The Use of MSVM and HMM for Sentence Alignment

Mohamed Abdel Fattah*

Abstract—In this paper, two new approaches to align English-Arabic sentences in bilingual parallel corpora based on the Multi-Class Support Vector Machine (MSVM) and the Hidden Markov Model (HMM) classifiers are presented. A feature vector is extracted from the text pair that is under consideration. This vector contains text features such as length, punctuation score, and cognate score values. A set of manually prepared training data was assigned to train the Multi-Class Support Vector Machine and Hidden Markov Model. Another set of data was used for testing. The results of the MSVM and HMM outperform the results of the length based approach. Moreover these new approaches are valid for any language pairs and are quite flexible since the feature vector may contain less, more, or different features, such as a lexical matching feature and Hanzi characters in Japanese-Chinese texts, than the ones used in the current research

Keywords—Sentence Alignment, English/ Arabic Parallel Corpus, Parallel Corpora, Machine Translation, Multi-Class Support Vector Machine, Hidden Markov model

1. INTRODUCTION

Recent years have seen a great interest in bilingual corpora that are composed of a source text along with a translation of that text in another language. Nowadays, bilingual corpora have become an essential resource for work in multilingual natural language processing systems [1-4], including data-driven machine translation [5], bilingual lexicography, automatic translation verification, automatic acquisition of knowledge about translation [6], and cross-language information retrieval [7, 8]. It is required that the bilingual corpora be aligned. Given a text and its translation, an alignment is a segmentation of the two texts such that the nth segment of one text is the translation of the nth segment of the other (as a special case, empty segments are allowed, and either corresponds to the translator's omissions or additions) [6, 9]. With aligned sentences, further analysis such as phrase and word alignment analysis [10-12], bilingual terminology, and collocation extraction analysis can be performed [13, 14].

In the last few years, much work has been reported in sentence alignment using different techniques. Length-based approaches (length as a function of sentence characters [15] or sentence words [16]) are based on the fact that longer sentences in one language tend to be translated into longer sentences in the other language, and that shorter sentences tend to be translated into shorter sentences. These approaches work quite well with a clean input, such as the Canadian Hansards corpus, whereas they do not work well with noisy document pairs [14].

Manuscript received July 5, 2011; first revision March 3, 2012; accepted April 13, 2012.
Corresponding Author: Mohamed Abdel Fattah
* Dept. of Electronics technology, Helwan University, Cairo, Egypt (mohafi2003@helwan.edu.eg)

Cognate based approaches were also proposed and combined with the length-based approach to improve the alignment accuracy [17-20]. Ceauşu, et al., exploited SVM to discriminate between "good" and "bad" alignments for sentence alignment tasks [21]. Vogel et al., applied HMM in word alignment to an English-French corpus. Their approach depends on the statistical information, such as length ratios, alignment probabilities, etc., given from the input bilingual documents but do not include any external lexical information [22]. We have exploited SVM and HMM for sentence alignment using a different framework that is based on different text features.

Sentence cognates such as digits, alphanumerical symbols, punctuation, and alphabetical words have been used. However all cognate based approaches are tailored to close Western language pairs. For disparate language pairs, such as Arabic and English, with a lack of a shared Roman alphabet it is not possible to rely on the aforementioned cognates to achieve high-precision sentence alignment of noisy parallel corpora (however, cognates may be efficient when used with some other approaches). Some other sentence alignment approaches are text based approaches, such as the hybrid dictionary approach [23], part-of-speech alignment [24], and the lexical method [25]. While these methods require little or no prior knowledge of source and target languages and give good results, they are relatively complex and require significant amounts of parallel text and language resources.

Instead of a one-to-one hard matching of punctuation marks in parallel texts, as used in the cognate approach of Simard [17], Thomas [14] did not allow for matching and one-to-several matching of punctuation matches. However, neither Simard nor Thomas took into account the text length between two successive cognates (Simard's case) or punctuations (Thomas's case), which increased the system confusion and lead to an increase in execution time and a decrease in accuracy. We have avoided this drawback by taking the probability of text length between successive punctuation marks into account during the punctuation matching process, as will be shown in the following sections.

In this paper, non-traditional approaches for English-Arabic sentence alignment are presented. For sentence alignment, we may have a 1-0 match, where one English sentence does not match any of the Arabic sentences and a 0-1 match where one Arabic sentence does not match any English sentences. The other matches we focus on are 1-1, 1-2, 2-1, 2-2, 1-3, and 3-1.

