Karlsruhe Institute of Technology

Homogenization of Arabic Corpora for Machine Translation

Student Project of

Mohamed Yassine Khelifi

At the Department of Informatics
Interactive Systems Lab (ISL)
Institute for Anthropomatics and Robotics

Reviewer:	Prof. Dr. Alexander Waibel
Advisor:	M.Sc. Mohammed Mediani

Duration: 10 April 2015 – 09 July 2015

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 07.01.2015

.....

(**Mohamed Yassine Khelifi**)

Abstract

It has always been a challenge for the statistical machine translation to deal with translations of corpora produced from the crowd. On the one hand such corpora are cheap and easy to acquire, on the other hand they contain a lot of inconsistencies and spelling mistakes. In the TED project [1] for instance there are in average about 200 translators per language to translate transcribed English speech. The Arabic corpora, have been translated by about 547 different translators. Depending on the dialect and linguistic knowledge of the individual translators the Arabic corpora sometimes show different spelling for the same word.

In this project we try to decrease the degree of diversity of word spelling in order to optimize the English-Arabic Machine Translation. Our goal is to detect correct alternative spellings for the same word in order to unify word spellings in the Arabic training corpora. Therefore, we first look for Arabic words with minimal spelling difference but carrying the same meaning. Then, we classify each pair or group of these words in a cluster. For detecting similar words in the Arabic training corpus, we propose an unsupervised approach, which applies different models to a word candidate list in a cascade manner. This word candidate list contains the words with highest lexical and semantic similarity. The application of each model returns a subset of the highest scoring candidates.

We chose the levenshtein ratio and the ngram of characters perplexity as lexical similarity measures. The first method measures only the edit distance of a word pair. However the second one trains a language model with the character n-grams from an Arabic corpus assumed to be 100% correct (development set of our SMT system). Then, applying that language model on the character n-grams from the training set, gives us a clue about the likelihood of the letter sequence in each word. Comparing the perplexity score of two words helps us measure their irregularity and eventually detect misspellings.

For the semantic similarity measure we apply the word2vec and the English-Arabic bilingual association score. While the first method learns semantic rules only from the Arabic training corpus, the second one involves the English training corpus. The English-Arabic bilingual association score extracts the semantic relationships from the English $\longleftrightarrow$ Arabic word alignments.

After our unsupervised approach based on finding the words with highest lexical and semantic similarity, we select a primary list of word pair candidates, which contains in addition to words that are relevant for the Arabic corpus homogenization, the normal word inflections. From the word pairs in the primary list we learn edit rules (insertions,

[1] see 4.1 and https://www.ted.com

deletions and substitutions) corresponding to each letter and use these statistics to reduce the size of the primary list of word pair candidates. Similarly to a normal edit distance calculation we try to modify the shortest word in the word pair into the other one. Then we select for the secondary list only the word pairs with the least common edit operations. Finally, we use the relative difference between the character n-gram perplexity measures of each word pair to rank the secondary list. We consider the words with highest relative perplexity difference for our Arabic corpus homogenization and for each word pair from the secondary list we consider the word with lowest perplexity as correct. In order to measure the efficiency of our approach we compare the perplexity of the Arabic corpus before and after the homogenization. In addition we measure the influence of the homogenization on the performance of our baseline SMT system in terms of BLEU score.

Zusammenfassung

Mit den aus dem Internet angesammelten Übersetzungen zu handeln, ist bisher immer eine große Herausforderung für die statistische maschinelle Übersetzung gewesen. Solche Übersetzungen sind einerseits leicht und kostengünstig zu akquirieren, aber andererseits beinhalten sie auch viele Inkonsistenzen und Rechtschreibungsfehler. Beim TED Projekt [2] gibt es zum Beispiel durchschnittlich ca. 200 ehrenamtliche Übersetzer pro Sprache zum Übersetzen von englischen Texten. Die arabische Korpora wurden beispielsweise von ca. 547 unterschiedliche Übersetzer übersetzt. Abhängig vom Dialekt und Sprachkenntnisse der einzelnen Übersetzern sind ab und zu unterschiedliche Schreibweisen für ein einziges Wort zu finden.

Bei diesem Projekt wollen wir die Englisch-Arabisch maschinelle Übersetzung optimieren, indem wir die Diversität der Wörterschreibweisen im arabische Korpus sinken. Unser Ziel ist die von einander abweichende alternative Schreibweisen desselben Wortes zusammen in einem Cluster zu klassifizieren. Um dies zu realisieren, suchen wir zuerst die arabische Wörter mit der gleichen Bedeutung und mit minimalem Unterschied in der Rechtschreibung. Zunächst klassifizieren wir jedes Wortpaar oder Wörtermenge von diesen Wörter in einem Cluster.

Um die zu einander ähnlichen Wörter zu herauszufinden, schlagen wir ein nicht überwachten Verfahren vor. Dieses Verfahren wendet unterschiedliche Modelle auf einer primären Liste mit Wortpaarkandidaten an. Diese Liste beinhaltet Wörter mit höhen lexikalische und semantische Ähnlichkeit. Die Anwendung jedes Modells liefert eine Untermenge von der primären List der Wortpaarkandidaten. Diese Untermenge selektiert nur die Wörter mit der höchsten lexikalischen und semantischen Ähnlichkeit.

Als Maß für die lexikalische Ähnlichkeit, wählen wir das levenshtein ratio und die Buchstaben n-gram Perplexität aus. Die erste Methode berechnet die Editierdistanz zwischen den zwei Wörter eines Wortpaares. Die zweite Methode trainiert ein Sprachmodell mit den Buchstaben n-grams von einem arabischen Korpus, was man als 100% korrekt bezeichnet (vom development set unseres SMT Systems). Dann die Anwendung dieses Sprachmodells auf die Buchstaben n-grams des training sets, beschafft uns Informationen über die Wahrscheinlichkeit des Vorkommens der Buchstabensequenz jedes Wortes. Der Vergleich zwischen den Perplexitäten von zwei Wörter ermöglicht uns ihre Irregularität zu messen und möglicherweise auch falsche Schreibweisen zu entdecken.

Für die Messung der semantischen Ähnlichkeit, wenden wir word2vec und Englisch-Arabisch bilinguales Assoziationsscore. Während word2vec die semantischen Regeln ausschließlich

[2] siehe 4.1 und https://www.ted.com

aus dem arabischen training Korpus lernt, involviert die Messung von Englisch-Arabisch bilinguales Assoziationsscore den englische Korpus. Da, werden die semantischen Beziehungen vom Englisch $\longleftrightarrow$ Arabisch Wörter Anordnung abgeleitet.

Nach unserem nicht überwachten Vorgehen basierend auf das Finden von Wörter mit höheren lexikalischen und semantischen Ähnlichkeit, selektieren wir eine primäre Liste mit Wortpaarkandidaten. Diese Liste beinhaltet zusätzlich zu den Wörter, die für Homogenisierung des arabischen Korpus sind, normale Wörterflexionen. Von den Wortpaaren in der primären Liste werden Editierregeln für jede Buchstabe gelernt (Einfügen, Löschen und Ersetzen) und diese Regeln werden dann um die primäre Liste zu kürzen verwendet. Es wird versucht, wie bei einer ordinären Editierdistanzberechnung, das kürzere Wort in dem längeren zu umwandeln. Dabei werden für die sekundäre Wortpaarkandidatenliste nur die Wortpaaren mit Editieroperationen, die am wenigsten vorkommen, ausgewählt. Zum Schluss wird die relative Differenz zwischen den Buchstaben n-gram Perplexitäten jedes Wortpaares verwendet, um die sekundäre Liste einzuordnen. Für unser Homogenisierung des arabischen Korpus werden die Wörter mit der höchsten relativen Differenz der Buchstaben Perpläxität ausgewählt. Dabei betrachtet man bei jedem Wortpaar in der sekundären List das Wort mit der niedrigeren Perplexität als korrekt.

