the same way as MADA to accept two input forms: MSA and Egyptian Arabic (EGY). Pasha et al. [135] have pointed out that MADAMIRA has outperformed both AMIRA and MADA and is state-of-the-art. In our work, as word order and language modelling have not been considered, we only applied segmentation and orthographic normalisation in the training datasets.

3.8.1 Word Segmentation

Word segmentation has been considered the same process as tokenisation within the Arabic language. It is one of many techniques that have been proposed to reduce morphological differences between languages such as Arabic and English [126]. Many tokenisation schemes have been introduced for Arabic and have been successfully applied. Researchers have studied the positive effect of morphological pre-processing on En-Ar SMT. El Kholy and Habash [129] found that tokenisation and orthographic normalisation improves the performance on SMT, especially when translating from a rich into a poor morphological language. Their work also shows that lemma-based word alignment improves the translation quality in En-Ar SMT.

Many researchers have studied the effect of different segmentation schemes in MT quality on both En-Ar and Ar-En SMT. For example, Habash and Sadat [136] show that rule-based segmentation improves the translation quality for a medium-sized corpus, but the benefit of word segmentation decreases when the corpus size is increased. Other researchers Al-Haj and Lavie [2] believe that tokenisation schemes with more splitting lead to a decrease in the OOV rate. On the other hand, increasing the number of token types can affect word alignment, translation model, and language model negatively as predicting these tokens correctly becomes more complex [129].

Researchers consider the Arabic tokenisation process one of the main solutions helping to decrease Arabic ambiguities in MT. There have been various rule-base segmentation schemes introduced (See Table 3.1). Some of these schemes are used in En-Ar SMT and they show the importance of word segmentation as a pre-processing step to minimise the differences between Arabic and English as well as its effects on SMT quality. The work of [137] shows a significant improvement in En-Ar SMT performance when combining segmentation with pre-processing and post-processing steps for small training data. Al-Haj and Lavie [2], El Kholy and Habash [129] have studied the effect of different segmentation schemes in En-Ar phrase-based machine translation (PBMT). Al-Haj and Lavie [2], in contrast to the previous work, investigate the effect of different segmentation schemes on a very large amount of training data of at least 150M words. Their work shows that simple segmentation performs better than complex segmentation as the complex segmentation has a negative effect by increasing the size of the phrase table.

3.8.2 Orthographic Normalization

Orthographic normalisation is an important process at the pre-processing stage. El Kholy and Habash [129] have introduced two schemes of orthographic normalisation: enriched Arabic (ENR) and reduced Arabic (RED). RED is used at the pre-processing level to convert all Hamzat-Alif forms to bare Alif (taking out Hamza) and Alif-Maqsura forms to Ya (add dots). ENR selects the correct Alif and Ya form in order to generate the correct Arabic form at the post-processing level.

3.9 Summary

In summary, machine translation offers a wide range of benefits. However, there are still many challenges. This is especially problematic when attempting to translate from complex language morphologies such as Arabic. Arabic has several features that make it challenging for MT (see Section 3.7). Bilingual word emeddings is the main aim in this research. According to research, joint learning produces more isomorphic embeddings, is less susceptible to hubness, and produces stronger out-

Table 3.1: Existing tokenisation schemes for Arabic [2]

D0/UT	No tokenization.
D1	Separates the conjunction proclitics.
D2	D1 + Separates prepositional clitics and particles.
D3/S1	Separates all clitics including the definite article and the pronominal enclitics.
S0	Splitting off the conjunction proclitic w+.
S2	Same as S1 but all proclitics are put together in a single proclitics cluster.
ATB	The Arabic Treebank is splitting the word into affixes.
S3	Splits off all clitics from the (CONJ+) class and all suffixes form the (+PRON)class. In addition to splitting of all clitics of (PART+) class except s+ prefix.
S0PR	S0 + splitting off all sufixes from (+PRON) class.
S4	S3 + splitting off the s+ clitics.
S5	Splits off all possible clitics (CONJ, PART, DET and PRON) classes.
S4SF	S4 + the (+PRON) clitics.
S5SF	S5 + the (+PRON) clitics.
S5ST	S5 + prefixes concatenated into one prefix.
S3T	S3 + prefixes concatenated into one prefix.
DIAC	One of MADA features that add diactresation to Arabic text.

comes in bilingual lexicon induction, indicating that current mapping methods have significant limitations. Therefore, BilBOWA model, which is a joint learning BWEs model, is used in this thesis. Thankfully a solution lies in incorporating dependency features has improved the learning process of BWEs for all used langauge pairs in this thesis and dramatically for Arabic-Englaish. In the following chapters we will apply these models experimentally to demonstrate how they can be used to translate different language pairs including Arabic.

Chapter 4

Bilingual Word Embeddings Without Word Alignments

Part of the following chapter has been published in the proceedings of the ACL Fourth Arabic Natural Language Processing Workshop [138].

A first attempt has been made in this chapter to address the research question what are the impacts of a variety of factors on learning bilingual word embeddings, including sentence length, and embedding size for three language pairs, including En-Es and En-De, in addition to morphological segmentation for En-Ar language pairs. Rather than aligning words, we used a bilingual word embeddings model without aligning words (BilBOWA).

According to our findings, for all language pairs, increasing the embeddings size leads the model to learn better bilingual word embeddings (BWEs). However, sentence length has different effects on the used language pairs. Using short sentences datasets improves the learning process of BWEs in En-Es language pairs. While it gives different effects on En-De and En-Ar language pairs. According to our results, for Arabic, utilising the D3 (more segmentation) segmentation scheme for morphological segmentation improves the accuracy of learning bilingual word embeddings by up to 10 percentage points when compared to the ATB (some segmentation) and D0 (no segmentation) schemes See (Table 3.1) in all training settings.

This chapter presents the related work as well as explains the used model BilBOWA, datasets and evaluation method. Finally, it shows the experiments in details.

4.1 Introduction

In the last decade, neural networks (NN) have attracted much attention and have shown very promising results in many natural language processing (NLP) tasks.

Many models have been introduced including: semantics and question answering [8, 9, 10], Machine Translation (MT) [11, 12], parsing [13, 14] and many works in word embeddings. Word embedding is one of the most important NLP tasks due to its ability to capture the semantic similarities between words.

The main idea behind learning word embeddings is to transform words from discrete space into a continuous vector space of features that capture their syntactic and semantic information. In other words, words that have similar meaning should have similar vectors. This similarity can be measured using different distance methods such as cosine similarity and Euclidean distance.

Now a days, many word embedding models have been introduced which show a significant improvement across different NLP tasks; language modelling [15, 16, 17], MT [18, 12, 19], named entity recognition [20], document classification and sentiment analysis [21, 22, 23] etc. Word embeddings can be classified, based on the objective function that needs to be learnt, into two main categories. Firstly, Monolingual word embedding, which is the process of learning similar word representations for similar word meaning within the same language. Secondly, Bilingual/cross-lingual approaches, which is the process of learning similar words between languages.

In this chapter, we investigate the effect of factors on learning bilingual word embeddings for En-Es, En-De and En-Ar language pairs. These factors are: sentence length and embedding sizes, in addition to different segmentation schemes for Arabic. The experiments show a noticeable accuracy change using different training settings. Firstly, we give an overview of some related recent works on bilingual word embeddings in Section 2.2. Chapter 3 gives a brief introduction to the Arabic language, and it describes the details of Arabic language morphological complex and preprocessing techniques. Next we present the experimental section that contains a description of the model architecture, training dataset, preprocessing settings and training hyper-parameters. The evaluation section presents the evaluation methods used as well as discussing the trained models' evaluation results. Finally, we conclude by demonstrating the outcomes and implications in Chapter 7.

4.2 Related Work

Bilingual or cross-lingual word embedding is the process of learning the semantic similarity across two or more languages word embeddings using two or more corpora. Many successful models have been introduced which use different model architectures and training corpora (with different alignment levels) to learn bilingual word embeddings. A selection of these are now discussed.

Firstly, at word-level alignment, Luong et al. [30] extend the skip-gram model to learn efficient bilingual word embeddings. Also, at phrase-level, a Bilingually-constrained Recursive Auto-encoder (BRAE) model learns source-target phrase embeddings by minimising the semantic distance between translation equivalents and maximising the semantic distance between non-translation equivalents [31]. Su et al. [32] extend the BRAE model by introducing a "bilingual correspondence recursive auto-encoder" (BCorrRAE) model, which incorporates word alignment to learn bilingual phrase embeddings by capturing different levels of their semantic relations. After that, Zhang et al. [3] introduce a Bi-dimensional attention-based recursive auto-encoder (BattRAE) model to learn bilingual phrase embeddings by integrating source-target interactions at different levels of granularity using attention-based models.

Using a sentence-aligned corpus, both Gouws et al. [33], Coulmance et al. [34] introduce BilBOW and Trans-gram methods to learn and align word embeddings without word alignment. With a document level aligned corpus, Vulic and Moens [35] present a model that learns bilingual word embeddings from non-parallel document-aligned data without using translation pairs. In addition, Mogadala and Rettinger [36] introduce a Bilingual paRAgraph VEctors (BRAVE) model that learns bilingual embeddings from either a sentence-aligned parallel corpus or label-aligned non-parallel document corpus. Vulic and Moens [35] also introduce a multilingual (two or more languages) word embeddings learning model using document-aligned comparable data.

In the literature we found three different bilingual embedding approaches: monolingual mapping, parallel corpus and joint optimisation. In monolingual mapping, word representations are learnt separately for each language using large monolingual corpora. Then, using word translation pairs, the model learns a transformation matrix that maps word representation from one language to the other [40]. Parallel corpus models require either word-level [41] or sentence level alignments [42, 43, 33]. These models aim to have same word/sentence representations for equivalence translations.

Finally, in the joint optimisation method, the monolingual and cross-lingual objectives are optimised jointly [33, 34]. Gouws et al. [33] propose a bilingual Bag-of-Words without word alignment model (BilBOWA) that uses a skip-gram model as the monolingual objective and jointly learns the bilingual embeddings by minimising the distance between aligned sentences, by assuming that each word in the source sentence is aligned to all words in the target sentence. This model shows success in translation and document classification tasks on Es-En and En-De languages pairs.

In the context of the Arabic language, no prior work has investigated learning bilingual word embeddings applied to such a morphologically complex language. Thus, in our work, due to the speed and success of BilBOWA models on learning bilingual words embeddings without word alignments, we train the model on Arabic because of its different language structure. This enables us to investigate the effects of complex language morphology in learning bilingual word embeddings. In addition, as this chapter addressing the RQ1, we investigate the factors effect namely: embedding size, sentence length on En-Es, En-De and En-Ar language pairs.

4.3 Experimental Setup

Due to the lack of research in investigating the effects of the used factors in this chapter: sentence length, embedding size (For all language pairs) and morphological segmentation (For Arbic language), the aim of this set of experiments is to evaluate the effect of these factors on the process of learning bilingual embeddings for all used language pairs (En-Es, En-De and En-Ar). We start with explaining the used model BilBOWA model in terms of model's architecture and learning objectives.

4.3.1 Model Architecture

The Bilingual Bag-of-Words without Alignments (BilBOWA) introduced in [33], is a simple and efficient model to learn bilingual distributed word representations without word alignment. In addition to these advantages, BilBOWA does not require the alignment process, which is a costly phase in NLP tasks. These advantages motivated us to use this model in our research. It assumes each word in the source language sentence is aligned to every word in the target language sentence, and vice versa, by using a sentence level aligned corpus, See Figure 4.1. This feature is an advantage of this model as the word alignment process is very time consuming. In this thesis, we adopt this model in addition to the effective and simplest BilBOWA model because it does not require the time-consuming alignment process. In the BilBOWA model both monolingual and bilingual objective functions are optimized jointly. The monolingual word representations are obtained by training word2vec using a skip-gram model which uses the negative sampling approach by [24]. The bilingual objective aims to minimise the distance between source and target sentences by minimising the mean of word representations in each aligned sentences pair.

4.3.2 Monolingual Objective

Instead of using softmax, Gouws et al. [33] implemented word2vec model using a simplified version of a noise-contrasting approach. The negative sampling training objective by [139] is modified as:

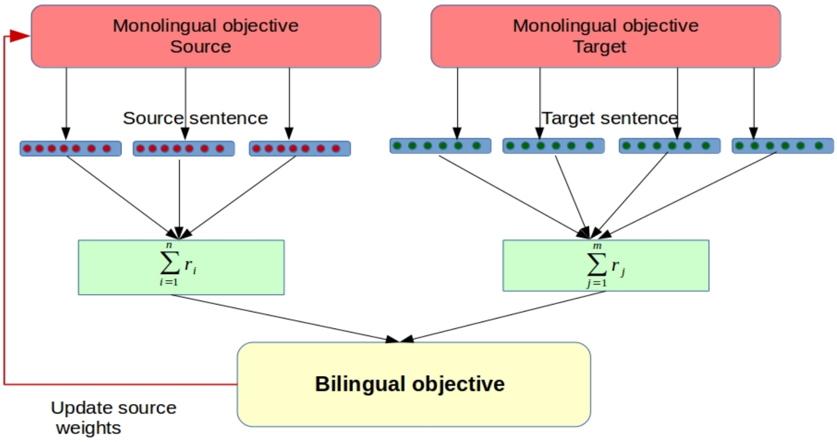


Figure 4.1: BilBOWA model

$$\begin{aligned} \log p(w|c) = &\log \sigma(v_w^{'T} v_{cp}) + \\ &\sum_{i=k}^{K} E_{w_i} \sim P_n(w)[\log \sigma(-v_w^{'T} v_{cn})] \end{aligned} \tag{4.1}$$

where v_w is word vector and v_{cp}, v_{cn} positive and negative context vectors respectively and K is the number of negative samples. This approach learns high-quality monolingual features and speeds up the computation process in this model architecture by converting the multinomial classification problem to a binary classification problem [139, 33].

4.3.3 Bilingual Objective

Gouws et al. [33] believe that, as is important between words in the same language, learning word representations that capture the relations and structure across languages may also improve performance. Therefore, the BilBOWA model learns word representations by updating the shared embeddings jointly for both monolingual and bilingual objectives. With the cross-lingual objective, this model minimises the loss between sentence representation pairs computed as the mean of Bag-of-Words of the parallel corpus. The bilingual objective is defined as:

$$\Omega = ||\frac{1}{m}\sum_{i=1}^{m} r_i - \frac{1}{n}\sum_{j=1}^{n} r_j||^2 \tag{4.2}$$

where m and n are the number of words in the source and target language, and r_i and r_j is a word representation for each language respectively.