There may be more categories in bi-texts, but they are rare. Therefore, only the previously mentioned categories are considered. If the system finds any other categories they will automatically be misclassified. As illustrated above, we have eight sentence alignment categories. As such, sentence alignment can be considered as a classification problem, which may be solved by using a Multi-Class Support Vector Machine (MSVM) and Hidden Markov Model (HMM) classifiers.

The paper is organized as follows. Section 2 introduces English-Arabic text features. Section 3 illustrates the new approaches. Section 4 discusses English-Arabic corpus creation. Section 5 shows the experimental results and finally, Section 6 gives concluding remarks and discusses future work.

2. ENGLISH-ARABIC TEXT FEATURES

As explained in [1], the most important feature for text is the text length, as Gale & Church

achieved good results using this feature. They used the fact that "longer sentences in one language tend to be translated into longer sentences in the other language, and shorter sentences tend to be translated into shorter sentences." As explained in [1] there is a good correlation (0.992) between English paragraph length and Arabic paragraph length (as a function of character numbers).

The second text feature is punctuation marks. When presenting the same content in two different languages, translators exhibit a strong tendency to use the same punctuation structure in the text pairs as much as possible. Take the following example, taken from a set of United Nations documents, to illustrate this fact:

"Some 40% indicated the programme of work was under review, while 47% had not reviewed it."

"وهناك حوالي 40% من البلدان بينت أن برنامج العمل يجرى استعراضه ، بينما قالت 47% من البلدان أنها لم تستعرضه".

The following table shows the punctuation marks that match between the previous two sentences.

English punctuation mark	%	,	%	.
Arabic punctuation mark	%	،	%	.

We can classify punctuation matching into the following categories:

A. 1-1 matching type, where one English punctuation mark matches one Arabic punctuation mark.

B. 1-0 matching type, where one English punctuation mark does not match any of the Arabic punctuation marks.

C. 0-1 matching type, where one Arabic punctuation mark does not match any of the English punctuation marks.

The probability that a sequence of punctuation marks $AP_i = Ap_1 Ap_2 .....Ap_i$ in an Arabic language text translates to a sequence of punctuation marks $EP_j = Ep_1 Ep_2 .....Ep_j$ in an English language text is P(APi, EPj). The system searches for the punctuation alignment that maximizes the probability over all possible alignments given a pair of punctuation sequences corresponding to a pair of parallel sentences from the following formula:

$$\arg\max_{AL} \; P(AL \mid AP_i, EP_j), \tag{1}$$

Since "AL" is a punctuation alignment. Assume that the probabilities of the individually aligned punctuation pairs are independent. The following formula may be considered:

$$\text{P (APi, EPj)} = \prod_{AL} P(Ap_k, Ep_k).P(\delta_k \mid match), \tag{2}$$

Where $P(Ap_k, Ep_k)$ = the probability of matching Ap_k with Ep_k, and it may be calculated as follows:

$$P(Ap_k, Ep_k) = \frac{Number\ of\ punctuation\ pair\ (Ap_k, Ep_k)}{Total\ number\ of\ punctuation\ pairs\ in\ the\ manually\ aligned\ data} \quad (3)$$

$P(\delta_k \mid match)$ is the length-related probability distribution function. δ_k is a function of the text length (text length between punctuation marks Ep_k and Ep_{k-1}) of the source language and the text length (text length between punctuation marks Ap_k and Ap_{k-1}) of the target language.

$P(\delta_k \mid match)$ is derived straight as in [1].

After specifying the punctuation alignment that maximizes the probability over all possible alignments given a pair of punctuation sequences (using a dynamic programming framework as in [15]), the system calculates the punctuation compatibility factor for the text pair under consideration as follows:

$$\gamma = \frac{c}{max\ (m, n)}$$

Where γ = the punctuation compatibility factor,
c = the number of direct punctuation matches,
n = the number of Arabic punctuation marks,
m = the number of English punctuation marks.

The punctuation compatibility factor is considered as the second text pair feature.

The third text pair feature is the cognate. For disparate language pairs, such as Arabic and English, that lack a shared alphabet it is not possible to rely only on cognates to achieve high-precision sentence alignment of noisy parallel corpora.