Zum Bewerten der Performanz unser Verfahren vergleichen wir die Perplexität für den arabischen Text Korpus vor und nach der Homogenisierung. Außerdem bewerten wir den Einfluss der Homogenisierung auf der Leistung des baseline SMT system mittels BLEU Score.

Contents

List of Figures

1. Introduction

Thanks to its economical aspect and easy application in different fields the machine translation (MT) enlarges the number of people able to participate in today's information revolution. It also bridges the language barrier gap by allowing the construction of communicative systems to the more than 7000 different languages in the world. It is often argued whether a machine translation system needs to understand the natural languages for a correct translation. Unlike the statistical approach the rule-based machine translation (RBMT) requires a large set of rules developed by skilled language experts. It also requires, as confirmed by [Arn93], additionally to the morphological and syntactic rules a semantic analysis of both source and target languages. Since the language is a skill that develops extraordinarily fast in the first years of the life of an individual, this developmental process has been the subject of many RBMT researches. Comparing the human language development with the rule understanding for a RBMT system provides some constraints that make it feasible to apply computational methods. For the example presented by [Ed.71], analog to human children who are almost only exposed to positive evidence, RBMT systems are provided with special rules so that during their learning process only evidence for what is a correct form is provided. Statistical machine translation (SMT) unlike RBMT does not requires a skilled linguist to design the grammar or the rules to be used. However a large bilingual corpus of data is required to generate translations using statistical methods. Since in our project we investigate unsupervised methods for the homogenization of Arabic corpora, we also opt for statistical approach to our MT system. Besides we chose to develop and improve systems that need as minimum manual intervention as possible.

1.1 Basics of Statistical Machine Translation

In this section we present the milestones of statistical machine translation as described in [Nie14]. After the preprocessing the next challenge in the SMT is to automatically align words and phrases within sentence pairs in the English-Arabic parallel corpus. Then, using this parallel corpus, probabilities are also determined automatically by training statistical models. Besides. these statistical models are used to create many alternatives (hypotheses) to each sentence. The next step is giving a score to each hypothesis in order to finally be able to perform the search, which consists on selecting the best one.

Figure 1.1 illustrates a basic SMT system architecture. The translation task is divided into training and test. While the training is generally based on huge parallel corpora and

aims to estimate language and translation models, the test, is applied on small parallel corpora and aims to evaluate the performance of not only the decoding task (Search) but also the whole translation process. The decoder uses the models generated during the training stage and produces the best translation possible for sentences in the source language. The closer the produced sentence matches the original sentence in the target language the higher the performance of the whole SMT system is.

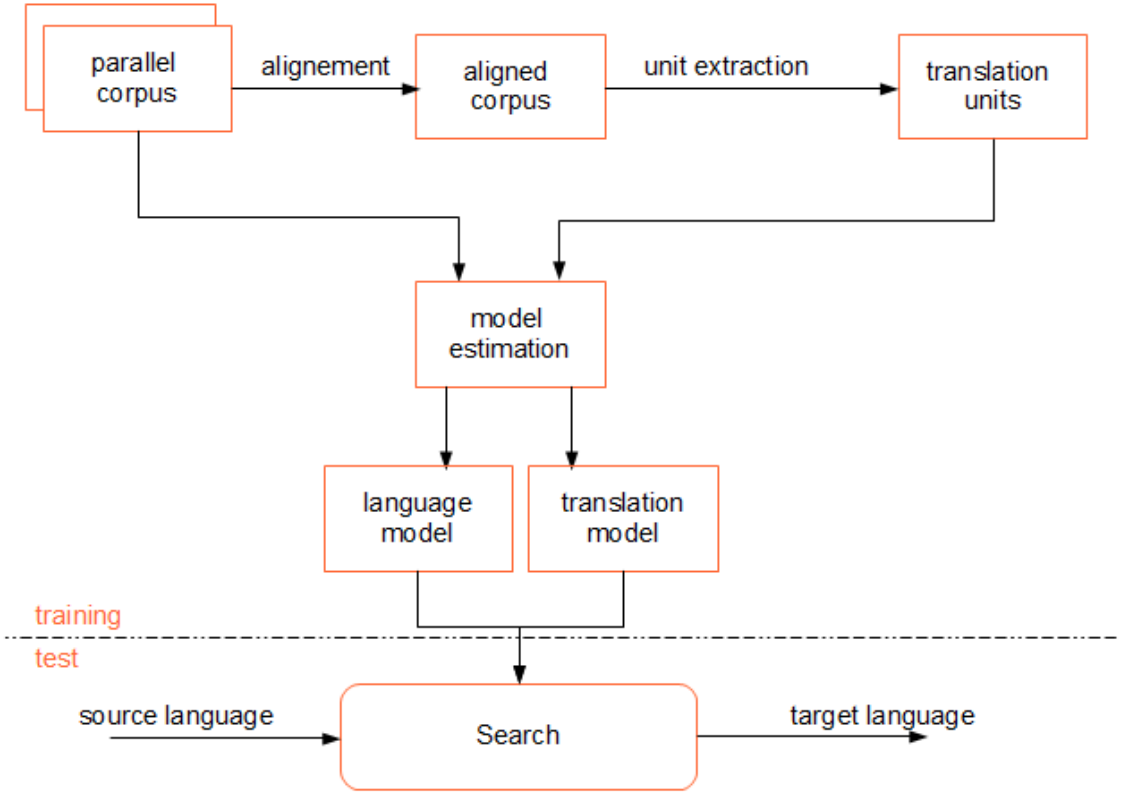

Figure 1.1: Architecture of a SMT system [Cle08].

One of the main advantages of SMT is that speed can be traded with quality, which makes it possible to have middle fast translations with a reasonable quality. Besides this trade-off avoids having only extreme fast translation with poor quality or very slow translations with outstanding quality. Since SMT dispenses with linguistic knowledge it requires minimal human effort and avoids the hard decisions when designing grammatical rules. As confirmed by [Nie14], given enough training data a SMT system can be created for any language pair and even a rapid prototyping of new systems can be realized at low cost. It is also possible, in case only few in-domain data are available, to adapt another SMT system of the same source and target languages to a specific domain. For example [NW12] adapte the phrase table to the TED domain (see Section 4.1) using the backoff approach and applying candidate selection.

However the difficulty of having enough data to train the model parameters is one of the greatest challenges of SMT. Moreover SMT does not explicitly deal with syntax, which increases the risk of learning meaningless rules or outputting incorrect translations [Nie14].