4.3.4 Training Data

In this experiments, we used the most common parallel corpus, that been used for MT tasks, for all language pairs: En-Ar , En-De and En-Es. For En-Ar, we used *Web Inventory of Transcribed and Translated Talks* (WIT3), plain MSA Arabic and English language parallel corpus [140]. While Europarl-v7 and News Commentary-v6403 used for En-Es and En-De languages pairs monolingual and bilingual objectives respectively. The dataset has been divided into a 50,000 monolingual-dataset and a 24,000 bilingual-dataset to train the monolingual and bilingual objectives. After preprocessing (See Table 4.1), two different bilingual training datasets have been extracted based on sentence length: 5 - 10 and 17 - 80 tokens sentence length. Giving the distribution of sentence length in the corpus, these sentence lengths (5-10 and 17-80 tokens) give us a reasonable size of dataset and distinction between short and long sentences. For the test dataset, similarly to [33], we created a set of 3K words by extracting the most common words in the training datasets. Then, the extracted words were translated word by word using Google translator (as is common practice in the field) to create a word-based dictionary for all language pairs.

Table 4.1: Number of tokens in training datasets with different segmentation schemes. Note that preprocessing changes sentence length, and different methods therefore produce different datasets

Datasets	**5-10**	**17-80**	**Mono50K-data**
Arabic ATB	195985	901013	902307
English ATB	153111	551508	554338
Arabic D3	187612	975221	1033188
English D3	132687	520190	553414
Arabic D0	190854	773826	771512
English D0	158577	557664	553414

Both sides of the datasets, are tokenised, cleaned, normalised and stop-words have been removed. For Arabic, a morphological segmentation process is applied in order to minimise the differences between each En and Ar language pair. The literature shows many different segmentation schemes for the Arabic language (Table 3.1). We use MADAMIRA, a state-of-the-art Arabic morphological analyser, [135] for Arabic tokenisation, segmentation, and normalisation processes in this work. Three different training datasets with different segmentation schemes were generated: D0, ATB, and D3. For an example see Table 4.2. For English, Spanish and German languages, we used the Moses toolkit [81] for tokenising the English dataset,

Table 4.2: Used Arabic tokenisation schemes examples

Arabic Form	وتاثرت طفولتي بالريف لدرجة أعجز عن شرحها كما تميزت بالفكر بما يفوق توقعاتكم .
D0	wtAvrt Tfwlty bAlryf ldrjp qd AEjz En $rHhA kmA tmyzt bAlfkr bmA yfwq twqEAtkm.
D3	wtAvrt Tfwlp +y b+ Al+ ryf l+ drjp qd AEjz En $rH +hA k+ mA tmyzt b+ Al+ fkr b+ mA yfwq twqEAt +km.
ATB	wtAvrt Tfwlp +y b+ Alryf l+ drjp qd AEjz En $rH +hA k+ mA tmyzt b+ Alfkr b+ mA yfwq twqEAt +km .

and for cleaning both sides.

4.3.5 Training

After preprocessing, we trained a BilBOWA model using datasets with different settings: two sentence-length (5-10 and 17-80). For Arabic, three different segmentation schemes that give a range of segmentation amounts from no segmentation to more complex segmentation (D0, ATB and D3). The trained models produce different embedding sizes: (100D , 200D and 300D). As mentioned in [33], the Asynchronous Stochastic Gradient Descent (ASGD) algorithm has been used to train the model and updating all parameters for each objective function (monolingual and bilingual threads) with a learning rate of 0.1 with linear decay. The number of negative samples is set to NS=5 for the skip-gram negative sampling objectives as we examined NS=15 and it didn't show an improvement in our language pairs. All trained models were trained on a machine that was equipped with four Quad-Core AMD Opteron processors running at 2.3 GHz and 128 GB of RAM. The training process took up to 30 minutes depending on the model's embeddings size and sentence length.

4.4 Evaluation

As with word-level bilingual word embeddings (BWEs), similarly to [33], the trained BWEs were evaluated on a word translation task using *EditDistance* [139]. First, we extracted the most frequent 3K words from the Ar-En, De-En and Es-En datasets and preprocessed them similarly to the training dataset. Then, we translated the extracted words using Google translator to create a dictionary. After that, for source and target, we computed the distances between vectors in order to extract the embeddings of the k nearest neighbours for a given source word embedding in the target word embeddings.

After computing the similarity, we computed accuracy. The top k nearest neighbours (for $k = 1, 3, 5$) were selected to compute the accuracy among the test dataset, which consists of 3000 words and their translations. We computed the accuracy of 10 runs randomly selecting 500 source words and their k nearest neighbours as:

$$Acc = \frac{ct}{T} \tag{4.3}$$

where ct is the number of correct translations and T is the number of all test samples.

The accuracy was computed for all experiments across all settings: sentence-length, embeddings size and segmentation schemes. The results are discussed below. We also took into account the observed variance when considering the significance of the observed differences in performance.

4.5 Results

After computing the accuracy of each run, we computed the model final performance by computing the mean of the output values for each experiment as shown in Tables 4.3, 4.4, 4.5 4.6, and 4.7. Based on the observed accuracy and using sample/population standard deviation (SSD and PSD) to indicate significant differences our results cover three aspects of the problem:

- **Embeddings size:**
  Training the model on different embeddings sizes (100D, 200D and 300D) showed that, for all language pairs, increasing the vector size allowed the model to capture more information and lead it to learn better Es-En, De-En and Ar-En BWEs. Figures 4.2 ,4.3, 4.4 and 4.5 show an increase in accuracy when the size of word representation is increased.

- **Sentence length:**
  Comparing results from using short and long sentences, we found that using language pairs with different language structures affect the learning process differently. For Ar-En language pairs, long sentences (which increase the number of words "tokens") outperformed the short sentences in 300D embeddings size models across all three segmentation schemes. While short sentences perform better only with 200D embeddings size and ATB segmentation scheme trained models. In De-En language pairs, training the model using long sentences improve the results in all embeddings sizes. However, training the BWEs models

Table 4.3: 100D Models' Results

En-Ar 100D	**k=1**			**k=3**			**k=5**		
5-10	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	17.86	1.82	1.73	23.45	1.89	1.79	28.31	2.01	1.91
D0	15.32	0.97	0.92	18.82	3.85	3.65	20.99	2.44	2.31
D3	**18.98**	1.87	1.78	**26.04**	2.28	2.17	**28.32**	2.62	2.49
17-80	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	17.88	1.32	1.25	23.85	1.86	1.77	27.49	1.24	1.17
D0	16.14	1.76	1.67	19.99	1.74	1.65	21.94	2.37	2.25
D3	**22.92**	1.09	1.04	**31.59**	2.6	2.5	**33.82**	1.9	1.8

using a more similar language pair (Es-En) shows that when short sentences are used, training data outperforms models trained on long sentences. Thus, for more complex language pairs, long sentences with 300D embeddings size allowed trained models to capture more information and learn better bilingual word representations. For more details see Tables 4.3, 4.4, 4.5 4.6, and 4.7.

- **Segmentation schemes:**
  For Ar-En language pair, different segmentation schemes showed different levels of learning BWEs. D3, which is more segmentation (breaking the word into more tokens: and splitting all clitics), has a significant effect on the model learning process as it outperforms both D0 and ATB segmentation schemes (See Tables: 4.3, 4.4, and 4.5). In other words, increasing the number of tokens in training datasets using the D3 segmentation scheme, as shown in Table 4.1, leads to better word alignment and consequently improves the model performance.

For languages with different morphology and sentence structure namely Ar-En and De-En language pairs, increasing embedding size, sentence length and more Arabic segmentation allows the model to capture more information and leads it to learn better BWEs (Figures 4.2, 4.3 and 4.5).

For Figure 4.2, short sentences training dataset shows that both segmented datasets: ATB and D3 give better results compared to D0 (No segmentation). D3 slightly outperforms ATB. In Figure 4.3, using the long sentence training dataset, D3 gives a much better performance compared to either of the other segmentation schemes, and increases the accuracy dramatically up to 10 %. For similar language pairs (Es-En language pair), increasing the embedding size improved the learning process. However, in contrast to Ar-En and De-En language pairs, short sentences training datasets allowed the model to learn better BWEs compared to models

Table 4.4: 200D Models' Results

En-Ar 200D	**k=1**			**k=3**			**k=5**		
5-10	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	25.86	1.23	1.16	33.14	1.53	1.46	37.6	2.46	2.33
D0	21.19	1.65	1.56	27.71	2.12	2.01	30.28	1.81	1.72
D3	**26.34**	2.58	2.44	**34.74**	1.53	1.45	**37.02**	2.03	1.92
17-80	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	22.89	2.18	2.07	30.19	2.66	2.52	31.6	1.38	1.31
D0	22.22	2.17	2.06	28.87	1.67	1.58	31.32	1.55	1.47
D3	**32.83**	1.48	1.41	**41.06**	2.35	2.23	**43.9**	1.39	1.32

Table 4.5: 300D Models' Results

En-Ar 300D	**k=1**			**k=3**			**k=5**		
5-10	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	31.12	1.96	1.86	39.94	3.4	3.29	42.72	1.63	1.55
D0	26.88	1.65	1.56	33.99	1.10	1.04	37.67	2.63	2.50
D3	**31.8**	1.86	1.77	**42.48**	1.93	1.84	**44.74**	1.61	1.53
17-80	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
ATB	33.81	3.29	3.12	43.73	2.76	2.62	46.04	1.92	1.83
D0	30.38	2.09	1.98	37.09	1.73	1.64	40.39	1.98	1.88
D3	**40.38**	1.99	1.89	**49.16**	1.54	1.46	**51.25**	2.94	2.79

trained using long sentences See Figure 4.4.

4.6 Summary

In this chapter, to address our research question, we have trained a BilBOWA model to investigate the effect of different training settings on learning BWEs for Es-En, De-Es and Ar-En language pairs. We studied the effect of different training settings (sentence-length and embeddings size in addition to morphological segmentation for Ar-En language pair). For Arabic, as a morphological segmentation process is essential in many Arabic NLP tasks, segmentation has a positive effect and leads to learning better bilingual word embeddings. Going from D0 (full word form) to D3 (more segmentation, which increases the number of tokens in training dataset),

Table 4.6: En-Es Results

En-Es	**k=1**			**k=3**			**k=5**		
5-10	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
100D	22.22	1.47	1.39	26.29	2.68	2.55´	28.04	1.69	1.60
200D	31.34	1.49	1.41	35.56	3.0	2.84	38.98	2.02	1.91
300D	**36.34**	2.78	2.64	**42.9**	2.36	2.24	**44.48**	2.04	1.93
17-80	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
100D	18.72	1.10	1.04	23.62	1.63	1.37	27.1	1.75	1.66
200D	31.07	2.42	2.30	37.12	1.81	1.72	38.34	1.77	1.68
300D	**34.26**	2.12	2.01	**43.18**	2.64	2.51	**44.28**	2.39	2.27

En-De	**k=1**			**k=3**			**k=5**		
5-10	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
100D	31.5	1.636	1.552	37.7	1.833	1.739	39.18	2.825	2.68
200D	43.14	2.45	2.325	50.78	2.226	2.11	51.26	2.42	2.296
300D	**47.02**	2.27	2.154	**53.32**	2.549	2.418	**55.5**	1.914	1.816
17-80	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**	**Mean**	**SSD**	**PSD**
100D	41.6	2.03	1.926	49.32	2.253	2.137	51.14	2.074	1.967
200D	50	2.458	2.332	57.48	2.274	2.158	60.52	1.953	1.852
300D	**52.56**	2.64	2.505	**62.24**	2.427	2.3026	**62.48**	2.059	1.9538

Table 4.7: En-De Results

decreases the distance between Ar-En pairs and increases the similarity more than 10 percentage points. For all language pairs, our results show that increasing the word embedding size leads to improvement in the learning process of bilingual word embeddings. When we compared the findings from using short sentences to those from using long sentences, we discovered that the learning process is affected differently when applied to language pairings with distinct grammatical structures. Long sentences performed better than short sentences in 300D embeddings size models when employing any of the three different segmentation schemes for the Ar-En language pair. This is because long sentences increase the total amount of words, or "tokens." While shorter phrases tend to perform better, this is only the case when the trained model on En-Es language pair. When training the model on the De-En language pair using long sentences, the results are improved across the board at all embedding sizes. However, as mentioned above, training the BWEs models

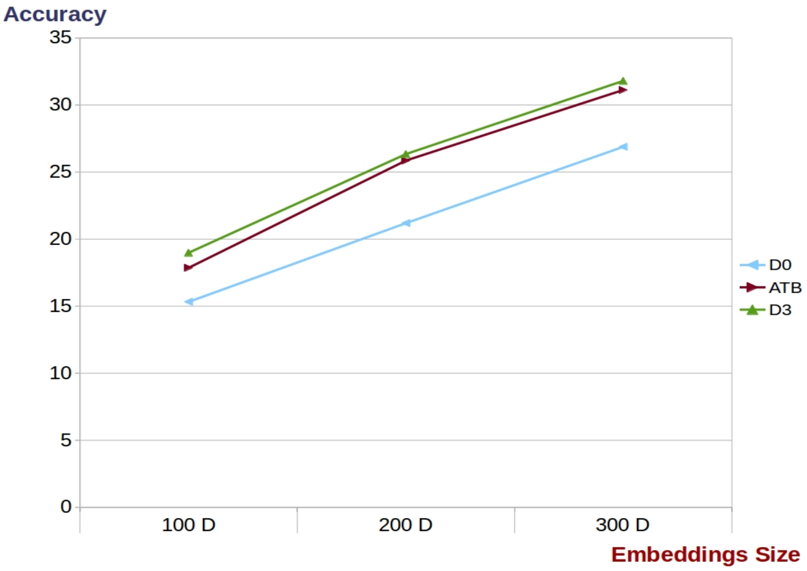


Figure 4.2: Ar-En translation accuracy using training samples of sentence length 5 - 10