However, many UN and scientific Arabic documents contain some English words and expressions. These words may be used as cognates. Take the following example illustrates this fact:

"Many species, including, inter alia, river dolphins and porpoises, freshwater seals, manatees, hippopotamuses, the Asian water buffalo, otters, the European mink, the fishing cat and the flat-headed cat, the desmans (Desmana moschata or Russian desman and the Pyrenean desman [Galemys pyrenaicus]), and the well known semi-aquatic beavers are threatened or endangered mainly from habitat loss and degradation, pollution, overexploitation or entrapment in nets, and other fishing gear."

"وكثير من الأنواع ، شاملة دلفين النهر وخنزير النهر وكلب النهر وخروف الماء وأفراس النهر وجاموس الماء الآسيوى والقضاعة (otters) ، والمكسيك الأوروبي ، والقط صائد الأسماك والقط ذا الرأس المفلطح ، والدسمان الروسى والدسمان البرانسي (Galemys pyrenaicus) و القندس المعروف جيداً الذي هو حيوان نصف مائي ، كلها مهددة أو معرضة للمخاطر بسبب ضياع الموائل أو تدهورها أو التلويث أو الإفراط في الاستغلال أو صيدها بالشباك أو غير ذلك من أدوات الصيد."

In the previous example, the words "otters," "Galemys," and "pyrenaicus," were used in the Arabic sentence as they have no translation. These words may be used as cognates.

We define the cognate factor (cog) as the number of common items in the sentence pair.

When a sentence pair has no cognate words, the cognate factor is 0.

3. THE PROPOSED SENTENCE ALIGNMENT MODEL

The classification framework of the proposed sentence alignment model has two modes of operation. The first is the training mode where features are extracted from 7,653 manually aligned English-Arabic sentence pairs and are used to train a Multi-Class Support Vector Machine (MSVM) and Hidden Markov Model (HMM) classifiers. THe second is the testing mode where features are extracted from the testing data and are aligned using the previously trained models. Alignment is done using a block of 3 sentences for each language. After aligning a source language sentence and target language sentence, the next 3 sentences are then looked at. We have used 18 input units and 8 output units for MSVM and HMM. Each input unit represents one input feature as in [1]. The input feature vector X is as follows:

$$X = [\ \frac{L(S1a)}{L(S1e)},\ \frac{L(S1a)+L(S2a)}{L(S1e)},\ \frac{L(S1a)}{L(S1e)+L(Se2)},\ \frac{L(S1a)+L(S2a)}{L(S1e)+L(S2e)},$$

$$\frac{L(S1a)+L(S2a)+L(S3a)}{L(S1e)},\ \frac{L(S1a)}{L(S1e)+L(S2e)+L(S3e)},\ \gamma(S1a,S1e),$$

$$\gamma(S1a,S2a,S1e),\ \gamma(S1a,S1e,S2e),\ \gamma(S1a,S2a,S1e,S2e),\ \gamma(S1a,S2a,S3a,S1e),$$

$$\gamma(S1a,S1e,S2e,S3e),\ Cog(S1a,S1e),\ Cog(S1a,S2a,S1e),\ Cog(S1a,S1e,S2e),$$

$$Cog(S1a,S2a,S1e,S2e),\ Cog(S1a,S2a,S3a,S1e),\ Cog(S1a,S1e,S2e,S3e)\ ]$$

Where:

$L(X)$ is the length in characters of sentence X (for instance $\frac{L(S1a)}{L(S1e)}$ is the length ratio between the first Arabic sentence and the first English sentence in a certain block); $\gamma(X,Y)$ is the punctuation compatibility factor between sentences $X\ and\ Y$ (for instance $\gamma(S1a,S1e)$ is the punctuation compatibility factor between the first Arabic sentence and the first English sentence in a certain block). $Cog(X,Y)$ is the cognate factor between sentences $X\ and\ Y$ (for instance $Cog(S1a,S1e)$ is the cognate factor between the first Arabic sentence and the first English sentence in a certain block).

The output is from 8 categories, which are specified as follows: $S1a \rightarrow 0$ means that the first Arabic sentence has no English match.

$S1e \rightarrow 0$ means that the first English sentence has no Arabic match. Similarly, the remaining outputs are $S1a \rightarrow S1e$, $S1a+S2a \rightarrow S1e$, $S1a \rightarrow S1e+S2e$, $S1a+S2a \rightarrow S1e+S2e$, $S1a \rightarrow S1e+S2e+S3e$ and $S1a+S2a+S3a \rightarrow S1e$.