Figure 1.2 describes how the SMT for an English-Arabic system works. The decoding task consists in finding the most probable translation e for a given source sentence f.

$$\alpha = \underset{e}{\operatorname{argmax}}\, p(e|f)$$

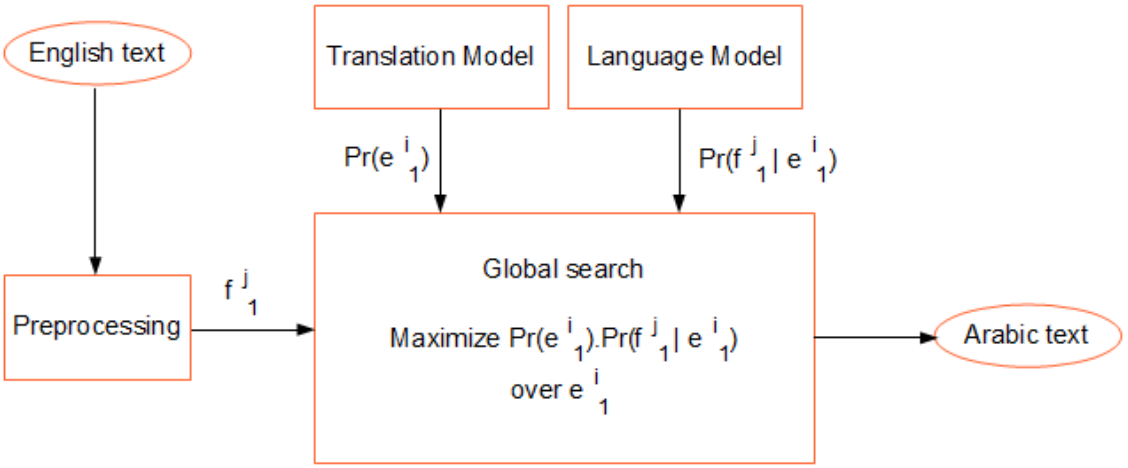

Figure 1.2: Architecture of an English-Arabic SMT system [Nie14].

Using the Bayes rule α can be easier calculated.

$$\alpha = \underset{e}{\operatorname{argmax}}\, p(e|f) = \underset{e}{\operatorname{argmax}}\, \frac{p(f|e)p(e)}{p(f)} = \underset{e}{\operatorname{argmax}}\, p(f|e)p(e)$$

1.2 Motivation

Although being not widely known yet, English-to-Arabic SMT is a challenging research issue for many of the researchers in the field of Arabic Natural Language Processing (NLP). Since the number of Arabic native speakers in the world is approximated to over 360 millions and since an important part of them do not speak English, we see a great potential in developing English-Arabic MT translation in order to develop more communicative systems and help the occidental culture to be better understood.

A challenging aspect for SMT is the parallel data sparsity. Despite that many volunteers in the web help translate English into Arabic language, it has always been an additional challenge for the statistical machine translation to deal with translation inconsistencies and spelling mistakes.

In our work we investigate homogenization methods of Arabic corpora for SMT and we aim to provide a better input text to the SMT system so that it learns better statistical rules. Since the rich and complex morphology of Arabic language has always been a challenge for machine translation, we only expect our system to work well enough for an Arabic native speaker to get the approximate meaning of what is written in English.

Another challenging aspect of the English-Arabic SMT is the influence of the dialect on the translation and the difference in word ordering between Arabic and English languages. This issue, according to [AOS14] increases the possibility of having more than one meaning for the same sentence. Having different possible word forms in Arabic makes it even worse, since a single sentence can be expressed in different forms. In our project we work on decreasing the number of possible word forms without changing the meaning of the sentences.

1.3 Goals of the project

In this project we aim to homogenize the Arabic text corpora for the English-Arabic MT. Our goal is to reduce the spelling diversity in the corpus and study its effects in a MT

environment. In order to achieve that, we apply some semantic and lexical word similarity measures on the words of the Arabic training corpus. Then we select a word pairs set with the highest similarity and try to differentiate between normal word inflections and correct alternative spellings for the same word. Finally we attempt to detect as much relevant words for the homogenization as possible using a fully unsupervised approach.

1.4 Structure of the thesis

After the introduction in Chapter 1 and the motivation to English-Arabic MT in Section 1.2, we present previous relevant research of other groups in Chapter 2. Then, we describe our methods for the spelling diversity detection in Chapter 3 and discuss our experiments and results in Chapter 4. Finally we give a summary of our work and present our future work and perspectives in Chapter 6.

2. Related work

Since there are a lot more attempts made to develop or enhance SMT systems of Arabic into other languages than the other way around, we first present few examples of Arabic-English MT researches before tackling the English-Arabic MT in Section 2.1. For example we cite the work of [Hab08], who describe four techniques for online handling of Out of Vocabulary (OOV) words in phrase-based Arabic-English SMT. They distinguish between the profile of OOV words as major challenges for Arabic processing and the profile of OOV words in Arabic-English SMT. The first challenge is treated with normal Arabic text normalization, such as removing all diacritics, normalizing Alif and Ya forms, and tokenizing Arabic text in the highly competitive Arabic Treebank scheme as described by [HS06]. To deal with the second challenge they present four techniques (MORPHEX, SPELLEX, DICTEX and TRANSEX) based on extending the phrase table with possible translations of the OOV words. While MORPHEX and SPELLEX techniques consist on matching the OOV word with a possible variant from IN Vocabulary (INV) list, DICTEX and TRANSEX techniques add completely new entries to the phrase table. The best BLEU score of 45.60 is achieved by applying a combination of all the four techniques.

[RMKM06] develop an Arabic-to-English speech-to-speech translation devise. The Arabic-English machine translation component is developed jointly at USC/ISI and Language Weaver, Inc. They combine some vocabulary optimization techniques on both Arabic and English sides with Arabic vocabulary optimization in form of morpheme segmentation and orthographic normalization. These optimization steps before training help achieve a final BLEU score of 29.72, which illustrates a +3.65 point increase over the 26.07 score for the baseline system.

2.1 English-Arabic Machine Translation

In this section we present some recent works on the English-Arabic SMT.

[BZG08] present an English-Arabic SMT system and investigate the benefits of the morphological decomposition of Arabic training corpora. They also describe different recombination techniques and report on the use of factored translation models for English-Arabic translation. For purposes of their experiments they use the International Workshop on Spoken Language Translation Arabic-English Corpus (IWSLT) [For07] and achieve a best BLEU score of 30.10.

[SD07] build an English to Iraqi Arabic MT system using a parallel corpus with 459K utterance pairs which is equivalent to 90K words (50K morphemes). They develop a joint

morphological-lexical language model (JMLLM) to be used in SMT of language pairs where one or both of them are morphologically rich. The JMLLM takes advantage of the rich morphology to reduce the Out-Of-Vocabulary (OOV) rate. They achieve a best MT result of 37.59 BLEU using a N-Best Oracle language model.

[Sul11] introduce two approaches to augmenting English-Arabic SMT with linguistic knowledge. For the first approach they add linguistically motivated syntactic features to particular phrases. These added features are added to penalize the incorrectly mapped phrase pairs, where the English part of these phrase pairs usually does not have a corresponding Arabic translation. The second approach improves morphological agreement in MT output through post-processing. The post-processor is based on a learning framework and it predicts inflections of words in MT output sentences using a training corpus of aligned sentence pairs. Thanks to both approaches they improve their baseline SMT System from 10.75 to 10.80 BLEU.

[ADG10] propose an approach to build a Transfer Module (TM) by building a new transfer-based system for MT using Artificial Neural Networks (ANN). The TM extracts the bilingual translation knowledge from the pairs of English and Arabic sentences. They chose not to use the performance of the MT as an evaluation metric but to use the number of sentences correctly translated. From the 200 sentences in the test set 64.5% of the transferred sentences had 60% or more of correct tags and 56% were even perfectly transferred.