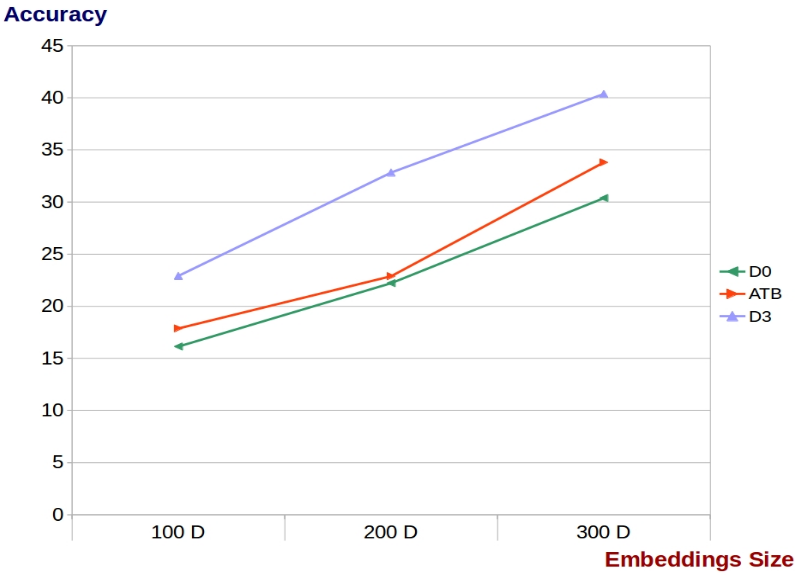


Figure 4.3: Ar-En translation accuracy using training samples of sentence length 17 - 80

using a language pair that is more comparable to each other, such as Spanish and English, suggests using short phrases as training data outperforms models trained on long sentences. Therefore, the model training should depend on the complexity and difference between the language pairs. For example: for more complicated language pairs, training models with long sentences that have an embedding size of 300D enables them to collect more information and develop stronger bilingual word representations. To conclude, to answer RQ1, our experiments show that increas-

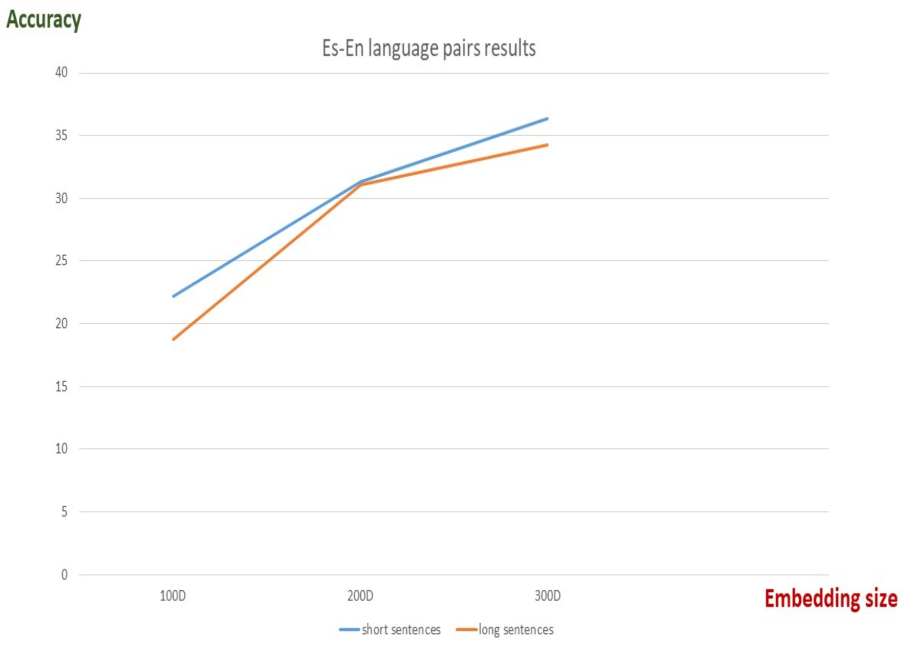


Figure 4.4: Es-En language pair models' results

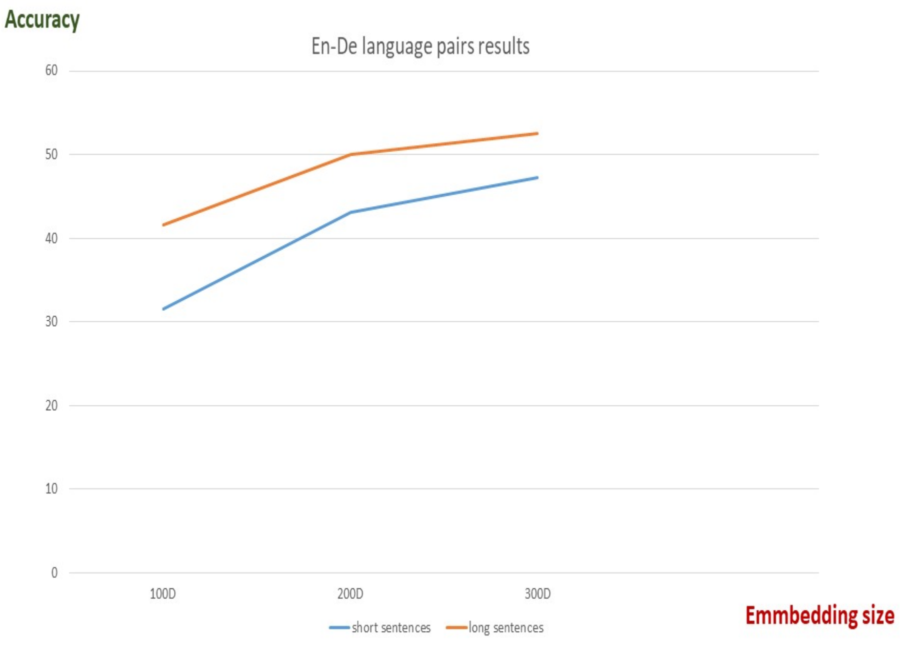


Figure 4.5: De-En language pair models' results

ing the embeddings size has a positive effect in all involved language pairs in this research, While sentence length is a language dependant feature. For Arabic, more segmentaion leads to learn better BWEs.

Chapter 5

Syntax-based Bilingual Word Embeddings without Word Alignments

Part of the following chapter was published in the proceedings of the International joint conference on neural network (IJCNN), 2020. In this chapter, we addressing second research question by investigating how incoroprating dependancy featues affect the learning process of bilingual word embeddings. We train Bilingual Bag-of-Words without Alignments (BilBOWA) models using linear Bag-of-Words contexts and dependency-based contexts. BilBOWA embedding models learn distributed representations of words by jointly optimizing a monolingual and a bilingual objective. We include dependency features in the training of the embeddings. When using these features to train towards the monolingual objective only, the accuracy improves by up to 6% in English-Spanish and up to 2.5% in English-German language pairs compared to the baseline model. However, using these dependency features in both objectives simultaneously, monolingual and bilingual, does not lead to any improvement in the English-Spanish language pair and only shows minor improvement for English-German translation. Moreover, our results provide evidence that using dependency features in bilingual word embeddings has a different effect based on the syntactic and sentence structure similarity of the language pair.

5.1 Introduction

Word embedding has improved across various Natural Language Processing (NLP) tasks by distributing word embeddings into a low dimensional continuous vector space. Dependency Parsing is the process of analysing a sentence's grammatical structure to identify related words and the sort of relationship between them. In our research, we integrate these features to learn bilingual word embeddings models based on the syntactic and semantic similarities between similar words in different languages. At monolingual level, Mikolov et al. [24] introduced a bag-of-words-based word embedding method that demonstrated a successful implementation on many NLP applications, including language modelling ([15, 16, 17]), machine translation ([141, 12, 19]), named entity recognition [20], document classification, sentiment analysis [142],[22] and [23] and parsing [143]. In cross-lingual word embeddings, many methods have been introduced for bi/cross-lingual word embedding. These methods drive similar words into a shared vector space of two or more languages. Bilingual word embeddings methods can be classified into four categories based on how the parallel corpus is used with different alignment levels: 1. A word aligned dictionary [30, 31, 32, 3], 2. Phrase/Sentence-aligned parallel corpus [33, 34], 3. Word and sentence level alignment datasets [33, 34] and 4. None aligned comparable datasets [35]. Luong et al. [30] extends the skip-gram model to learn an efficient bilingual word embedding. The bilingually-constrained recursive auto-encoder (BRAE) model learns source-target phrase embeddings by minimising the semantic distance between translation equivalents and maximising the semantic distance between non-translation equivalents (introduced by [31]). Su et al. [32] extended the BRAE model to produce a "bilingual correspondence recursive auto-encoder" (BCorrRAE) model by incorporating a word alignment that learns better bilingual phrase embeddings by capturing different levels of their semantic relations. Zhang et al. [3] introduced an attention-based method which uses a Bi-dimensional Attention-based Recursive Auto-Encoder (BattRAE) model that learns bilingual phrase embeddings by integrating source-target interactions at different levels of granularity.

Regarding sentence level alignment, many models have recently been developed including: the BilBOWA model [33] and the Transgram method [34]. These models learn and align word embeddings without word alignment. Moreover, Mogadala and Rettinger [36] proposes a Bilingual paRAgraph VEctors (BRAVE) model that learns bilingual embeddings from either a sentence-aligned parallel corpus or label-aligned non-parallel document corpus. While a multilingual (two or more languages) word embeddings model that uses document-aligned comparable data has been proposed by [35]. Xu et al. [144] utilise bilingual word embeddings with syntactic dependency (DepBiWE). They extract context from dependency parsed trees to be used jointly with Bag-of-Words context to learn bilingual word embeddings.

Obtaining word alignment is an expensive and time consuming process. In this thesis we follow the work of [145] and use an extension to the BilBOWA model, integrating it with syntax features. The BilBOWA model is trained by jointly optimising a monolingual objective for each language and a bilingual objective that aligns the representations of the two languages. The skip-gram objective with negative sampling is used as the monolingual objective while the bilingual objective minimises the Euclidean distance of the Bag-of-Words representation between the two languages in the embedding space. We compare four different methods by adding syntactic information to the BilBOWA model. We use a dependency based skip-gram model for the monolingual objective while keeping the bilingual objective the same (MonoDep-BilBOWA), or extending the Bag-of-Words representation with dependency features for the bilingual objective (BiMonoDep-BilBOWA).

The main contribution of this research is to consider different syntactic structures in learning bilingual word representations without word alignment. In this chapter, we show that the MonoDep-BilBOWA model, learns better bilingual word embeddings using Bag-of-Words and dependency contexts. We extend the BilBOWA model by integrating dependency features in both monolingual and bilingual objectives to investigate their effects on learning bilingual word embeddings on the cross-lingual dictionary induction (CLDI) task.

In Section 5.2, we give an overview of some related recent work on dependency-based word embeddings. Section 5.4 describes the proposed models. This is followed by the implementation section which contains the training dataset, preprocessing settings and training hyper-parameters for each trained model. The evaluation section explains the methods used to evaluate and presents our findings which are subsequently discussed in more detail. Finally, we draw our conclusions within section 5.7.

5.2 Related Work

In this section we describe existing work that is used as a basis for our experiments.

5.2.1 Monolingual Dependency-based Word Embeddings

Recently, estimating word representation has attracted attention as it shows very promising results across many Natural Language Processing (NLP) applications. Since the success of word2vec's skip-gram and CBOW models, several modifications have been proposed to integrate syntax features in the learning process [146], [7], [28]. The research shows that syntax-based embeddings capture better functional properties of words compared to their window-based counterparts. Omer and Yoav [28] modified the skip-gram model by replacing the linear Bag-of-Words context with features from a word's neighbourhood in a dependency graph. Komninos and Manandhar [7] propose another variation of dependency-based skip-gram word embedding model which extends the notion of token co-occurrence in a dependency neighbourhood to include additional pairs. This is compared to the model of [28].

They show that the dependency features can be used in various sentence representations to improve performance in several sentence classifications tasks. Li et al. [146] also introduces a multi-order dependency-based context into the skip-gram model with adaptive dependency weights.

5.2.2 Bilingual Dependency-based Word Embeddings

In terms of the learning process, bilingual word embeddings have been classified into three categories, 1. monolingual mapping, 2. cross-lingual training and 3. joint optimisation approaches. In monolingual mapping, after learning word representations separately for each language, the model learns a transformation matrix to map the word representation from one language to the word representation from another, using word translation pairs [40]. Parallel corpus models require either word-level [41] distributed or sentence-level alignments [42], [43] . These models aim to have the same word/sentence representations for equivalent translations. In the joint optimisation method, the monolingual and cross-lingual objectives are optimised jointly to enforce bilingual constraints[33], [34]. Gouws et al. [33] proposes a bilingual Bag-of-Words without word alignment model (BilBOWA) that uses a skip-gram model as the monolingual objective. It jointly learns the bilingual embeddings by minimising the distance between aligned sentences, by assuming that each word in the source sentence is aligned to all words in the target sentence. The model can utilize large amounts of monolingual data along with a few translation pairs of sentences. The model shows success in the English-Spanish (En-Es) translation task and the English-German (En-De) languages pair in document classification task. 19

Recently, Xu et al. [144] proposed the first model that learns bilingual word embeddings using syntactic dependencies. Their model learns the bilingual word embeddings using both dependency context and Bag-of-Words context. As with the Bag-of-Words method, word order has been ignored in cross-lingual scenarios as it can produce context words that are not related to the target words. Xu et al. [144] obtains the dependency contexts of aligned words to capture the syntactic information among languages.

5.3 Languages Syntax Differentiation

In our work, we use different language pairs with range from similar to different language sentence structures. English-Spanish (En-Es), English-German (En-De) and English-Arabic(En-Ar) language pairs. The Arabic language, which is the official language of 22 countries from the Arabic Gulf to Morocco with variant dialects between countries or within regions in the same country, has been chosen for our work as it is still a challenging language in MT. Arabic language structure is in many ways very different from English, Spanish and German languages. In this research, we focus on the Arabic language from Modern Standard Arabic (MSA) as it is the most accessible form. The Arabic language has two types of sentences: nominal (starts with a name) and verbal (starts with a verb). One of the main differences between Arabic and English is the order of words. As with other languages, Arabic sentences are built of verb, subject and object. Usually, an Arabic sentence is post-verbal (VSO) so the verb comes first and then the subject is followed by the object, whereas English, Spanish, and German are (SVO) or (SOV). However, it is possible to be pre-verbal (SVO) as in the English language, although not always preferred [122]. In both cases, VSO or SVO, an Arabic sentence is flexible with its verb position. However, the subject needs to come before the object (except in passive sentences in which it can be either before its subject or without its subject).

Secondly, in Arabic, the adjective always comes after its noun. While in English and German, the adjective comes before the noun. In Spanish, it may come either before or after. Finally, indicating possession and compounding in Arabic is called Idafa. Idafa consists of one or more nouns that have been defined by the following noun [122]. In our work, it is interesting to investigate how a neural network learns these languages' complexities by integrating more features, namely syntax features in the learning process and what benefit can the model gain from infusing these features.

5.4 Models

We extend the work of [145] and propose a dependency-based model. We investigate the effect of integrating the syntax features on different sizes of training data-sets for languages with different sentence structures. As in the work of [145], we used the BilBOWA model [1] , which is a simple model that shows efficiency in learning bilingual word embeddings, to train our proposed models with variant syntax information. Alqaisi et al. [145] state that a syntax-based representation of a sentence is a directed graph with one node per word and type labelled edges representing the syntactic relations between nodes. For the syntactic relation types, we used Universal Dependencies (UD) [147]. The UD types are specifically designed to be consistent among different languages, making them suitable for multilingual syntactic analysis. The dependency features were extracted from the parse tree. This enabled BilBOWA-Dep models to be implemented using different settings–modelling dependency features at the monolingual objective (words-relation-contexts (Mono-DepWRC) and

[1] https://github.com/gouwsmeister/bilbowa

words-contexts only without relations (Mono-DepWC)), and modelling dependency features at both monolingual and bilingual objectives (BiMonoDep-WRC) as described below.