3.1 Multi-Class Support Vector Machine (MSVM)

The SVMs were originally designed for binary classification problems. However, when dealing with several classes, one needs an appropriate multi-class method. As two-class or

binary classification problems are much easier to solve, a number of methods have been proposed for its extension to multi-class problems. The one class against the others is exploited in this work. One class against the other methods compares a given class with all the others put together. It basically constructs L (L = 8 in this work) hyperplanes where each hyperplane separates one class from the other classes. In this way, it generates L decision functions and an observation X is mapped to a class with the largest decision function. It is suitable to discuss the basics of binary SVM in the following text.

The Support Vector Machine (SVM) as a classification method has often been found to provide good classification results [26-32]. The SVM approach seeks to find the optimal separating hyperplane between classes by focusing on the training cases that are placed at the edge of the class descriptors. These training cases are called support vectors. This way, not only is an optimal hyperplane fitted, but also less training samples are effectively used; thus high classification accuracy is achieved with small training sets. To introduce the basic principles of SVM, let us consider a supervised binary classification problem.

Assume that the training data are represented by {xi, yi}, i = 1, 2, …, N, and yi $\in$ {-1, +1}, where N is the number of training samples, yi=+1 for class ω1 and yi=-1 for class ω2. Suppose the two classes are linearly separable so that it is possible to find at least one hyperplane defined by a vector w with a bias w0, which can separate the classes without error:

$$f(x) = w.x + w_0 = 0 \tag{4}$$

To find such a hyperplane, w and w0 should be estimated under the condition of $y_i(w.x_i + w_0) \geq +1$ for yi = +1 (class ω1) and $y_i(w.x_i + w_0) \leq -1$ for yi = -1 (class ω2). The previously mentioned two equations can be combined as follows:

$$y_i(w.x_i + w_0) - 1 \geq 0 \tag{5}$$

Many hyperplanes could be fitted to separate the two classes. The goal is to search for the optimal hyperplane that leaves the maximum margin between classes. The support vectors lie on two hyperplanes, which are parallel to the optimal and are given by:

$$w.x_i + w_0 = \pm 1 \tag{6}$$

If a simple rescale of the hyperplane parameters w and w0 takes place, the margin can be expressed as: $\frac{2}{\|w\|}$

The optimal hyperplane can be found by solving the following optimization problem:

$$\text{Minimize } \frac{1}{2}\|w\|^2 \tag{7}$$

subject to equation 5.

Using a Lagrangian formulation, the above problem can be expressed as follows:

$$\text{Maximize} \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j (x_i . x_j) \tag{8}$$

Subject to: $\sum_{i=1}^{N} \lambda_i y_i = 0 \, and \, \lambda_i \geq 0, i = 1, 2, ...N$

where λ_i are the Lagrange multipliers.

Therefore, the optimal hyperplane discriminant function becomes:

$$f(x) = \sum_{i \in S} \lambda_i y_i (x_i x) + w_0 \tag{9}$$

where S is a subset of training samples that correspond to nonzero Lagrange multipliers.

Normally, classes are not linearly separable. Hence the constraint of equation 5 cannot be satisfied. In this case, a cost function can be formulated to combine the maximization of margin and minimization of error criteria by using a set of variables ξi. This cost function is defined as:

$$\text{Minimize} \quad J(w, w_0, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i \tag{10}$$

Subject to: $y_i (w.x + w_0) \geq 1 - \xi_i$

The previously mentioned realization is called C-SVC. Another possible realization is called the nu-SVC of a soft margin variant of the optimal hyperplane, which uses the nu parameterization. In it, the parameter C is replaced by a parameter $nu \in [0,1]$ which is the lower and upper bound on the number of examples that are support vectors and that lie on the wrong side of the hyperplane, respectively.

The previous method can be generalized for the non-linear discriminant functions by considering the fact that the inner product of the vectors in the mapping space can be expressed as a function of the inner products of the corresponding vectors in the original space. The inner product operation can be expressed as follows:

$$\phi(x)\phi(z) = K(x, z) \tag{11}$$

where K(x,z) is called the kernel function. There are 4 types of kernels:

Linear: $K(x_i, x_j) = x_i^T x_j$

Polynomial: $K(x_i, x_j) = (\gamma . x_i^T x_j + r)^d, \gamma > 0$

Radial Basis Function: $K(x_i, x_j) = \exp(-\gamma . \|x_i - x_j\|^2), \gamma > 0$

Sigmoid: $K(x_i, x_j) = \tanh(\gamma . x_i^T x_j + r)$

where γ, r and d are the kernel parameters. We have exploited the Polynomial type.