Since the homogenization of text corpora is a part of the preprocessing, we also present in this section some state of the art Arabic text preprocessing techniques for MT. [HS06] measure the the effect of different word-level preprocessing decisions for Arabic on Arabic-English SMT system performance. They define a specific kind of preprocessing as a ”scheme” and differentiate it from the ”technique” used to obtain it. They define six different preprocessing schemes:

- A simple tokenization (ST) where they split off punctuations and numbers from words and remove all diacritics.
- Three decliticizations approaches (D1, D2, D3) where they split off the class of conjunction clitics in different ways.
- Splitting the words into morphemes (MR).
- English-like scheme where they try to minimize differences between Arabic and English by decliticizing similarly to D3 but using lexeme and English-like POS tags instead of the regenerated word.

[AHL12] apply various segmentation schemes in the preprocessing step of English-Arabic SMT to both of the training and the test sets. They explore a full spectrum of Arabic segmentation schemes ranging from full word form to fully segmented forms separating every possible Arabic clitic. Afterwards they examine the effects on system performance. The achieved results of SMT systems show a difference of 2.61 BLEU points between the best (36.25) and worst segmentation schemes (33.64).

2.2 Arabic text normalization

In this section we present some recent works on the Arabic text normalization. [AS09] introduce an algorithm to normalize noisy Arabic text. The goal of this algorithm is extorting structured or semi-structured information from data that has been previously considered noisy and unstructured. Additionally to the algorithm, a new similarity measure to stem Arabic noisy document is introduced. This similarity measure is very similar to common word edit distance algorithms but includes some Arabic language specifications

such as excluding words with three or less characters, substituting the long vowels with the wildcard character "?" (see Figure 2.1) and reducing any consecutive similar characters $x...x$ to only one character x. They argument the need for such a new measure stems from the fact that the common rules applied in stemming cannot be applied on noisy texts, which do not conform to the known grammatical rules and have various spelling mistakes. The term stemming is defined in linguistic morphology and information retrieval fields as a process for reducing inflected words to their word stem, base or root form.

Figure 2.1: Mapping of the long vowels to a wildcard character in Arabic.

[Att08] build an Arabic parser using XLE (Xerox Linguistics Environment) which allows writing grammar rules and notations that follow the Lexical Functional Grammar (LFG) formalisms. XLE includes, additionally to a parser, transfer and generator components, which makes it suitable for Machine Translation. The Arabic parser can also be described as an ambiguity-controlled morphological analyzer in a rule-based system, which takes the stem as the base form using finite state technology. Since syntactic ambiguity is very common in Arabic natural language, they try to identify sources of syntactic ambiguities in Arabic, focusing on four ambiguity-generating areas with the greatest impact. These ambiguity sources are:

- the pro-drop nature of the language,
- the word order flexibility,
- the lack of diacritics,
- and the multifunctionality of Arabic nouns.

3. Arabic text homogenization

In this chapter we describe our homogenization technique based on the detection of spelling diversity in the Arabic text corpora. Our unsupervised approach relies on selecting word pairs with the highest lexical and semantic similarity, in order to use them for corpus homogenization task.

3.1 Spelling diversity detection

For the purpose of our experiments, we look for Arabic words with minimal spelling difference but carrying the same meaning. Classifying each pair or group of these words in a cluster is a part of the homogenization of Arabic text corpora. An example of such words is the spelling of the word America. As shown in Figure 3.1, the Middle East and North African Arabic native speakers spell and pronounce the same word differently. In order to flag such words in the Arabic corpus, we propose an unsupervised approach based on finding the words with highest lexical and semantic similarity.

Figure 3.1: Example of Arabic words that need to be classified into the same cluster.

3.1.1 Levenshtein distance

The levenshtein distance (LD) of two words is defined as the minimum number of changes necessary to convert one word into another, where each change is the insertion, deletion, or substitution of a letter. Furthermore we present a brief description of the algorithm for the calculation of the levenshtein distance of two words a and b with lengths m and n:

$$D_{0,0} = 0$$
$$D_{i,0} = i \quad \text{for i in} \quad [1..m]$$
$$D_{0,j} = j \quad \text{for j in} \quad [1..n]$$
$$D_{i,j} = min \begin{cases} D_{i-1,j-1} & +0 \quad \text{if} \quad a_i = b_j \\ D_{i-1,j-1} & +1 \quad \text{in case of a substitution} \\ D_{i,j-1} & +1 \quad \text{in case of an insertion} \\ D_{i-1,j} & +1 \quad \text{in case of a deletion} \end{cases}$$

Figure 3.2 shows the iterations of computing the levenshtein distance of two Arabic words (/america/ and /amrica/).

Figure 3.2: Example of the calculation of the levenshtein distance between two Arabic words.

Since the Levenshtein approach neglects differences in word length, we use the levenshtein ratio, which we get from dividing the LD by the number of symbols of the longer of the two compared words. For example the levenshtein ratio makes the difference between a pair of long words and a pair of short words having the same levenshtein distance. After computing the levenshtein ratio to each couple of words in our $158K$ sized Arabic training vocabulary list, we prune the result to select only word pairs with relevant ratios. For example we select for each word w only the 10 words $w_1, ..., w_{10}$ which have the highest levenshtein score with that word. In the case where there are more than 10 words that have a levenshtein score higher than a threshold of 0.8 we append them to the top 10 list. After the pruning we reduce the size of our word pairs list from $(158K)^2 = 25000M$ to only $1,6M$.

3.1.2 Word-Vector distance

The word2vec toolkit is developed by Mikolov, Sutskever, Chen, Corrado and Dean in 2013 at Google Research. It takes a text corpus as input and produces the word vectors as output. These vector representations of words are used in our experiments in order to estimate the semantic similarity between the words of our Arabic corpus. Word2vec assumes that the word meaning and the relationships between words are encoded spatially, which makes

spatial distance corresponds to word similarity. Besides, the distributed representations of words in a vector space is more practical for grouping similar words. [Wan14] describes word2vec as a successful example of "shallow" learning that can be trained as a very simple neural network. This neural network has a single hidden layer without any non-linearities and includes no unsupervised pre-training of layers. As confirmed by [MSC+13], what makes the word representations computed using neural networks even more interesting is the fact that the learned vectors explicitly encode many linguistic regularities and patterns. Figure 3.3 shows some Arabic word pairs with the highest word2vec scores and their translations. These are in deed semantically very close from each other, which makes word2vec an efficient and reliable tool for measuring semantic similarity.

Word2vec score	Arabic word pairs	Translation in English
0.995728	الاثنين الخميس	Thursday Monday
0.995187	سبعة ثمانية	Eight Seven
0.995078	الخميس الأربعاء	Tuesday Thursday
0.992589	الاثنين الأربعاء	Tuesday Monday
0.992191	السادسة السابعة	Seventh Sixth
0.992095	ثمانية تسعة	Nine Eight
0.990908	عشرين ثلاثين	Thirty Twenty

Figure 3.3: Example of similar Arabic word pairs flagged using word2vec score.

Analog to the pruning process described in Section 3.1.1 and after computing the word2vec distance to each couple of words in our 158k sized Arabic training vocabulary list, we prune the result to select only word pairs with relevant distances. Additionally to the selection of only word pairs with word2vec distance higher than a threshold of 0.2, we only retain the 10 words with highest semantic similarity to each word w. This pruning reduces the size of our word pair list from about $25M$ to $102K$.

Furthermore we investigate the influence of using more training data on the quality of the word2vec similarity measure. Thus, we select bigger text corpora (Gigawords + UN) to train word vectors and select the list of word pairs with most relevant semantic distances. In Section 4.2 we present and discuss all the results of word2vec word similarity measures.