5.4.1 Bilingual Word Embeddings without Word Alignment (BilBOWA)

Using a sentence-level aligned corpus, BilBOWA models was trained for En-Es, En-De and En-Ar language pairs as a baselines. The baseline model assumes that each word in the source language sentence is aligned to every word in the target language sentence and vice versa (this feature is an advantage of this model as the word alignment process is very time consuming). In the BilBOWA model, both monolingual and bilingual objective functions are learnt jointly.

- Monolingual Features:
  The BilBOWA model learns monolingual word representations using a skip-gram model with the negative sampling approach by [139]. The skip-gram model learns distributed representations of words by estimating the conditional probability of a target word w occurring in the context of word c. The (target, context) pairs are determined by a context definition function, which is typically a predefined window around each target word. To avoid the computational cost of estimating a categorical distribution over all possible words, the objective is converted to a binary classification problem. The target word is assigned a positive label and a small number of sampled words are used as the negative samples. The skip-gram with negative sampling training objective for a single sample is given in [139] as:

$$\log p(w|c) = \log \sigma(v_w^{'T} u_{cp}) + \sum_{i=ng}^{NG} E_{w_i \sim P_n(w)}[\log \sigma(-v_w^{'T} u_{cn})] \tag{5.1}$$

  where v_w denotes the target word representation and v_{cp}, v_{cn} represent positive and negative context representations respectively. NG is the number of negative samples and σ is the sigmoid logistic function. The objective is averaged over each word instance in the corpus and maximised by a stochastic gradient ascent. The skip-gram model maintains two different representations of each word:v_w to be used as the target word and v_c to be used a context word. The sampling distribution $Pn(w)$ is the unigram distribution of words estimated by their frequency in the training corpus, raised to the power of $3/4$. skip-gram also sub-samples training instances based on the frequency of the target word, i.e., frequent words have a higher probability of being modelled.

- Bag-of-Words Bilingual/Cross-lingual Features:
  The bilingual word embeddings are learned by minimising the distance between source and target sentence representations in each aligned sentence pair. In other words, the model minimises the mean square error loss between sentence representation pairs, where sentence representations are computed as the mean of their word embeddings.[33] defines the bilingual objective as:

$$\Omega = ||\frac{1}{m}\sum_{i=1}^{m} r_i - \frac{1}{n}\sum_{j=1}^{n} r_j||^2 \tag{5.2}$$

  where m and n are the number of words in the source and target language, and r_i and r_j denote the word representation for each language respectively. While this objective can be trivially minimised by setting all the vectors equal to zero, when used along with the monolingual objective it acts as a regularizer that forces the word representations of the two languages to share a common aligned space, where translation word pairs are close.

5.4.2 Dependency Based Bilingual Word Embeddings without Word Alignment (Dep-BilBOWA)

We used three different dependency-based BilBOWA models that learn word representations by updating the shared embeddings jointly for both monolingual and bilingual objectives using dependency context features. The BilBOWA model uses a skip-gram model to learn monolingual relations between words in the same language. In this chapter, we follow the work of [7], to extend the use of the skip-gram model and integrate dependency contexts with Bag-of-Words contexts, as explained below. The main purpose of the proposed models is investigate the effect of different levels of dependency features on the learning BWES.

- Model 1: BilBOWA Model(The Baseline)
  In This chapter, BilBOWA model is considered as the baseline model to compare with.

- Model 2: Monolingual Dependency-Based words-relations-contexts model (MonoDep-WRC)
  At the monolingual level, dependency-based skip-gram embedding models learn representations by extracting (target, context) token pairs from dependency graphs instead of word sequences. To encode the graph's structure, they use two types of tokens: words and dependency features. Words correspond to nodes of the dependency graph and dependency features are composite

features representing a node and an incident edge as a unit. We denote dependency features as a concatenated string of the edge type and word. The direction of the edges are encoded by adding $a^{-}1$ to the edge type if it is an outgoing edge. Dependency-based skip-gram models jointly learn distributed representations of both token types using the same objective as skip-gram, but change the context definition that determines co-occurring tokens from a window to a node neighbourhood. The extended dependency-based skip-gram [7] defines context as the (target, context) token pairs that can be extracted within the one-hop neighbourhood of a dependency graph node. In particular, pair extraction is performed by visiting each node in the dependency graph and constructing one bag with the neighbouring words and one bag with the dependency features formed by the neighbouring nodes and their edges. The centre node is added to both bags. The (target, context) pairs are then all the ordered pairs of tokens that can be formed within each of the two bags. In this model, the bilingual objective remains the same as the baseline model (Bag-of-Words sentence representations). For more details See Figure. 5.1

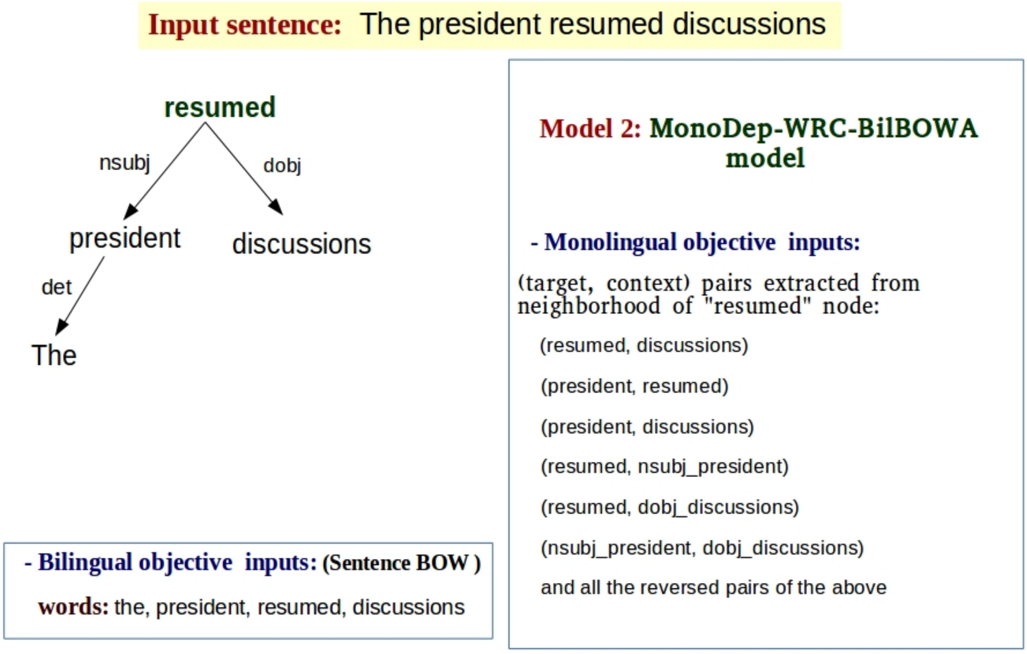

Input sentence: The president resumed discussions
resumed
nsubj
dobj
president
discussions
det
The
Model 2: MonoDep-WRC-BilBOWA model
- Monolingual objective inputs:
(target, context) pairs extracted from neighborhood of "resumed" node:
(resumed, discussions)
(president, resumed)
(president, discussions)
(resumed, nsubj_president)
(resumed, dobj_discussions)
(nsubj_president, dobj_discussions)
and all the reversed pairs of the above
- Bilingual objective inputs: (Sentence BOW)
words: the, president, resumed, discussions

Figure 5.1: Model 2 input features example

- Model 3: Monolingual Dependency-Based words-contexts model (MonoDep-WC)

This is similar to model 2 with the main difference that relations have not been included from the training dataset. In another words, only words and words contexts that have a syntax's relations have been considered in the learning process (See Figure 5.2).

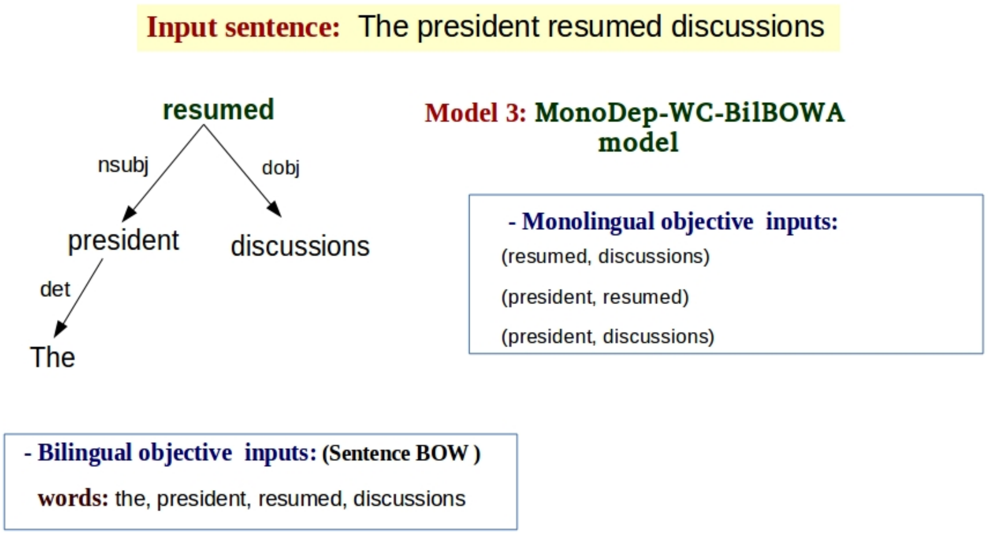
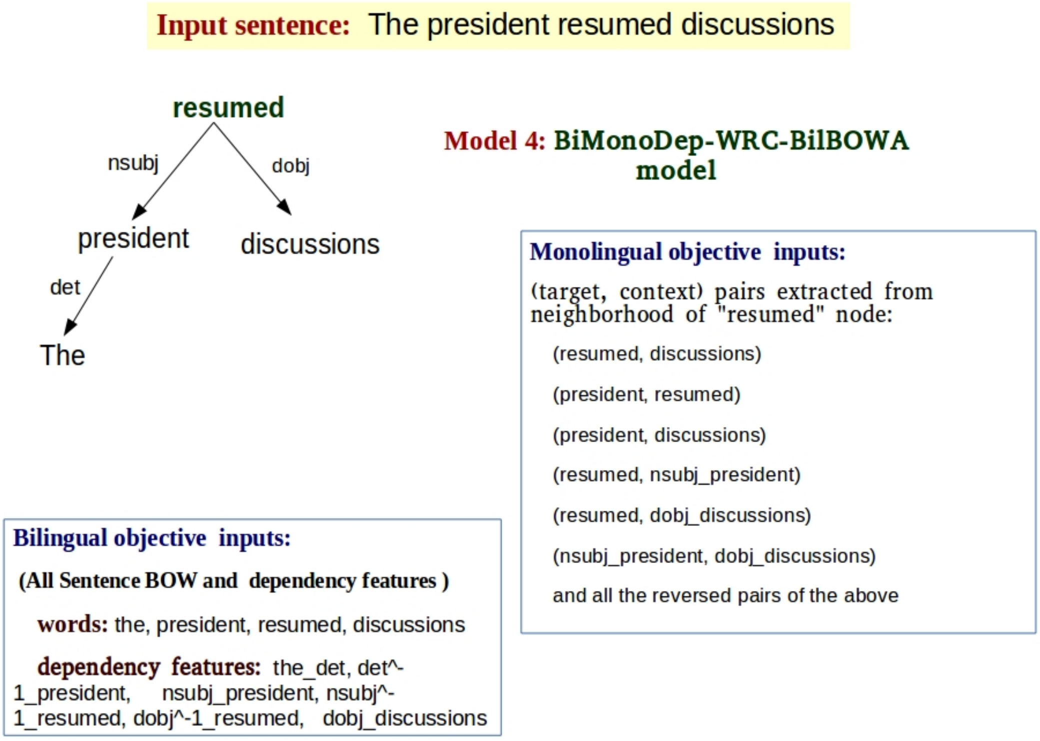

Input sentence: The president resumed discussions
resumed
nsubj
dobj
president
discussions
det
The
Model 3: MonoDep-WC-BilBOWA model
- Monolingual objective inputs:
(resumed, discussions)
(president, resumed)
(president, discussions)
- Bilingual objective inputs: (Sentence BOW)
words: the, president, resumed, discussions

Figure 5.2: Model 3 input features example

- Model 4: Bi/Monolingual Dependency-Based model (BiMonoDep-WRC)
  In addition to the dependency-based monolingual WRC objective, and similar to the baseline, the dependency-based bilingual objective minimises the loss between sentence representation pairs. The Bag-of-Words representation for sentences is modified to include syntactic information by adding dependency features extracted from the sentence's dependency graph. The sentence's distributed representation is then formed by the mean of embeddings of all the sentence tokens (words and dependency features) in the bag. As the number of dependency features (twice the number of edges in the graph) is larger than the number of words in the sentence, a weighting scheme can be applied to balance their contribution in the representation [144]. Alternatively, we can represent each sentence with two separate feature bags, one for each token type, and form two aligned representations for each parallel sentence pair as shown in Figure 5.3.


Input sentence: The president resumed discussions
resumed
nsubj
dobj
president
discussions
det
The
Model 4: BiMonoDep-WRC-BilBOWA model

Monolingual objective inputs:

(target, context) pairs extracted from neighborhood of "resumed" node:

(resumed, discussions)

(president, resumed)

(president, discussions)

(resumed, nsubj_president)

(resumed, dobj_discussions)

(nsubj_president, dobj_discussions)

and all the reversed pairs of the above

Bilingual objective inputs:

(All Sentence BOW and dependency features)

words: the, president, resumed, discussions

dependency features: the_det, det^-1_president, nsubj_president, nsubj^-1_resumed, dobj^-1_resumed, dobj_discussions

Figure 5.3: Model 4 input features example

5.5 Implementation

We trained four different versions of the BilBOWA model for En-Es, En-De and En-Ar language pairs: Baseline-BilBOWA, MonoDep-WRC, MonoDep-WR and BiMonoDep-WRC models. We use the same code as [33] to train the models. Our implementation is based on the observation that the extended dependency skip-gram can be trained as a window-based skip-gram by an appropriate transformation of the input. For each context neighbourhood in the corpus we create two auxiliary sentences, one with the word context features and one with the dependency context features. Each sentence consists of all the tokens in the target word's neighbourhood in any order. Setting the window size larger than the length of the longest auxiliary sentence (or equivalently larger than the maximum degree of the dependency graphs in the corpus) results in creating all the positive pairs defined by the extended dependency skip-gram model. We note that no undesired pairs are created by having a large window because windows do not go across line breaks. We can create a Bag-of-Words sentence representation with dependency features for the bilingual objective by including all the dependency context features to the Bag-of-Words representation of the sentence. To implement the weighting scheme of [7] where word and dependency tokens are given equal weight, we instead form two aligned sentences per original sentence pair, one for each type of token. The models were trained with 200 dimensional word embeddings, with window size 35, and 15 negative samples for 5 epochs using stochastic gradient decent.