The dual optimization problem can be expressed as follows:

$$\text{Maximize} \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j K(x_i . x_j) \tag{12}$$

Subject to: $\sum_{i=1}^{N} \lambda_i y_i = 0 \, and \, \lambda_i \geq 0, i = 1, 2, ...N$

The resulted classifier equation is expressed as follows:

$$f(x) = \sum_{i \in S} \lambda_i y_i K(x_i x) + w_0 \tag{13}$$

The classification process is based on one against the other approaches by using the above equation.

3.2. Hidden Markov Model (HMM)

The use of the Hidden Markov Model as a classification tool is motivated by the fact that it is very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications [33]. The discriminative power grows from its ability to learn the sequential evolution of the observation. Therefore, the degree of sequential structure can be encoded.

A block of 3 sentences for each language may be classified as one of eight types as mentioned in Section 3. Sentence alignment prediction is about trying to guess what the next alignment category type is, based on the observations of the block of a sentence's feature parameters. The statistical model for sentence alignment category prediction is constructed. We collect statistics on what the category Cn is like depending on what the previous categories (Cn-1, Cn-2 …) were. Therefore, the following conditional probability is considered:

$$P(C_n \mid C_{n-1}, C_{n-2}, ...C_1) \tag{14}$$

Using the Markov assumption to simplify the above probability:

$$P(C_n \mid C_{n-1}, C_{n-2}, ...C_1) = P(C_n \mid C_{n-1}) \tag{15}$$

The probability of a certain sequence {C1, C2, …, Cn} may also be expressed by using the Markov assumption as follows:

$$P(C_1, C_2, ..., C_n) = \prod_{i=1}^{n} P(C_i \mid C_{i-1}) \tag{16}$$

Equation 16 is the Markov model. The only information of a certain block are the 18 feature parameters mentioned in Section 2. These feature parameters are observable while the actual alignment category is hidden. Finding the probability of the certain category of Ci can only be

based on the observation Xi. This conditional probability $P(C_i \mid X_i)$ can be written according to Bayes' rule:

$$P(C_i \mid X_i) = \frac{P(X_i \mid C_i)P(C_i)}{P(X_i)} \tag{17}$$

Or for n alignment categories {C1,…,Cn}, as well as n feature sequence {X1,…,Xn}, This conditional probability can be:

$$P(C_1,...,C_n \mid X_1,...,X_n) = \frac{P(X_1,...,X_n \mid C_1,...,C_n)P(C_1,...,C_n)}{P(X_1,...,X_n)} \tag{18}$$

The probability $P(X_1,...,X_n \mid C_1,...,C_n)$ can be estimated as $\prod_{i=1}^{n} P(X_i \mid C_i)$, if we assume that, for all i, the Ci and Xi are independent of all Xj and Cj, for all $j \neq i$.

We want to draw conclusions from our observations about the alignment category. We can therefore omit the probability $P(X_1,...,X_n)$. We get a measure for the probability, which is proportional to the likelihood L as follows:

$$P(C_1,...,C_n \mid X_1,...,X_n) \alpha L(C_1,...,C_n \mid X_1,...,X_n) = P(X_1,...,X_n \mid C_1,...,C_n).P(C_1,...,C_n) \tag{19}$$

With the Markov assumption it turns to:

$$L(C_1,...,C_n \mid X_1,...,X_n) = \prod_{i=1}^{n} P(X_i \mid C_i).\prod_{i=1}^{n} P(C_i \mid C_{i-1}) \tag{20}$$

The classification of alignment category is based on the above likelihood value.

4. ENGLISH-ARABIC CORPUS

Although, there are very popular Arabic-English resources among the statistical machine translation community that may be found in some projects such as the "DARPA TIDES program, http://www.ldc.upenn.edu/Projects/TIDES/," we have decided to construct our Arabic-English parallel corpus from the Internet to have significant parallel data from different domains. The approach of [10] is used to construct the English-Arabic corpus.