3.1.3 English-Arabic bilingual association score

The calculation of the English-Arabic bilingual association score, as described by [CCB+05], is based on both word alignment tables of English-Arabic and Arabic-English. First we combine both alignment tables into one bi-alignment table with a more compact format. Then, we select for each word w_{AR} the list of English words $w^1_{EN}, ..., w^N_{EN}$ associated to it. For each one of these English words we select the list of Arabic words $w^1_{AR}, ..., w^M_{AR}$ associated to it and that differ from the initial Arabic word w_{AR}. Finally we use a counting file produced by giza, which contains the frequency of each training sentence to assign scores to each word pair $\{w_{AR}, w^i_{AR} | 1 \leq i \leq M * N\}$. We also apply the pruning to the produced list of word pairs by selecting for each word w_{AR} the 10 Arabic words with the highest English-Arabic bilingual association scores. This reduces the size of the word pairs list from about $25M$ to $312K$.

3.1.4 Character n-gram perplexity probability

Another word similarity measure applied in our work is the character n-gram perplexity probability. This measure is based on the perplexity measure of character n-grams of

each word. First we split our Arabic corpus into a single word in each line. Then as illustrated in Figure 3.4, we insert a single space between the letters of each word so that each letter can be considered as independent string. Since we need two different text sets (training and test) for a reliable perplexity measure, we select the Arabic development set of our SMT system as LM-training set and evaluate with the Arabic training corpus. We argument this choice by the fact that we want to compute the perplexity for the words of the training corpus and with the better quality of the text in the development compared with the training set.

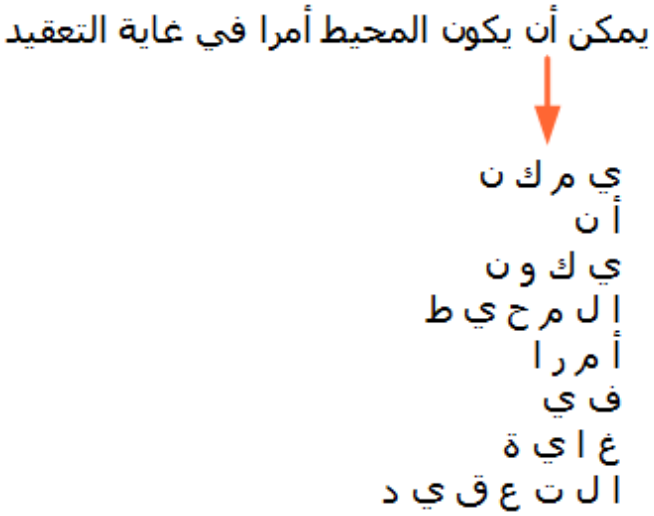

Figure 3.4: Example of splitting an Arabic sentence into character n-grams.

After that we build a 20-gram language model (LM) to cover all the context information of long words. Our LM is built using SRILM toolkit described in Section 4.1.1.5. Evaluating that LM with the character n-grams of the words in our Arabic corpus we get the perplexity score of each word. This perplexity score carries implicit information about how often a particular sequence of letters appears in the whole text corpus. Finally we eliminate the spaces between the letters of each word in order to generate a list of all the words of the training corpus together with their perplexity scores. The perplexity scores are presented and discussed in Section 4.2.

4. Experiments and results

In this chapter we describe our baseline English-Arabic SMT system and the Homogenization process of Arabic text corpora.

4.1 Baseline English-Arabic Machine Translation system

Our baseline is a phrase-based English-Arabic SMT system. It uses:

- a translation model (TM), that captures the lexical and word reordering relationships between English and Arabic languages,
- a distortion model, which represents the relative source position of to adjacent target phrases,
- a word-count model, which represents the number of generated target words, since the language model prefers short translations,
- a phrase-count model for longer phrase pairs, since they capture longer context,
- and an Arabic language model (LM) to help the decoder chose the best translation model hypothesis.

As training material for our English-Arabic SMT system we use TED parallel corpora from The 11th International Workshop on Spoken Language Translation 2014 (IWSLT14). TED, as described by [CNS+14] is a nonprofit organization that makes video recordings of many famous thinkers and authors from allover the world giving the talk of their lives. Since all the talks have English captions and there are many volunteers worldwide to translate them into other languages, TED corpora are reasonably suitable for training MT systems.

Our baseline system is developed using the Systembuilder tool, which is implemented by Jan Niehues from the Institute of Cognitive Sciences at the Karlsruhe Institute of Technology. The Systembuilder combines both of the training and test tasks and is configured through an XML file, which we call description file.

4.1.1 SMT system training

In this section we describe the SMT training steps realized by the Systembuilder. As described in Figure 4.1 the SMT training requires additionally to the parallel corpora a monolingual corpus in the target language, which is in our case Arabic. While the parallel corpora are used for the word alignment and phrase table generation, the monolingual corpus is used for language model training.

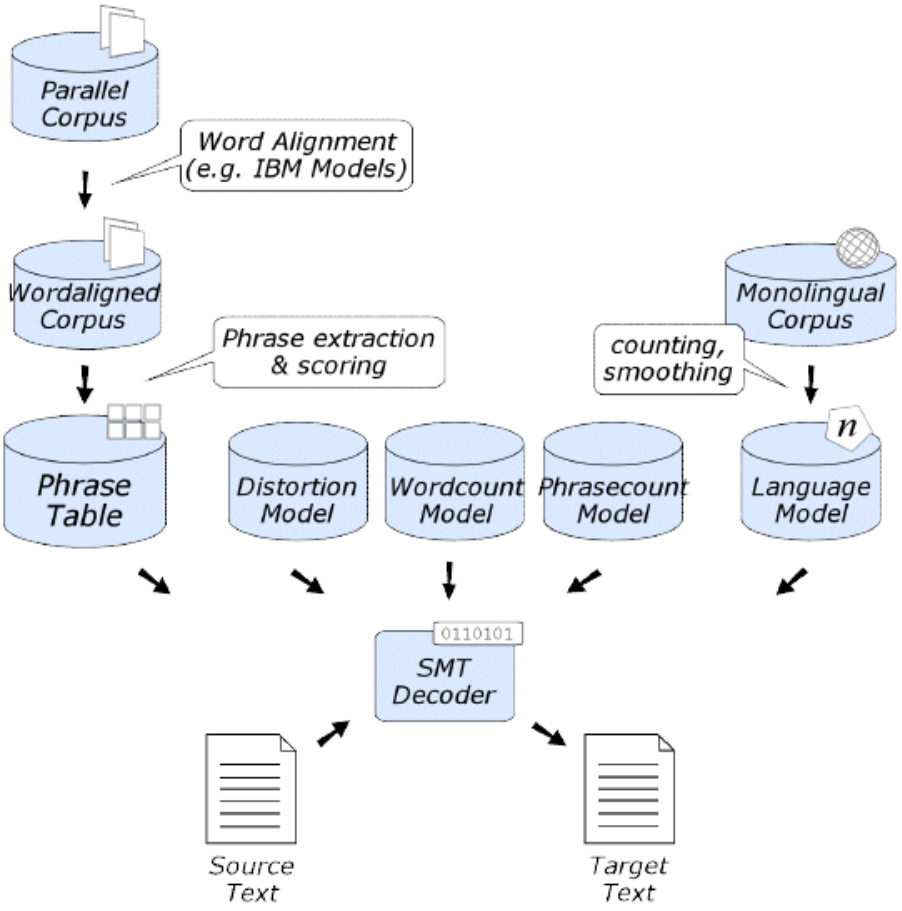

Figure 4.1: Phrase based SMT system [Nie14].