5.5.1 Dataset and Preprocessing

In this chapter, for En-Es and En-De languages pairs, we used Europarl-v7 and News Commentary-v6 monolingual and bilingual objectives respectively. We used a Single-labeled Arabic News Articles Dataset (SANAD) [148], which is a large Arabic dataset of textual data collected from three news portals, for Arabic monolingual training. WIT3, Web Inventory of Transcribed and Translated Talks, plain MSA Arabic and English language parallel corpus [140] were used for bilingual objective training. In all our experiments, the datasets used were tokenised (See Table 5.1), and lower-cased and empty lines were removed. For the dependency -based models, a dependency parser was used to parse all training datasets. Then, we extracted the dependency contexts from the parsed training datasets and used for monolingual and bilingual training. For parsing, we used a neural network based model for joint part-of-speech (POS) tagging and dependency parsing, introduced by [4][2] (For model structure see Figure 5.4). This model is an extension of the BIST graph-based dependency parser proposed by [149]. They incorporated BiLSTM-based tagging to predict POS tags for the parser automatically. We parsed the En, Es and De Europarl datasets and SANAD dataset for Ar language and used in the monolingual objective to train MonoDep-WRC and MonoDep-WC models. BiMonoDep, Europarl and News Commentary datasets were used for training the En-De and En-Es languages pairs and SANAD and WIT3 for Arabic with monolingual and bilingual objectives respectively. Parsing the datasets increased the number of features (tokens) dramatically as shown in Table 5.1. The increase happens due to multiple dependency features being extracted for each word.

[2] https://github.com/datquocnguyen/jPTDP

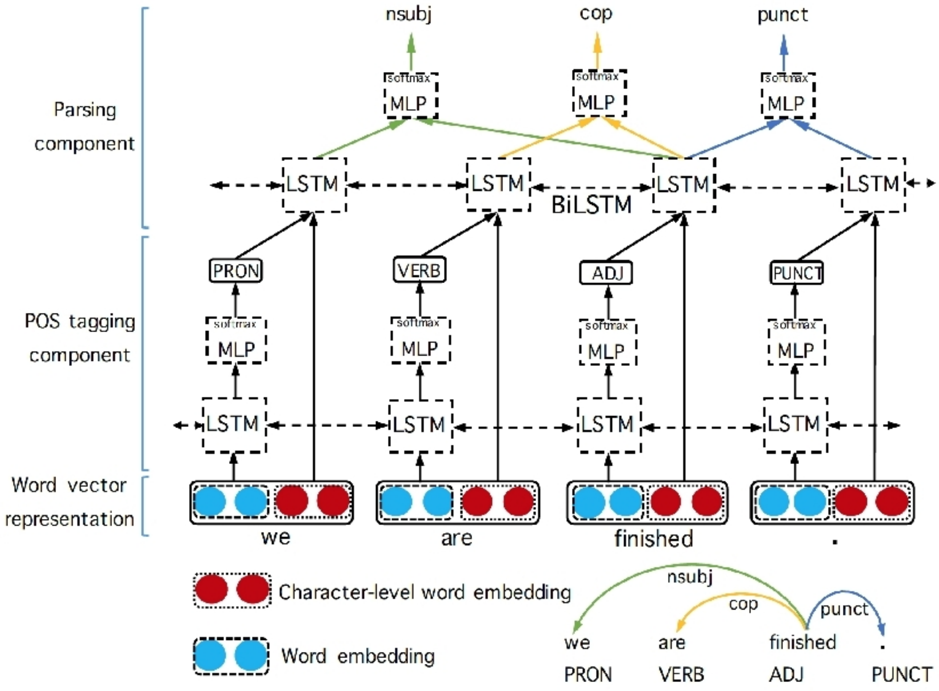


Figure 5.4: JPTDP model [4]

Table 5.1: Tokenised and Cleaned Large Datasets

Language pair	**Monolingual Dataset**				**Bilingual Dataset**		
En-Es	**Sentences**	**tokens**	**WRC tokens**	**WC tokens**	**Sentences**	**tokens**	**BiDep tokens**
en	1916071	51520106	301933100	150638284	132571	3280918	9406630
es	1916071	53804104	316022276	158002894	132571	3737853	10737580
En-De							
en	1879003	50896257	297861530	148930765	176850	4492424	13123596
de	1879003	48458495	283234958	141617479	176850	4547691	13289413
En-Ar							
en	1879003	50896257	297861530	148930765	135785	2692314	15882314
Ar	1766407	48066079	281330864	140665429	135785	1905410	11743904

Table 5.2: Precision at k on word-level translation task

Language pair	**100K**			**500K**			**2M**		
En-Es	**k=1**	**k=3**	**k=5**	**k=1**	**k=3**	**k=5**	**k=1**	**k=3**	**k=5**
BilBOWA-Baseline	**63.76**	**71.66**	**74.58**	63.82	76.22	75.84	63.54	76.42	78.74
MonoDep-WRC	61.14	69.90	72.36	**65.16**	**76.64**	**78.78**	**70.28**	**82.3**	**84.38**
MonoDep-WC	62.26	70.9	73.8	64.18	75.78	78.12	65.48	77.3	80
BiMonoDep-WRC	57.61	65.26	68.8	63.14	71.94	74.82	63.62	75.4	80.22
En-De	**k=1**	**k=3**	**k=5**	**k=1**	**k=3**	**k=5**	**k=1**	**k=3**	**k=5**
BilBOWA-Baseline	50.46	58.03	63.1	53.88	66.8	69.56	55.08	68.89	72.5
MonoDep-WRC	**51.62**	62.49	**66.78**	54.22	66.79	69.64	**57.44**	70.62	73.82
MonoDep-WC	51.62	**62.84**	65.56	**55.45**	**68.02**	**72.18**	57.42	**71.64**	**74.66**
BiMonoDep-WRC	50	59.52	62.60	54.82	66.16	70.9	57.09	70.14	73.26
En-Ar	**k=1**	**k=3**	**k=5**	**k=1**	**k=3**	**k=5**	**k=1**	**k=3**	**k=5**
BilBOWA-Baseline	29.64	39.64	41.4	34.52	44.72	49.1	38.98	51.88	55.84
MonoDep-WRC	**40.94**	50.1	**54.72**	**45.24**	**57.1**	**60.28**	**45.58**	58.6	62.8
MonoDep-WC	39.1	**51.28**	52.5	43.12	58.08	59.54	44.5	**60.82**	**63.38**
BiMonoDep-WRC	40.02	49.78	52.38	44.36	55.66	57.14	45.14	59.24	61.86

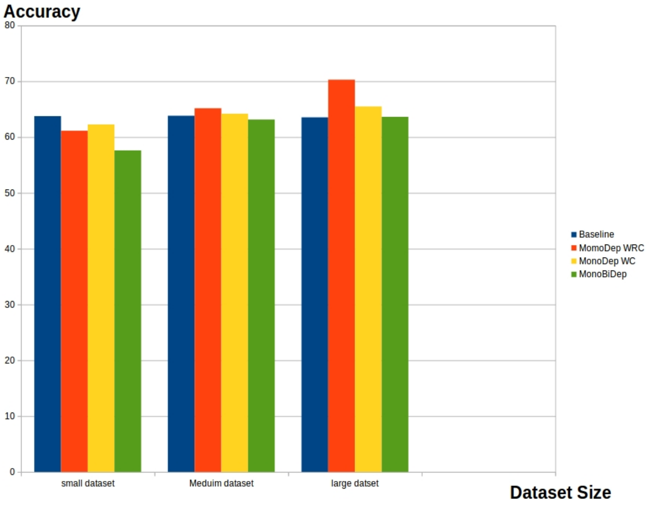


Figure 5.5: En-Es Results

5.6 Evaluation

Similarly to [33], as has been explained in Section 4.4, we used a word translation task, namely the Cross Language Dictionary Induction (CLDI) task, to evaluate the trained bilingual word embeddings using the same setting introduced by [24]. We created three test dataset pairs, for En-Es, En-De and En-Ar language pairs. Firstly, we extracted the most frequent 4,000 words from the training corpus for En-Es, En-De and SANAD Arabic datasets. Then we used the Google translator to translate the extracted words to form a dictionary for each language pair. Having these translation pairs (wl1,wl2), allow us to calculate the precision at k for word translation by finding target word wl2 in the nearest k neighbourhoods (1,3 and 5) to a given source word wl1 in the embedding space. Finally, we computed the mean precision from 10 runs, by selecting random 500 source words and their k (1, 3, and 5) nearest neighbours.

5.6.1 Results and Discussion

In these experiments, we compared the four different trained models using different dependency settings: BilBOWA-Baseline No Dependency features, MonoDep-WRC, MonoDep-WC and MonoBiDep-WRC dependency features. This comparison allowed us to investigate the effect of utilising dependency context features on languages pairs with different sentence structures and using different training datasets sizes at both monolingual and bilingual objectives. Our experiments demonstrate that language sentence structure differentiation affects the learning process differently for each training sitting as shown in Table 5.2. Based on the observed accuracy, our results cover two aspects:

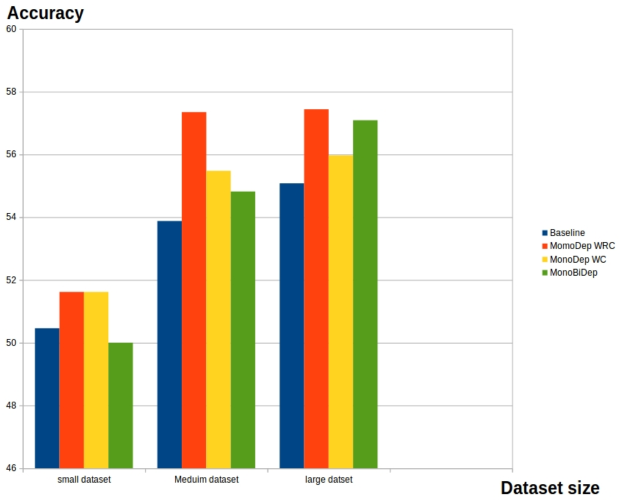


Figure 5.6: En-De Results

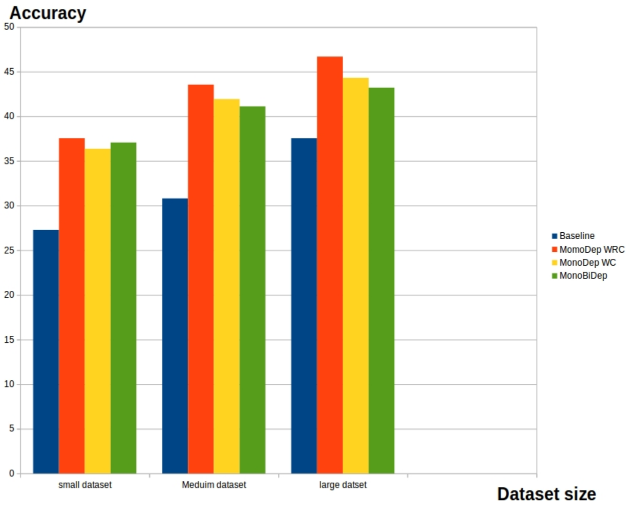


Figure 5.7: En-Ar Results

- Datasets size:
  Using small datasets shows variant effects on all three languages pairs. For En-Es, the baseline model outperformed the other three dependency-based models. While there is a slight benefit of the use of dependency features on En-De language pairs. For En-Ar, using these features improved the models performance dramatically for all dataset sizes. As incorporating syntax features with language pair that has language structure differentiation helps the model to improve the learning process. However, Trained models using medium and large training datasets show that, incorporating dependency-based features has improved the model performance to learn better word embeddings for all three languages pairs: En-De, En-Es and En-Ar (Figures 5.5, 5.6, and 5.7). Also, for En-Ar language pair, Figure 5.7 shows that MonoDep-WRC has out-

perform other dependency-based models in all dataset sizes experiments, while not the case in the other language pairs.

- Dependency Features:
  At the monolingual level, except in small dataset for En-Es language pair experiment, incorporating syntax features has a positive effect on the learning process. These dependency contexts lead to better learning of bilingual word embeddings in the CLDI task compared to the baseline BilBOWA model. In contrast, the BiMonoDep-BilBOWA model, that uses dependency features with monolingual and bilingual objectives, has demonstrated dramatic improvements in the En-Ar bilingual word embeddings models, slight improvements in En-De models and no improvement (or similar results to the BilBOWA baseline model) when using En-Es language pairs (See Table 5.2). Our experiments show that training models with language pairs which have different sentence structures produces most benefits when using dependency features. For the medium dataset, the accuracy was found to increase in the CLDI task by more than 10% points in En-Ar compared to the baseline, as shown in Figure 5.7 and 1.34%, 1.57% for En-Es and En-De respectively (See Figures 5.6, and 5.7).

5.7 Summary

This chapter aims to answer the second research question, which is " How would incorporating syntax features on the learning process of BWEs affect the model performance?". We compared four different BilBLOWA models using a range of contextual features: no dependency features BilBOWA baseline, dependency features at monolingual level word context and relation WCR and word context no relation WC and dependency features at both mono/bilingual levels MonoBil Dep. We used different language pairs with different language complexity levels: EN-Es , En-De and En-Ar pairs. Our results show that dataset size plays a core role in the learning process regardless the variant affects of the language differentiation. Moreover, by increasing the dataset size, dependency word embeddings at the monolingual level learned better bilingual word embeddings. This was found to improve the performance of word translation tasks across all three language pairs: En-Es, En-De and En-Ar in medium and large datasets compared to the baseline model. However, these features showed no improvement in the learning process of En-Es language pair using small datasets. Similarly, while the BiMonoDep features improved the performance on En-De and En-Ar language pairs, almost no impact on En-Es language pair was found. As a result, incorporating syntax features in model training leads to an improved learning process where languages pairs have different languages structure.

Chapter 6

Dependency-based Neural Machine Translation

In this chapter we transfer our trained dependency-based bilingual word embeddings from Chapter 5 into neural machine translation models. We show that the machine translation quality has been affected positevily by utilizating pretrained depedancy-based BWEs (MonoWRC and MonoWC BWEs) to train attention-based NMT models comparing to the baseline. This chapter is addressing the third research question which requires studying the effect of dependency-based BWEs on NMT. Therefore, improving MT quality is the contribution of this chapter.