4.1.1.1 Preprocessing

The common preprocessing as described by [HHN+13] is applied to the raw data before performing any model training. Additionally to the normalization of special symbols, dates and numbers, the preprocessing task includes a smart-case model, which normalizes the first letter of every sentence.

Another part of the Preprocessing in our baseline SMT system is the Part-of-Speech reordering (POS reordering), which encodes different possible reorderings of the source sentence in a lattice (see Figure 4.2). As mentioned by [Nie14], it also assigns probabilities to the different paths in the lattice and provides a better restriction than simple a reordering window.

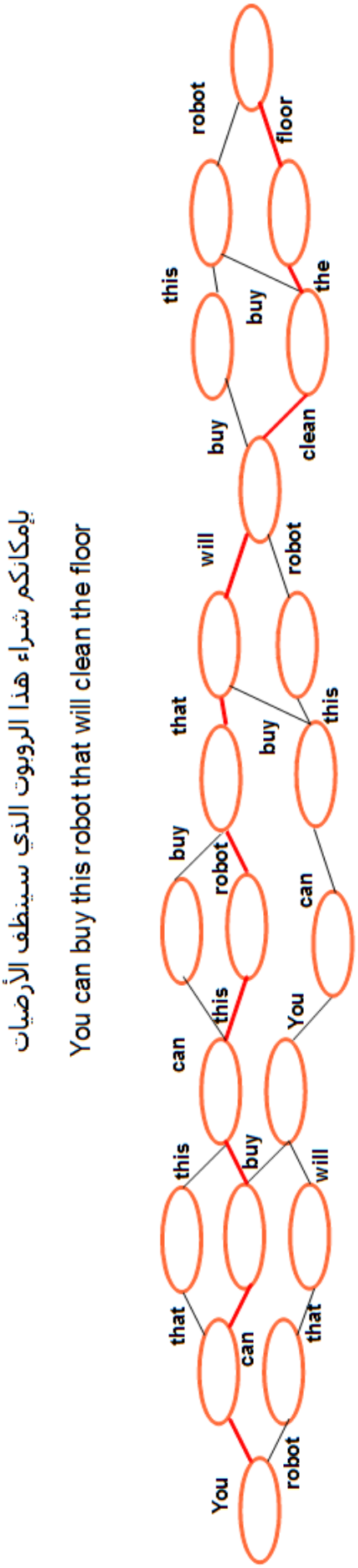

Figure 4.2: Example of a lattice after the POS reordering.

4.1.1.2 Word alignment

The word alignment as a part of a word-based translation model (WBTM) was first introduced by [BCP+90]. In order to directly translate source words to target words, they define word-by-word translation probabilities. Since this technique is outdated by the appearance of phrase-based translation, the generation of word alignment is only used for the phrase extraction in phrase based models. As illustrated in Figure 4.3 the rich morphology of the Arabic language causes a lot of alignment difficulties but these can be overcome by looking at the translation process from both directions.

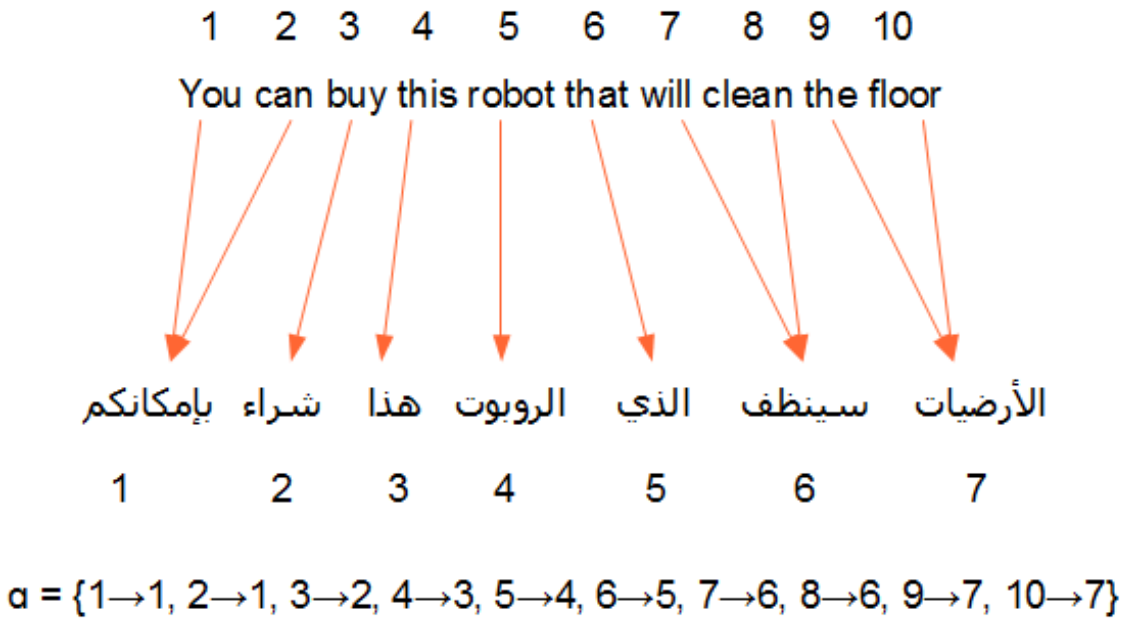

Figure 4.3: Example of English-Arabic word alignment.

4.1.1.3 Reordering rules

[AOP06] define the fundamental difference between decoding for MT and decoding for speech recognition to be in the word reordering. The word order for speech signal decoding is evident, since the recognition is based on an acoustic signal that contains obviously the word order. However, that is not necessarily the case in MT due to the fact that different languages have different word order requirements. For example, in Arabic adjectives are mainly noun post-modifiers, whereas in English adjectives are noun pre-modifiers. This makes the reordering for English-Arabic SMT obligatory.

4.1.1.4 Phrase tables

The phrase tables of our baseline SMT system are trained using GIZA++ alignment. GIZA++, as described by [CV07], is an extension of the program GIZA, which is a training program written in C++ with the Standard Template Library (STL library) and which learns statistical translation models from bilingual corpora. We use GIZA++ to generate word alignments in an Arabic-English parallel corpus. This means that both alignments "target-to-source" and "source-to-target" are generated before being combined. After that, we use these alignments for building phrase-based models.

4.1.1.5 Language modeling

Since we prefer statistical approaches in our baseline MT system, we chose to use a statistical language model. Such a LM assigns a probability to a sequence of words $P(w_1..w_n)$

by means of a probability distribution. This probability is computed using statistics from large Arabic corpora. In our baseline system we build a 4-gram language model with SRI Language Modeling toolkit (SRILM), which is described by [Sto00] and [Sto02]. In order to estimate the probability of a sentence, we need to beak up into the prediction of single words. This formula calculates the product of word probabilities given history:

$$p(w_1, w_2, ..., w_N) = p(w_1) * p(w_2|w_1) * ... * p(w_N|w_1, w_2, ..., w_{N-1})$$

In order to estimate the 4-gram Probabilities, we calculate the maximum likelihood estimation for the word sequence $w_1, ..., w_4$:

$$p(w_4|w_1, w_2, w_3) = \frac{count(w_1, w_2, w_3, w_4)}{\sum_w count(w_1, w_2, w_3, w_4)}$$

Besides we apply the Kneser-Ney Smoothing in our language modeling, which, as described by [Nie14], modifies the probability of a word w to

$$p_{KN}(w) = \frac{N_{1+}(*w)}{\sum_i N_{1+}(*w_i)}$$

where $N_{1+}(*w) = |\{w_i : c(w_i, w) > 0\}|$ is the count of history for a word w.