6.1 Introduction

As has been mentioned earlier in Chapter 2, machine translation (MT) is one of the most important topics in the field of Natural Language Processing (NLP), and it has been studied in depth in the last few decades. Its goal is to use computers to translate real sentences from one language to another [150]. From parallel corpora, SMT learns word alignment and phrases. Despite its success, this method is unable to simulate long-distance dependencies between words, which has had an impact on translation quality. Because of the encouraging development made thus far, NMT has got even more attention and many models have been introduced as shown in Chapter 2. Unlike SMT, NMT uses an end-to-end model that trains the entire translation process using a single NN with a basic architecture and is capable of capturing long dependencies in the phrase. The discrete symbolic representation is utilised by SMT, whereas the continuous representation is utilised by NMT [150]. Word embedding is the term used to describe this continuous form. It shows its ability to capture the syntactic and semantics relation in different NLP tasks: semantics and question answering model [8, 9, 10], Machine translation [11, 12], and parsing [13, 14].

[151] state that syntactic analysis is playing a core role in understanding natural language. In this chapter, we experiment with the effects of integrating syntax features, mainly using our trained embeddings above dependency-based word embeddings to build NMT models. We trained a NMT-Keras attention-based model using two different language pairs: Es-En and De-En. We used these different languages to investigate how language differentiation can affect the learning process. Due to less research study the effect of pretrained BWEs on building NMT system, we aim in this chapter to investigate the impact of syntax-based BWEs on learning NMT. As translation can be modeled at different levels: words, sentence, paragraph and document levels, we focus on sentence level translation and other levels are left for future research. Thus, our contribution is improving the NMT quality by incorporating dependency features that have been learnt from Chapter 5.

6.2 Related Work

NMT models range from very simple network architecture to deep learning networks. A simple NMT model architecture consists of embedding layers, a classification layers, an encoder and decoder network. At sentence level translation, encoder-decoder NMT model can be viewed as a sequence-to-sequence model. This NMT model called autoregressive NMT [150]. In this model, the encoder encodes a source sentence into a fixed-length vector and the decoder generate the translation [152]. This is considered a limitation that affects the model performance. Therefore, many models have been introduced to solve this problem [152, 153]. The out-of-vocabulary (OOV) words is another limitation of using NMT that occurs due to the limited target vocabulary number. Jean et al. [153] address this problem Using an approach similar to those provided by [154], they replace generated out-of-vocaulary tokens with the corresponding source words.

Researchers have adopted autoregressive NMT model and have employed a recurrent neural network (RNN) to the encoder and the decoder to represent the source sentence and generate a target sentence respectively. Recurrent Neural Networks (RNNs) to handle the variable-length source and target sentences is used and a variety of RNNs, including the LSTM and GRU variations, have been implemented [155]. A significant improvement in SMT has been demonstrated using the sequence-to-sequence framework in conjunction with the combined attention mechanism. Many attention-based NMT models have been developed as a result [152, 19, 93]. Bahdanau et al. [152] proposed a soft-search model, which is a first an attention model, that translate and align jointly.

Later, Luong et al. [19] introduced two attention-based mechanisms: global and local attention models. The global approach considers all source words , and a local approach that only considers a selection of source words at a time. Their models improve the NMT performance dramatically compared to the basic NMT models. As one of the limitation of sequence to sequence NMT models is sentence length, research shows that incorporates syntactic information in the learning process has improve the NMT performance due to the long distance relations that can be obtained from using syntactic trees [156, 157, 158]. Chen et al. [159] incorporate source-side syntactic trees to improve NMT model performance. They introduce two NMT models: a bidirectional tree encoder and a tree-covarege model and their models improve the MT quality on Chinese-English. Similarliy, Zhang et al. [160] proposed a syntax-aware word representations (SAWRs) model that incorporate source-side implicitly. Then, to improve the fundamental NMT models, they simply concatenate SAWRs with ordinary word embeddings.

In neural network models for NLP applications, pre-trained word embeddings have proven to be very effective in text classification [22] and sequence tagging [161]. However, research show that it is much less common to use pretrained word embeddings in NMT. The existed work use monolingual word embeddings in NMT models. Qi et al. [162] proof that using pretrained word embeddings improve the translation quality. Their experiments show that a better encoding of the source sentence accounts for the majority of the gain from pretrained word embeddings. Some researchers show the effect of pretrained monolingual word embeddings [94, 163, 164]. Thus, in the next section, we study the effect of integrating dependancy-based BWEs in NMT by investigating their effect on machine translation quality from word to word to phrase to phrase translation.

6.3 Experiments

In our experiments we are using the pretrained dependency-based BWEs from Chapter 5. Our aim is to investigate transfer learning for BWEs infused with dependency in NMT models. This investigation allow us to identify strength and limitations of incorporating syntax features on learning NMT models. We built different NMT models with and without dependency features to compare as fallow: Baseline (Random Initilising the word embedding), MonoDep-WRC and MonoDep-WC NMT models. Based on the previous chapter's results, BioMono-WRC experiment is avoided.

Language pair	**En-De Dataset**		**En-Es Dataset**	
	En	**De**	**En**	**Es**
Training sentences	115231	115231	101688	101688
Training tokens	2128396	2121680	1979087	1794377
Dev sentences	2056	2056	1938	1938
Dev tokens	34961	35957	34539	32794
Test sentences	1399	1399	1344	1344
Test tokens	22654	24285	22328	22599

Table 6.1: Tokenised and cleaned large datasets

6.3.1 Tools and Datasets

There are several publicly available parallel corpora for MT. In this experiment, we use Europarl-v7 for En-De and En-Es language pairs [165]. En-Ar language pair due to the lack of resources has been excluded. For prepossessing, similarly to Chapter 5, all datasets have bees tokenised and clean by removing empty lines (See Table 6.1 for more details). To train the NMT-Keras, we use the attention-based model using pre-trained bilingual word embeddings [5] for En-De and En-Es language pairs. The embeddings have been trained on BilBOWA model (See Chapter 4 for more details) using dependency-parsed datasets [33] (See Chapter 5). These embeddings have been evaluated on word sense induction task and show an improvement on learning bilingual word embeddings by capturing the semantic proprieties of words as shown in Table 5.2.

6.3.2 Models

In this chapter, we use NMT-Keras, which is a flexible toolkit for neural machine translation. This tool is an extension of Keras library for deep learning [5]. It has been introduced to allow users to develop neural machine translation models using attention. NMT-Keras also can be applied to other problems including: image and video captioning, sentence classification and visual question answering.

6.3.3 Hyperparameters

In this experiment, we built NMT systems using NMT-Keras [5]. We used a pre-trained dependency-based BWEs: MonoWRC and MonoWC from Chapter 5 for Es-En and De-En language pairs. The primary justification for choosing these models is because they outperformed other proposed models. Due to resources limitations, the models were trained using 200 dimensions word embeddings. And Adam was the learning algorithm for all base NMT systems with learning rate 0.001.

6.3.4 Model Architecture

For Keras-NMT, due to the success of attention-based methods in MT, we used attentional NMT model developed by [5] (See Figure 6.1).

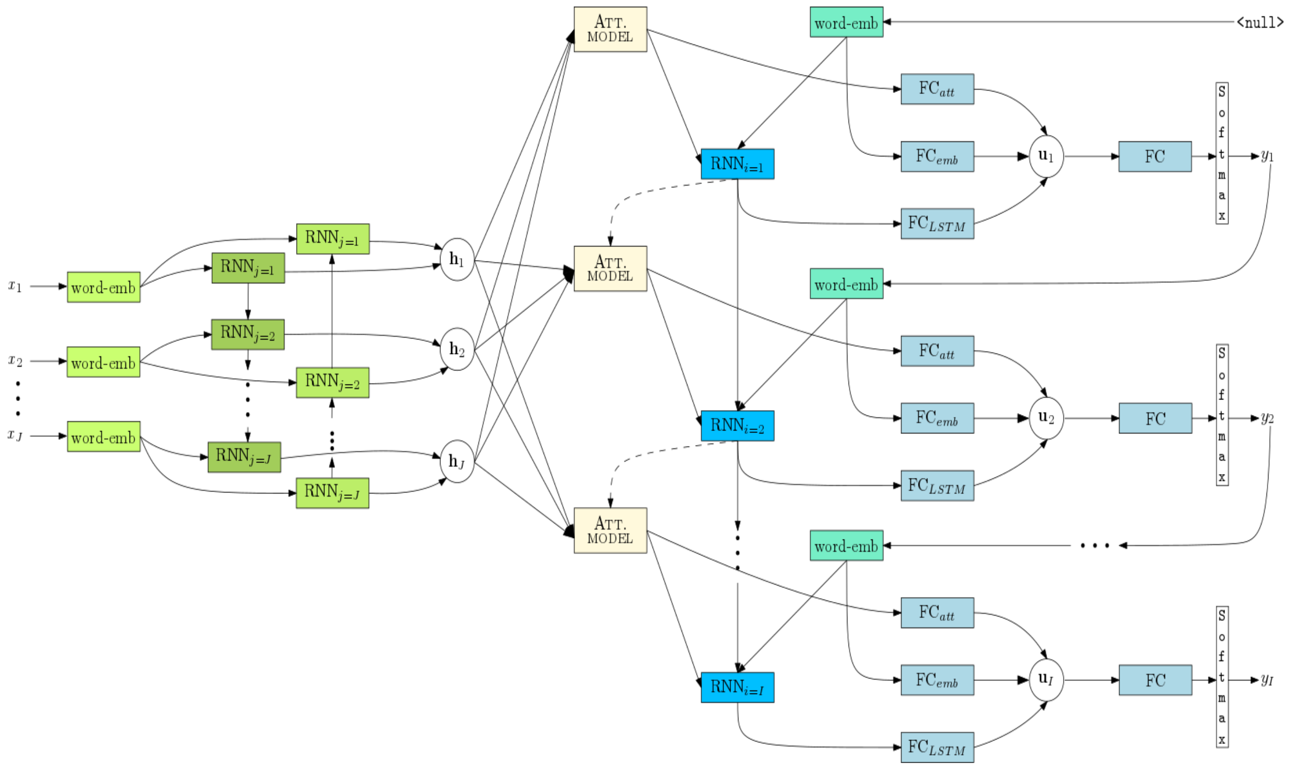


Figure 6.1: Neural Network with Attention Mechanism [5]

6.4 Results and Discussion

In this chapter, different metrics were used to assess the quality of the trained NMT systems: BLEU, CIDEr, METEOR, ROUGE_L and TER (See Section 2.5.2 for evaluation methods more details).

In all used evaluation scores, our results show that training NMT models using pretrained dependancy-based BWEs improve the phrase-based machine translation quality comparing to the baseline in both language pairs. However, BLEU and TER give a slight different results on En-De language pair. As has been mentioned above, in En-De results, the translation quality are vary form word-level translation (BLEU-1) to phrase-levels (BLEU-2,3 and 4). The baseline gives the best results at word-level translation (Best BLEU-1 score). At two words phrase long MT evaluation (BLEU-2) , MonoDep-WRC NMT model outperforms other models. While in longer phrases (3 and 4 words phrase), MonoDep-WC NMT model gives the best evaluation results on BLEU-3 and BLEU-4 scores. For En-ES language pairs, MonoDep-WRC NMT model outperforms other NMT models in all BLEUs scores (1,2,3 and 4) comparing to the baseline and MonoDep-WC NMT models. According to our results, utilising other evaluation methods: CIDEr ,METEOR, and ROUGE show that MonoDep-WRC NMT models improve the translation quality in both language pairs and outperform the base line as well as the MonoDep-WC NMT models (See Table 6.2). At the most common phrase level MT evaluation method BLEU-4, our results show that MonoDep-WC NMT model outperforms the baseline by 0.34% in En-De language pair. While for En-Es language pair, the MonoDep-WRC NMT model improved the translation quality up to 0.68% comparing to the baseline NMT model.

6.5 Summary

In this chapter, we used our pretrained syntax-based BWEs form Chapter 5 to investigate their effects on the quality of MT task. To answer our RQ3, we trained different NMT models using our trained dependency-based bilingual word embeddings. At pharse-level MT, our results show that using pretrained depedancy-based BWEs (MonoWRC and MonoWC BWEs) to train NMT-Keras (attention-based NMT) models has a positive impact on machine translation quality comparing to the baseline. In specific MonoDep-WRC BWEs in all evaluation methods scores apart of BLEU and TER, which give different outcomes on En-De language pair (See Table 6.2). Thus, this chapter contribute to improve the MT quality. En-Ar language pair has not been used in this work due to resources limitations. In addition, BiMonoDep BWEs have not been evaluated due to its results from the previous task.

Table 6.2: MT Evalutation

Language pair								
En-De	**Bleu_1**	**Bleu_2**	**Bleu_3**	**Bleu_4**	**CIDEr**	**METEOR**	**ROUGE_L**	**TER**
Baseline	**40.3**	24	15.1	9.86	0.87513	0.19187	0.3649	**0.795756**
MonoDep-WRC	40	**24.1**	15.3	9.94	**0.9385**	**0.2012**	**0.374598**	0.801
MonoDep-WC	40	24	**16**	**10.2**	0.902	0.197	0.3677	0.8012
En-Es	**Bleu_1**	**Bleu_2**	**Bleu_3**	**Bleu_4**	**CIDEr**	**METEOR**	**ROUGE_L**	**TER**
Baseline	48.257	32.2995	23.013	16.86	1.4721	0.4160	0.4409	0.6851
MonoDep-WRC	**49.1**	**33.254**	**23.865**	**17.5484**	**1.55012**	**0.431572**	**0.44853**	**0.67448**
MonoDep-WC	48.35	32.31	22.99	16.849	1.492	0.415568	0.441595	0.68334

Chapter 7

Conclusion

7.1 Thesis summary and contributions

In this thesis, we investigate the effects of different training settings (sentence-length and embedding size) in addition to different morphological segmentations on learning BWE for Ar-En language pairs. Also, the effect of integrating syntax features has been investigated. Our research has improved the BWEs and led to better MT quality. In Chapter 4 and 5, we used En-Ar language pair as to our knowledge, there is lack of research in this area using Arabic language.

In Chapter 4, to assess RQ1, our results proved that increasing the vector size allowed the models to capture more information and consequently learn better Es-En, De-En, and Ar-En BWEs and improves the accuracy. However, when we compared the effects of using short and long sentences, we discovered that using language pairs with distinct language structures has a different effect on the learning process. In all three segmentation approaches for Ar-En language pairs, long sentences (which increase the number of words "tokens") beat short sentences in 300D embeddings size models. On the other hand, short sentences only perform better when models are trained with 200D embeddings and the ATB segmentation scheme. Using long sentences to train the model in De-En language pairs shows that findings improve the accuracy across all trained BWEs models. However, training the BWEs models with a more similar language pair (Es-En) reveals that models that are trained on short sentences lead to better results. Long sentences with 300D embeddings allowed trained models to capture more information and acquire stronger bilingual word representations for more different sentence structure language pairs. A morphological segmentation approach is required for many Arabic NLP tasks. As shown in Chapter 4 segmentation also had a good effect since it led to the learning of better multilingual word embeddings. Moving from D0 (complete word form) to D3 (more segmentation, which increases the number of tokens in the training dataset) reduced the distance between Ar-En pairs and increased similarity significantly.

In Chapter 5, we employ a variety of dependence settings, including BilBOWA-Baseline No Dependency features, MonoDep-WRC, MonoDep-WC, and MonoBiDep-WRC (For more details see Chapter 6). We were able to explore the effect of integrating dependency context features on language pairings with varying sentence structures and different training dataset sizes for both monolingual and bilingual objectives. Our results answering RQ2 in this research by suggesting that language sentence structure differentiation influences the learning process differently for each training setting. When small datasets are used, variational effects are visible in all three language pairs. The baseline model performed better than the other three dependency-based models for En-Es. For En-De language pairs, there is a slight advantage to using dependence features. Using these features significantly enhanced the En-Ar model's performance.

Trained models utilising medium and large training datasets, on the other hand, reveal that including dependency-based features improves model performance in learning better word embeddings for all three language pairs: En-De, En-Es, and En-Ar. The BiMonoDep-BilBOWA model, has shown significant improvements in the En-Ar bilingual word embeddings models, minor improvements in En-De models, and no improvement (or similar results to the BilBOWA baseline model) when utilising En-Es language pairs.

When using dependency features, our findings indicate that training models using language pairs that have different sentence structures yields better results. Thus, to conclude, our findings show that more comparable language pairs, namely En-Es, are more likely to benefit from incorporating the dependency features (edges) in training datasets. However, language pairs with different sentence structures are gaining fewer benefits from using these features. And incorporating word contexts, that are related to the words in the syntactic consistently improved the learning process of BWEs.

In Chapter 6, to answer RQ3, we looked at how our dependancy-based BWEs that were trained in Chapter 5 affected the quality of MT tasks. We used the dependancy-based BWEs we had produced to train a variety of NMT models. When compared to the baseline, NMT-Keras (attention-based NMT) models trained with pre-trained BWEs (MonoWRC and MonoWC) improve the quality of machine translation. All evaluation methods apart from BLEU and TER yield different results on the En-De language pair in certain MonoDep-WRC BWEs (See Table 5.2). Thus, this chapter helps to increase the quality of the MT output.

This thesis makes a valuable contribution to the field of machine learning by investigating the factors that affect the learning process of BWEs for different language pairs with different language structure levels: En-Es, En-De and En-Ar. Further-more, this research improved the BWEs by incorporating syntax features into the learning process, and the trained BWEs are available for further studies. From a NMT point of view, our research proves that using pre-trained dependency-based BWEs has a positive effect on MT quality. The main limitation in this research is the lack of the Segmented Arabic Language Parsing dataset. This prevents us from examining the impact of dependency features on the NMT and BWE learning processes for the Arabic segmented dataset.

7.2 Future Work

By extending our experiments, there are many research opportunities for future work in the area of learning BWEs and NMT as following:

- As incorporating knowledge-based information (syntax features) leads to better results in machine translation task, integrating POS can also improve the learning process of BWEs and consequently improving NMT.
- Investigating the effect of incorporating dependency features and POS tags on languages with complex sentence structures (for example, Turkish, Polish, and Danish languages) needs more research.
- To investigate the effect of incorporating syntax features on the learning process of BWEs on Arabic , a dataset for Arabic language parsing must be generated.
- Investigating the effect of syntax features on:
  - Different NLP tasks such as name entity and question answering.
  - Other levels than word and sentence levels machine translation such as paragraph and document levels.
  - comparable dataset language pairs MT.

Bibliography

[1] P. Koehn, *Statistical Machine Translation.* New York, NY, USA: Cambridge University Press, 2010.

[2] H. Al-Haj and A. Lavie, "The impact of arabic morphological segmentation on broad-coverage english-to-arabic statistical machine translation," *Machine translation*, vol. 26(1-2), pp. 3–24, 2012.

[3] B. Zhang, D. Xiong, and J. Su, "Battrae: Bidimensional attention-based recursive autoencoders for learning bilingual phrase embeddings," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[4] D. Q. Nguyen and K. Verspoor, "An improved neural network model for joint pos tagging and dependency parsing," *arXiv preprint arXiv:1807.03955*, 2018.

[5] Á. Peris and F. Casacuberta, "NMT-Keras: a Very Flexible Toolkit with a Focus on Interactive NMT and Online Learning," *The Prague Bulletin of Mathematical Linguistics*, vol. 111, pp. 113–124, 2018. [Online]. Available: https://ufal.mff.cuni.cz/pbml/111/art-peris-casacuberta.pdf

[6] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the association for computational linguistics*, vol. 3, pp. 211–225, 2015.

[7] A. Komninos and S. Manandhar, "Dependency based embeddings for sentence classification tasks," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1490–1500.

[8] S. R. Bowman, C. Potts, and C. D. Manning, "Recursive neural networks can learn logical semantics," *arXiv preprint arXiv:1406.1827*, 2014.

[9] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," *Advances in neural information processing systems*, vol. 28, 2015.

[10] K. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[13] L. Kong, C. Dyer, and N. A. Smith, "Segmental recurrent neural networks," *arXiv preprint arXiv:1511.06018*, 2015.

[14] M. Lewis, K. Lee, and L. Zettlemoyer, "Lstm ccg parsing," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, 2016, pp. 221–231.

[15] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.

[16] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 234–239.

[17] Y. Shi, W. Zhang, J. Liu, and M. T. Johnson, "Rnn language model with word clustering and class-based output layer," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, pp. 1–7, 2013.

[18] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[19] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[20] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[21] C. N. Dos Santos and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts,” in *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, 2014, pp. 69–78.

[22] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.

[23] A. Severyn and A. Moschitti, “Twitter sentiment analysis with deep convolutional neural networks,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 959–962.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *Advances in neural information processing systems*, vol. 26, 2013.

[25] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation.” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[26] W. Ling, W. Dyer, A. Black, and I. Trancoso, “Two/too simple adaptations of word2vec for syntax problems,” in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 1299–1304. [Online]. Available: https://aclanthology.org/N15-1142

[27] A. Trask, P. Michalak, and J. Liu, “sense2vec - a fast and accurate method for word sense disambiguation in neural word embeddings,” *arXiv preprint arXiv:1511.06388*, 2015.

[28] L. Omer and G. Yoav, “Dependency-based word embeddings,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 302–308.

[29] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, “Semi-supervised recursive autoencoders for predicting sentiment distributions,” in *Proceedings of the 2011 conference on empirical methods in natural language processing*, 2011, pp. 151–161.

[30] T. Luong, H. Pham, and C. D. Manning, “Bilingual word representations with monolingual quality in mind,” in *Proceedings of the 1st workshop on vector space modeling for natural language processing*, 2015, pp. 151–159.

[31] J. Zhang, S. Liu, M. Li, M. Zhou, C. Zong *et al.*, "Bilingually-constrained phrase embeddings for machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 111–121.

[32] J. Su, D. Xiong, B. Zhang, Y. Liu, J. Yao, and M. Zhang, "Bilingual correspondence recursive autoencoder for statistical machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1248–1258.

[33] S. Gouws, Y. Bengio, and G. Corrado, "Bilbowa: Fast bilingual distributed representations without word alignments," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 748–756.

[34] J. Coulmance, J.-M. Marty, G. Wenzek, and A. Benhalloum, "Trans-gram, fast cross-lingual word embeddings," *arXiv preprint arXiv:1601.02502*, 2016.

[35] I. Vulic and M. Moens, "Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, vol. 2. ACL; East Stroudsburg, PA, 2015, pp. 719–725.

[36] A. Mogadala and A. Rettinger, "Bilingual word embeddings from parallel and non-parallel corpora for cross-language text classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, 2016, pp. 692–702.

[37] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *towards data science*, vol. 6, no. 12, pp. 310–316, 2017.

[38] H. Peng, J. Li, Y. Song, and Y. Liu, "Incrementally learning the hierarchical softmax function for neural language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[39] F. Torregrossa, R. Allesiardo, V. Claveau, N. Kooli, and G. Gravier, "A survey on training and evaluation of word embeddings," *International Journal of Data Science and Analytics*, vol. 11, pp. 85–103, 2021.

[40] S. Ruder, I. Vulić, and A. Søgaard, "A survey of cross-lingual embedding models," *Journal of Artificial Intelligence Research*, vol. 65, pp. 569–631, 2019.

[41] M. Xiao and Y. Guo, "Distributed word representation learning for cross-lingual dependency parsing," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 2014, pp. 119–129.

[42] K. Hermann and P. Blunsom, "The role of syntax in vector space models of compositional semantics," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013, pp. 187–197.

[43] S. Lauly, A. Boulanger, and H. Larochelle, "Learning multilingual word representations using a bag-of-words autoencoder," *arXiv preprint arXiv:1401.1803*, 2014.

[44] W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning, "Bilingual word embeddings for phrase-based machine translation," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1393–1398.

[45] M. Alawneh, N. Omar, T. Sembok, S. Wiwatwithaya, P. Phasukkit, S. Tungjitkusolmun, M. Sangworasilp, C. Pintuviroj, S. Parvaresh, A. Ayatollahi *et al.*, "Machine translation from english to arabic," *Heart*, vol. 40, p. 9.

[46] O. Shirko, N. Omar, H. Arshad, and M. Albared, "Machine translation of noun phrases from arabic to english using transfer-based approach." *Journal of Computer Science*, vol. 6(3), no. 3, pp. 350–356, 2010.

[47] J. Hutchins, "The development and use of machine translation systems and computer-based translation tools," *International Journal of Translation*, vol. 15, no. 1, pp. 5–26, 2003.

[48] M. Aref, M. Al-Mulhem, and H. Al-Muhtaseb, "English to arabic machine translation: A critical review and suggestions for development," 1995, pp. 421–427.

[49] J. Slocum, "A survey of machine translation: Its history, current status, and future prospects," *Computational Linguistics*, vol. 11, no. 1, pp. 1–17, 1985.

[50] J. Hutchins, "Current commercial machine translation systems and computer-based translation tools: System types and their uses," *International Journal of Translation*, vol. 17, no. 1-2, pp. 5–38, 2005.

[51] M. Okpor, "Machine translation approaches: issues and challenges," *International Journal of Computer Science Issues (IJCSI)*, vol. 11, no. 5, p. 159, 2014.

[52] C. Parsing, “Speech and language processing,” 2009.

[53] A. Garg and M. Agarwal, “Machine translation: a literature review,” *arXiv preprint arXiv:1901.01122*, 2018.

[54] M. Nagao, “A framework of a mechanical translation between japanese and english by analogy principle,” *Artificial and human intelligence*, pp. 351–354, 1984.

[55] H. Somers, “Review article: Example-based machine translation,” *Machine Translation*, vol. 14 (2), pp. 113–157, 1999.

[56] S. Tripathi and J. Sarkhel, “Approaches to machine translation,” *Annals of library and information studies*, vol. 57 (4), pp. 388–393, 2010.

[57] J. Carbonell, S. Klein, D. Miller, M. Steinbaum, T. Grassiany, and J. Frey, “Context-based machine translation,” in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, 2006, pp. 19–28.

[58] W. Weaver, “Translation,” *Machine Translation of Languages*, vol. 14, pp. 15–23, 1955.

[59] M. F. Alawneh and T. M. Sembok, “Rule-based and example-based machine translation from english to arabic,” in *2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications*. IEEE, 2011, pp. 343–347.

[60] M. Alawneh and T. Mohd, “Handling agreement and words reordering in machine translation from english to arabic using hybrid-based systems,” *Journal of Computer Science*, vol. 11, no. 6, pp. 93–97, 2011.

[61] H. Sawaf, “Arabic dialect handling in hybrid machine translation,” in *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), Denver, Colorado*, 2010.

[62] A. Berger, P. Brown, S. Della Pietra, V. Della Pietra, J. Gillett, J. Lafferty, R. Mercer, H. Printz, and L. Ure, “The candide system for machine translation,” in *Proceedings of the workshop on Human Language Technology*, 1994.

[63] P. Koehn, F. Och, and D. Marcu, “Statistical phrase-based translation,” in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 48–54.

[64] G. Neubig, T. Watanabe, E. Sumita, S. Mori, and T. Kawahara, "An unsupervised model for joint phrase alignment and extraction," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1.* Association for Computational Linguistics, 2011, pp. 632–641.

[65] P. Brown, V. Pietra, S. Pietra, and R. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational linguistics*, vol. 19 (2), pp. 263–311, 1993.

[66] R. Zens, F. Och, and H. Ney, "Phrase-based statistical machine translation," in *Annual Conference on Artificial Intelligence.* Springer, 2002, pp. 18–32.

[67] D. Marcu and W. Wong, "A phrase-based, joint probability model for statistical machine translation," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 2002, pp. 133–139.

[68] D. Chiang, "Hierarchical phrase-based translation," *computational linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[69] T. Khemakhem, S. Jamoussi, and A. Hamadou, "The miracl arabic-english statistical machine translation system for iwslt 2010," in *Proceedings of the 7th International Workshop on Spoken Language Translation: Evaluation Campaign*, 2010, pp. 119–125.

[70] V. Van Nguyen, T. Nguyen, M. Le Nguyen, and A. Shimazu, "A model lexicalized hierarchical reordering for phrase based translation," *Procedia-Social and Behavioral Sciences*, vol. 27, pp. 77–85, 2011.

[71] P. Koehn and H. Hoang, "Factored translation models," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 868–876.

[72] C. Tillmann, S. Vogel, H. Ney, and A. Zubiaga, "A dp-based search using monotone alignments in statistical translation," in *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, 1997, pp. 289–296.

[73] I. Youssef, M. Sakr, and M. Kouta, "Linguistic factors in statistical machine translation involving arabic language," *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, pp. 154–159, 2009.

[74] D. Chiang, “A hierarchical phrase-based model for statistical machine translation,” in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, 2005, pp. 263–270.

[75] H. Hoang, P. Koehn, and A. Lopez, “A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation,” in *Proceedings of the 6th International Workshop on Spoken Language Translation: Papers, IWSLT*, 2009.