4.1.2 SMT system decoding

As illustrated in Figure 4.1 the SMT decoder requires additionally to the phrase table and the language model, a distortion model, a word-count model and a phrase-count model. Using all these five features the decoder calculates the translation probability of each sentence using log-linear models. The general formulation of log-linear probability is:

$$p(x) = exp(\sum_{i=1}^{n} \lambda_i h_i(x))$$

where λ_i is the coefficient of the feature f_i and $h_i(x)$ is the function that calculates the score of the feature f_i.

In order to optimize the translation performance the decoder applies parameter tuning on the coefficients of the features used in the translation. The decoder requires also two sets of parallel text that are not a part of the training set: The development set and the test set. As described in Figure 4.4 the tuning cycle starts with initial parameters, that are used by the decoder to translate the English part of the development set. Then it compares the generated translation to the original Arabic text from the development set and optimizes the parameter. The new parameter are then given back to the decoder so that the tuning circle can be closed. Once the parameter converge or a specific number of tuning iterations is reached the decoder uses the final parameters to translate the English part of the test set. Comparing the translation to the original Arabic test set, we can measure the performance not only of the decoder but also of the complete SMT system.

4.2 Results of word-word similarity measures

In this section we give an overview of the results for the Arabic word-word similarity measures. In our work we consider two lexical similarity measures, which are the levenshtein ratio and the character n-gram perplexity. We also use two semantic similarity measures, which are the word2vec and the English-Arabic bilingual association score. We combine

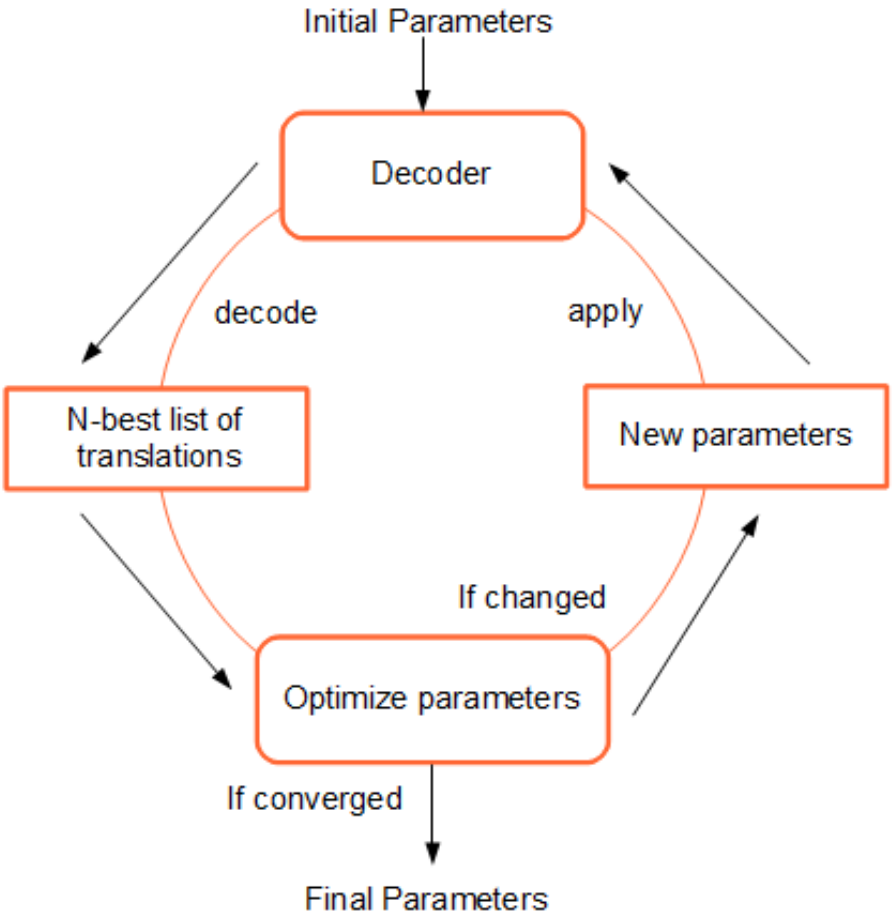

Figure 4.4: Decoder tuning cycle and parameter estimation [Nie14].

both lexical and semantic similarity features to generate a primary list of word-candidates that might be classed in the same cluster in order to reduce the morphology of our Arabic training corpus. Thus we select from each word pair lists of the different similarity measures only the word pairs that have:

- a levenshtein score
- a word2vec score
- and an English-Arabic bilingual association score

Since each word of our Arabic training corpus has a perplexity score, we append the perplexity scores of each word pair to our primary list of word-candidates. Table 4.2 presents some examples of Arabic word pairs with corresponding word-word similarity scores. The green colored word pairs are for example different correct spellings for the same word in Arabic language. These are the words we look forward to find in order to reduce the complexity and morphology of the Arabic text Corpus. The blue colored pairs are a combination of a correct word and a misspelled one. These help us also normalize the Arabic text corpus in order to enhance its quality. The rest of the word pairs represent normal word inflections in Arabic language. Therefore from now on we focus on filtering the 2816 sized primary candidates word pair list in order to get a final word pair list with only or at least as much as possible word pairs that can be used to optimize the Arabic text corpus.

Word1	Word2	Levenshtein	Word2vec	Bilingual association	PPL word1	PPL word2
نشر	نشرا	0.857	0.465	1.180	20.120	25.011
نشر	نشرت	0.857	0.657	22.774	20.120	12.584
نشر	نشره	0.857	0.748	0.102	20.120	11.849
غوانتانامو	غوانتنامو	0.947	0.822	0.333	63.156	42.407
يتوجب	يجب	0.75	0.695	29.795	6.614	13.769
ثلاث	ثلالث	0.888	0.425	0.326	6.001	49.303
الرجال	الرجل	0.909	0.613	0.149	5.288	6.063
عندما	عنما	0.888	0.483	6.339	3.649	50.502
اعلام	الاعلام	0.833	0.605	0.056	13.049	10.68
ميتشل	ميتشيل	0.909	0.950	1.777	131.574	66.02
تمتد	يمتد	0.75	0.712	2.665	63.767	50.120

Table 4.1: Examples of Arabic word pairs with corresponding word-word similarity scores.

5. Evaluation of Arabic text corpus homogenization

After our unsupervised approach based on finding the words with highest lexical and semantic similarity, we select a primary list of word pair candidates, which contains in addition to words that are relevant for the Arabic corpus homogenization, the normal word inflections. From the word pairs in the primary list we learn edit rules (insertions, deletions and substitutions) corresponding to each letter and use these statistics to reduce the size of the primary list of word pair candidates. Similarly to a normal edit distance calculation we try to modify the shortest word in the word pair into the other one. Then we select for the secondary list only the word pairs with the least common edit operations. Finally, we use the relative difference between the character n-gram perplexity measures of each word pair to rank the secondary list. We consider the words with highest relative perplexity difference for our Arabic corpus homogenization and for each word pair from the secondary list we consider the word with lowest perplexity as correct. In order to measure the efficiency of our approach we compare the perplexity of the Arabic corpus before and after the homogenization using increasingly more word pairs (see Table 5.1). In addition we measure the influence of the homogenization on the performance of our baseline SMT system in terms of BLEU score.

Number of word pairs used for homogenization	Number of words corrected	Perplexity	BLEU
Original Corpus (baseline)	0	1323.72	0
10	61	1323.71	7.78
20	154	1323.47	7.78
30	244	1323.37	7.78
40	378	1323.19	7.78
50	510	1323.03	7.78
60	579	1322.85	7.78
70	1170	1323.1	7.78
80	1398	1323.81	7.78
90	1572	1323.46	7.78
100	1891	1323.27	7.78

Table 5.1: Influence of homogenization on Arabic corpus and English-Arabic SMT systems.

Table 5.1 illustrates the influence of homogenization on Arabic corpus and English-Arabic SMT systems. In order to measure the perplexity, we build different language models using homogenized Arabic corpora and evaluate them on the Arabic corpus from the test set of our baseline SMT system. In addition we build different SMT systems with the homogenized Arabic corpora and measure their performance with the BLEU score. The improvement of the quality of the Arabic text is justified by the number of corrected words using our unsupervised approach. Nevertheless the perplexity results and the BLEU scores show almost no improvement. We explain this by the fact that the amount of corrected words in the training text, which has 2.46 million words is too small.

6. Summary and future work

In this project we try to decrease the degree of diversity of word spelling in Arabic text in order to optimize the English-Arabic Machine Translation. Our goal is to detect correct alternative spellings for the same word in order to unify word spellings in the Arabic training corpora. Therefore, we first look for Arabic words with minimal spelling difference but carrying the same meaning. Then, we classify each pair or group of these words in a cluster. For detecting similar words in the Arabic training corpus, we propose an unsupervised approach, which applies different models to a word candidate list in a cascade manner. This word candidate list contains the words with highest lexical and semantic similarity. The application of each model returns a subset of the highest scoring candidates. We chose the levenshtein ratio and the ngram of characters perplexity as lexical similarity measures. For the semantic similarity measure we apply the word2vec and the English-Arabic bilingual association score. After our unsupervised approach based on finding the words with highest lexical and semantic similarity, we select a primary list of word pair candidates and apply several different selection approaches to optimize it. We try to reduce the number of word pairs with normal word inflections by analyzing the likelihood of corresponding edit rules necessary to modify one word to another for each word pair. Once we select our final list of word pairs, we consider for each word pair the word with lowest perplexity as correct.

In order to measure the efficiency of our approach we compare the perplexity of the Arabic corpus before and after the homogenization. In addition we measure the influence of the homogenization on the performance of our baseline SMT system in terms of BLEU score. Both of the perplexity measure and the BLEU score show small influence of the homogenization on Arabic corpora, which can be explained by the fact that the amount of corrected words in the training text, which has 2.46 million words is too small.

A part of our future work is developing a hybrid homogenization process combining our unsupervised approach of detecting words with highest lexical and semantic similarity with a manual selection of word pair candidates that can be relevant to the Arabic corpus homogenization. We propose a web based tool, where users check through radio buttons for each word pair, whether they represent a normal word inflection or not.

A further objective is to use more Arabic training data for the optimization of our similarity measures such as the word2vec vectors and bilingual association score. For example, it would be extreme helpful to have more in domain Arabic text data to learn better semantic rules and have a preciser estimation of the likelihood of word edit rules.

In order to consider more real scenarios, our future experiments will involve Arabic text normalization, since we believe that its combination with text homogenization might improve the performance of Arabic SMT systems considerably. Thus, our focus will turn into not only reducing the number of correct alternative spellings of the same word but also correcting misspelled words and reducing the number out of vocabulary words.

Bibliography

[ADG10] R. Al Dam and A. Guessoum, "Building a neural network-based English-to-Arabic transfer module from an unrestricted domain," in *Machine and Web Intelligence (ICMWI), 2010 International Conference on.* IEEE, 2010, pp. 94–101.

[AHL12] 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, no. 1-2, pp. 3–24, 2012.

[AOP06] Y. Al-Onaizan and K. Papineni, "Distortion models for statistical machine translation," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2006, pp. 529–536.

[AOS14] A. Alqudsi, N. Omar, and K. Shaker, "Arabic machine translation: a survey," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 549–572, 2014.

[Arn93] D. e. a. Arnold, " Machine Translation: an Introductory Guide," 1993.

[AS09] E. T. Al-Shammari, "A Novel Algorithm for Normalizing Noisy Arabic Text," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 4. IEEE, 2009, pp. 477–482.

[Att08] M. A. Attia, "Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation," Ph.D. dissertation, University of Manchester, 2008.

[BCP+90] 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, no. 2, pp. 79–85, 1990.

[BZG08] 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.* Association for Computational Linguistics, 2008, pp. 153–156.

[CCB+05] B. Chen, R. Cattoni, N. Bertoldi, M. Cettolo, and M. Federico, "The itc-irst smt system for iwslt-2005." in *IWSLT*, 2005, pp. 88–94.

[Cle08] J. M. C. Clemente, "Architecture and modeling for n-gram-based statistical machine translation," Ph.D. dissertation, Universitat Politècnica de Catalunya, 2008.

[CNS+14] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 11th IWSLT Evaluation Campaign, IWSLT 2014," in *Proceedings of the eleventh International Workshop on Spoken Language Translation (IWSLT), Lake Tahoe, CA*, 2014, pp. 2–17.

[CV07] F. Casacuberta and E. Vidal, "GIZA++: Training of statistical translation models," 2007.

[Ed.71] D. S. Ed., "On two types of models of the internalization of grammars," *The ontogenesis of grammar: A theoretical perspective*, 1971.

[For07] C. S. Fordyce, "Overview of the IWSLT 2007 evaluation campaign," 2007.

[Hab08] N. Habash, "Four techniques for online handling of out-of-vocabulary words in Arabic-English statistical machine translation," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, 2008, pp. 57–60.

[HHN+13] T.-L. Ha, T. Herrmann, J. Niehues, M. Mediani, E. Cho, Y. Zhang, I. Slawik, and A. Waibel, "The KIT translation systems for IWSLT 2013," in *Proceedings of IWSLT*, 2013.

[HS06] N. Habash and F. Sadat, "Arabic preprocessing schemes for statistical machine translation," 2006.

[MSC+13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[Nie14] J. Niehues, "Maschinelle Uebersetzung," *Interactive Systems Lab (ISL), Institute for Anthropomatics and Robotics, Karlsruher Institut for Technology (KIT), Karlsruhe, Deutschland*, 2014. [Online]. Available: http://isl.anthropomatik.kit.edu/english/2274_2325.php

[NW12] J. Niehues and A. Waibel, "Detailed analysis of different strategies for phrase table adaptation in SMT," in *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*, 2012.

[RMKM06] J. Riesa, B. Mohit, K. Knight, and D. Marcu, "Building an english-iraqi arabic machine translation system for spoken utterances with limited resources." in *INTERSPEECH*, 2006.

[SD07] R. Sarikaya and Y. Deng, "Joint morphological-lexical language modeling for machine translation," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Association for Computational Linguistics, 2007, pp. 145–148.

[Sto00] A. Stolcke, "SRI language modeling toolkit," *Version*, vol. 1, no. 3, p. 2000, 2000. [Online]. Available: http://www.speech.sri.com/

[Sto02] ——, "SRILM - AN EXTENSIBLE LANGUAGE MODELING TOOLKIT," *Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado*, September 2002. [Online]. Available: http://www.speech.sri.com/cgi-bin/run-distill?papers/icslp2002-srilm.ps.gz

[Sul11] S. Sultan, "Applying morphology to english-arabic statistical machine translation," Ph.D. dissertation, Master's Thesis Nr. 11 ETH Zurich in collaboration with Google Inc., 2011, 2011.

[Wan14] H. Wang, "Introduction to Word2vec and its application to find predominant word senses," 2014.