[76] E. Charniak, K. Knight, and K. Yamada, “Syntax-based language models for statistical machine translation,” in *Proceedings of MT Summit IX*, 2003.

[77] F. j. Och and H. Ney, “A systematic comparison of various statistical alignment models,” *Computational Linguistics*, vol. 1 (29), pp. 19–51, 2003.

[78] Z. Li, C. Callison-Burch, C. Dyer, J. Ganitkevitch, L. Khudanpur, S.and Schwartz, W. Thornton, J. Weese, and O. Zaidan, “Joshua: An open source toolkit for parsing-based machine translation,” in *Proceedings of the Fourth Workshop on Statistical Machine Translation.* Association for Computational Linguistics, 2009, pp. 135–139.

[79] P. Koehn, “Noun phrase translation,” Ph.D. dissertation, University of Southern California, 2003.

[80] P. Koen, “Pharaoh: a beam search decoder for phrase-based statistical machine translation models,” in *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Washington, USA: Springer, Sep. 28 - Oct. 2 2004, pp. 115–124. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-30194-3_13

[81] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, “Moses: Open source toolkit for statistical machine translation,” in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions.* Association for Computational Linguistics, 2007, pp. 177–180.

[82] L. R. Bahl, F. Jelinek, and R. L. Mercer, “A maximum likelihood approach to continuous speech recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 2, pp. 179–190, 1983.

[83] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, “A statistical approach to machine translation,” *Computational linguistics*, vol. 16, pp. 79–85, 1990.

[84] E. Mays, F. J. Damerau, and R. L. Mercer, "Context-based spelling correction," *Information Processing & Management*, vol. 27, pp. 517–522, 1991.

[85] M. Federico and M. Cettolo, "Efficient handling of n-gram language models for statistical machine translation," in *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007, pp. 88–95.

[86] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *Seventh international conference on spoken language processing*, 2002.

[87] M. Federico, N. Bertoldi, and M. Cettolo, "Irstlm: An open source toolkit for handling large scale language models," in *Ninth Annual Conference of the International Speech Communication Association (Interspeech)*, 2008.

[88] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 2011, pp. 187–197.

[89] Q. Gao and S. Vogel, "Parallel implementations of word alignment tool," in *Software Engineering, Testing, and Quality Assurance for Natural Language Processing.* Association for Computational Linguistics, 2008, pp. 49–57.

[90] J. Zhang and C. Zong, "Deep neural networks in machine translation: An overview," *IEEE Intelligent Systems*, 2015.

[91] S. Liu, N. Yang, M. Li, and M. Zhou, "A recursive recurrent neural network for statistical machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 1491–1500.

[92] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[93] J. Niehues, E. Cho, T. Ha, and A. Waibel, "Pre-translation for neural machine translation," in *COLING*, 2016.

[94] W. He, Z. He, H. Wu, and H. Wang, "Improved neural machine translation with smt features," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[95] X. Wang, Z. Lu, Z. Tu, H. Li, D. Xiong, and M. Zhang, "Neural machine translation advised by statistical machine translation," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[96] M. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," *arXiv preprint arXiv:1410.8206 CoRR*, 2014.

[97] D. Xiong, Q. Liu, and S. Lin, "Maximum entropy based phrase reordering model for statistical machine translation," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006, pp. 521–528.

[98] P. Li, Y. Liu, and M. Sun, "Recursive autoencoders for itg-based translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 567–577.

[99] M. Sundermeyer, T. Alkhouli, J. Wuebker, and H. Ney, "Translation modeling with bidirectional recurrent neural networks," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 14–25. [Online]. Available: https://aclanthology.org/D14-1003

[100] K. W. Church and E. H. Hovy, "Good applications for crummy machine translation," *Machine Translation*, vol. 8, no. 4, pp. 239–258, 1993.

[101] A. Lommel, M. Popovic, and A. Burchardt, "Assessing inter-annotator agreement for translation error annotation," in *MTE: Workshop on Automatic and Manual Metrics for Operational Translation Evaluation*. Language Resources and Evaluation Conference Reykjavik, 2014, pp. 31–37.

[102] X. Song, T. Cohn, and L. Specia, "Bleu deconstructed: Designing a better mt evaluation metric," *Int. J. Comput. Linguistics Appl.*, vol. 4, no. 2, pp. 29–44, 2013.

[103] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf, "Accelerated dp based search for statistical translation." in *Eurospeech*. Citeseer, 1997.

[104] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, Massachusetts, USA: Association for Machine Translation in the Americas, Aug. 2006, pp. 223–231. [Online]. Available: https://aclanthology.org/2006.amta-papers.25

[105] B. Babych and A. Hartley, "Extending the bleu mt evaluation method with frequency weightings," in *Proceedings of the 42nd Annual Meeting on Association*

for Computational Linguistics. Association for Computational Linguistics, 2004, pp. 621–628.

[106] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “Bleu: A method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002, pp. 311–318.

[107] P. Koehn, A. Axelrod, A. Mayne, C. Callison-Burch, M. Osborne, D. Talbot, and M. White, “Edinburgh system description for the 2005 nist mt evaluation,” in *Proceedings of Machine Translation Evaluation Workshop*, 2005.

[108] S. Banerjee and A. Lavie, “Meteor: An automatic metric for mt evaluation with improved correlation with human judgments,” in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[109] A. Lavie and A. Agarwal, “Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments,” *Proceedings of the Second Workshop on Statistical Machine Translation WMT-08*, 2007.

[110] C. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text summarization branches out*, 2004, pp. 74–81.

[111] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, “Cider: Consensus-based image description evaluation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[112] M. Zhang, “A survey of syntactic-semantic parsing based on constituent and dependency structures,” *Science China Technological Sciences*, vol. 63, no. 10, pp. 1898–1920, 2020.

[113] K. Yamada and K. Knight, “A syntax-based statistical translation model,” in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, 2001, pp. 523–530.

[114] Y. S. Chan and D. Roth, “Exploiting syntactico-semantic structures for relation extraction,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 551–560.

[115] H. Zou, X. Tang, B. Xie, and B. Liu, “Sentiment classification using machine learning techniques with syntax features,” in *2015 International Conference on Computational Science and Computational Intelligence (CSCI).* IEEE, 2015, pp. 175–179.

[116] S. H. Unger, “A global parser for context-free phrase structure grammars,” *Communications of the ACM*, vol. 11, no. 4, pp. 240–247, 1968.

[117] J. Armengol-Estapé and M. R. Costa-jussà, “Semantic and syntactic information for neural machine translation,” *Machine Translation*, vol. 35, no. 1, pp. 3–17, 2021.

[118] N. Habash, “Introduction to arabic natural language processing,” *Synthesis Lectures on Human Language Technologies*, vol. 3, no. 1, pp. 1–187, 2010.

[119] G. Mosa and A. Ali, “Arabic phoneme recognition using hierarchical neural fuzzy petri net and lpc feature extraction,” *Signal Processing: An International Journal (SPIJ)*, vol. 3, no. 5, p. 161, 2009.

[120] A. Hatem, N. Omar, and K. Shaker, “Morphological analysis for rule-based machine translation,” in *Semantic Technology and Information Retrieval (STAIR), 2011 International Conference on*, June 2011, pp. 260–263.

[121] A. Monem, K. Shaalan, A. Rafea, and H. Baraka, “Generating arabic text in multilingual speech-to-speech machine translation framework,” *Machine translation*, vol. 22, no. 4, pp. 205–258, 2008.

[122] J. Elming and N. Habash, “Syntactic reordering for english-arabic phrase-based machine translation,” in *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages.* Association for Computational Linguistics, 2009, pp. 69–77.

[123] S. Hilder, B. Theobald, and R. Harvey, “In pursuit of visemes,” in *Auditory-Visual Speech Processing 2010 (AVSP)*, 2010, pp. 8–2.

[124] P. Damien, “Visual speech recognition of modern classic arabic language,” in *Humanities, Science & Engineering Research (SHUSER), 2011 International Symposium on.* IEEE, 2011, pp. 50–55.

[125] S. Alkuhlani and N. Habash, “A corpus for modeling morpho-syntactic agreement in arabic: Gender, number and rationality,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 357–362.

[126] M. Akeel and R. B Mishra, “A statistical method for english to arabic machine translation,” *International Journal of Computer Applications*, vol. 86(2), pp. 13–19, 2014.

[127] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 8(4), p. 14, 2009.

[128] M. Shquier and O. Shqeer, "Hybrid-based approach to handle irregular verb-subject agreements in english-arabic machine translation," in *The proceeding of International conference on soft computing and soft engineering*, 2013.

[129] A. El Kholy and N. Habash, "Orthographic and morphological processing for english-arabic statistical machine translation," *Machine Translation*, vol. 26, pp. 25–45, 2012.

[130] A. Soudi, G. Neumann, and A. Van den Bosch, "Arabic computational morphology: Knowledge-based and empirical methods," in *Arabic computational morphology.* Springer, 2007, pp. 3–14.

[131] N. Habash and O. Rambow, "Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005.

[132] N. Habash, O. Rambow, and R. Roth, "Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization," in *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt*, 2009.

[133] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for Arabic," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations.* San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 11–16. [Online]. Available: https://www.aclweb.org/anthology/N16-3003

[134] M. Boudchiche, A. Mazroui, M. O. Bebah, A. Lakhouaja, and A. Boudlal, "Alkhalil morpho sys 2: A robust arabic morpho-syntactic analyzer," *Journal of King Saud University – Computer and Information Sciences*, vol. 29, no. 2, pp. 141–146, April 2017.

[135] A. Pasha, M. Al-Badrashiny, M. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth, "Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic," in *Proceedings of the Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland*, 2014, pp. 1094–1101.

[136] N. Habash and F. Sadat, “Arabic preprocessing schemes for statistical machine translation,” 2006.

[137] I. Badr, R. Zbib, and J. Glass, “Segmentation for english-to-arabic statistical machine translation,” in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, 2008, pp. 153–156.

[138] T. Alqaisi and S. O’Keefe, “En-ar bilingual word embeddings without word alignment: Factors effects,” in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 97–107. [Online]. Available: https://aclanthology.org/W19-4611

[139] T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting similarities among languages for machine translation,” *CoRR*, vol. abs/1309.4168, 2013. [Online]. Available: http://arxiv.org/abs/1309.4168

[140] C. Girardi, “Wit3: Web inventory of transcribed and translated talks,” 03 2012. [Online]. Available: https://wit3.fbk.eu/mt.php?release=2012-02-plain

[141] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[142] C. N. Dos Santos and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts,” in *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, 2014, pp. 69–78.

[143] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, “Parsing with compositional vector grammers,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 455–465.

[144] L. Xu, W. Ouyang, X. Ren, Y. Wangand, and L. Jiang, “Enhancing semantic representations of bilingual word embeddings with syntactic dependencies,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18),pp. 4518-4524*, 2018.

[145] T. Alqaisi, A. Komninos, and S. O’Keefe, “Dependency based bilingual word embeddings without word alignment,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–6.

[146] C. Li, J. Li, Y. Song, and Z. Lin, "Training and evaluating improved dependency-based word embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[147] M. de Marneffe, C. D. Manning, J. Nivre, and D. Zeman, "Universal Dependencies," *Computational Linguistics*, vol. 47, no. 2, pp. 255–308, Jun. 2021. [Online]. Available: https://aclanthology.org/2021.cl-2.11

[148] O. Einea, A. Elnagar, and R. Al Debsi, "Sanad: Single-label arabic news articles dataset for automatic text categorization," *Data in Brief*, vol. 25, 2019.

[149] E. Kiperwasser and Y. Goldberg, "Simple and accurate dependency parsing using bidirectional lstm feature representations," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 313–327, 2016.

[150] Z. Tan, S. Wang, Z. Yang, G. Chen, X. Huang, M. Sun, and Y. Liu, "Neural machine translation: A review of methods, resources, and tools," *AI Open*, vol. 1, pp. 5–21, 2020.

[151] T. Limisiewicz and D. Marecek, "Syntax representation in word embeddings and neural networks - A survey," *CoRR*, vol. abs/2010.01063, 2020. [Online]. Available: https://arxiv.org/abs/2010.01063

[152] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *3rd International Conference on Learning Representations*, 2015.

[153] S. Jean, O. Firat, K. Cho, R. Memisevic, and Y. Bengio, "Montreal neural machine translation systems for wmt," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 134–140. [Online]. Available: https://aclanthology.org/W15-3014

[154] T. Luong, I. Sutskever, Q. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 11–19. [Online]. Available: https://aclanthology.org/P15-1002

[155] A. A. Verma and P. Bhattacharyya, "Literature survey: Neural machine translation," *CFILT, Indian Institute of Technology Bombay, India*, 2017.

[156] P. Williams, R. Sennrich, M. Post, and P. Koehn, *Syntax-based Statistical Machine Translation*, 4th ed., ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, Aug. 2016, vol. 9, pp. 1–208.

[157] X. Shi, I. Padhi, and K. Knight, "Does string-based neural MT learn source syntax?" in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.* Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1526–1534. [Online]. Available: https://www.aclweb.org/anthology/D16-1159

[158] J. Li, D. Xiong, Z. Tu, M. Zhu, M. Zhang, and G. Zhou, "Modeling source syntax for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 688–697. [Online]. Available: https://aclanthology.org/P17-1064

[159] H. Chen, S. Huang, D. Chiang, and J. Chen, "Improved neural machine translation with a syntax-aware encoder and decoder," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1936–1945. [Online]. Available: https://aclanthology.org/P17-1177

[160] M. Zhang, Z. Li, G. Fu, and M. Zhang, "Syntax-enhanced neural machine translation with syntax-aware word representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 1151–1161. [Online]. Available: https://aclanthology.org/N19-1118

[161] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 260–270. [Online]. Available: https://aclanthology.org/N16-1030

[162] Y. Qi, D. Sachan, M. Felix, S. Padmanabhan, and G. Neubig, "When and why are pre-trained word embeddings useful for neural machine translation?" in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 529–535. [Online]. Available: https://aclanthology.org/N18-2084

[163] Y. Cheng, W. Xu, Z. He, W. He, H. Wu, M. Sun, and Y. Liu, "Semi-supervised learning for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1965–1974. [Online]. Available: https://aclanthology.org/P16-1185

[164] P. Ramachandran, P. J. Liu, and Q. V. Le, "Unsupervised pretraining for sequence to sequence learning," 2016. [Online]. Available: https://arxiv.org/abs/1611.02683

[165] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of Machine Translation Summit X: Papers*, Phuket, Thailand, Sep. 13-15 2005, pp. 79–86. [Online]. Available: https://aclanthology.org/2005.mtsummit-papers.11