Based on [10], Resnik [34], used a technique called STRAND (Structural Translation Recognition, Acquiring Natural Data) to collect parallel text from the Internet archive. The drawback of this approach is that a lot of significant parallel documents that exist on the Internet are not of the same structure. We avoided this problem when we extracted English/Arabic bitexts from the Internet archive as follows:

Three steps are required to find parallel documents:

1- Locating the pages that might contain parallel documents,
2- Generating the document pairs that might be a translation of each other,
3- Filtering-out of non-translation candidate pairs.

Unlike Resnik [34], who used the AltaVista search engine, Kraaij [35] used more than one search engine and his mining process is divided into two main steps: identification of candidate parallel pages, and verification of their parallelism. We have used different Internet search engines to collect bi-texts from different domains too. We simply sent queries to the different Internet search engines. These queries contained some words like: "Arabic version," "English version," "Arabic," "English," "To Arabic," and "To English," in order to download the pages that might contain English / Arabic parallel documents.

We have collected 652 document pairs that contain 191,623 English sentences and 183,542 Arabic sentences. This collected corpus contains noisy text. Moreover the beginning and end of each paragraph is not easily specified in English and Arabic texts. In order to avoid the accumulation of error during sentence alignment procedures, we have specified some anchor points in the English and Arabic texts based on some words or symbols that appeared at the beginning of some sentences and that had to be reasonably frequent. We checked 15,652 English-Arabic sentences manually to extract a list of anchors. The complete description of the Arabic-English parallel corpus can be found in [10].

5. EXPERIMENTAL RESULTS

5.1. Length based approach

Let's consider Gale & Church's length based approach as explained in [1]. We constructed a dynamic programming framework to conduct experiments using their length based approach as a baseline experiment to compare with our proposed system. First of all it was not clear to us which variable should be considered as the text length, character, or word. To answer this question, we had to do some statistical measurements on 1,000 manually aligned English - Arabic sentence pairs that were randomly selected from the previously mentioned corpus. We considered the relationship between English paragraph length and Arabic paragraph length as a function of the number of words. The results showed that there is a good correlation (0.987) between English paragraph length and Arabic paragraph length. Moreover, the ratio and corresponding standard deviation were 0.9826 and 0.2046 respectively. We also considered the relationship between English paragraph length and Arabic paragraph length as a function of the number of characters.

The results showed a better correlation (0.992) between English paragraph length and Arabic paragraph length. Moreover, the ratio and corresponding standard deviation were 1.12 and 0.1806 respectively. In comparison to the previous results, the number of characters as the text length variable is better than words since the correlation is higher and the standard deviation is lower.

We applied the length based approach (using text length as a function of the number characters) experiment on a 1,200 sentence pair sample, which was not taken from the training data. Table 1 shows the results. In Section 3, the categories $S1a \rightarrow 0$ and $S1e \rightarrow 0$ are mapped as 1-0, 0-1 in the table. Similarly, $S1a \rightarrow S1e$ is mapped as 1-1 , $S1a + S2a \rightarrow S1e$

and $S1a \rightarrow S1e + S2e$ are mapped as 2-1 and 1-2, $S1a + S2a \rightarrow S1e + S2e$ is mapped as 2-2, $S1a \rightarrow S1e + S2e + S3e$ and $S1a + S2a + S3a \rightarrow S1e$ are mapped as 1-3 and 3-1. The first column in Table 1 represents the category, the second column is the total number of sentence pairs related to this category, the third column is the number of sentence pairs that were misclassified, and the fourth column is the percentage of this error. Although 1-0, 0-1, and 2-2 are rare cases, we have taken them into consideration to reduce errors. Moreover, we did not consider other cases like (3-3, 3-4, etc.) since they are very rare cases and considering more cases requires more computations and processing time. When the system finds these cases, it misclassifies them.

5.2. Multi-Class Support Vector Machine (MSVM)

The system extracted features from 7,653 manually aligned English-Arabic sentence pairs and used them to train a Multi-Class Support Vector Machine model. 1,200 English-Arabic sentence pairs were used as the testing data. These sentences were used as inputs for the Multi-Class Support Vector Machine after the feature extraction step. Alignment was done using a block of 3 sentences for each language. After aligning a source language sentence and target language sentence, the next 3 sentences were then looked at as follows:

Extract features from the first three English sentences and do the same with the first three Arabic sentences.

Construct the feature vector X.

Use this feature vector as an input of the Multi-Class Support Vector Machine model.

According to the model output, construct the second feature vector. For instance, if the result of the model is $S1a \rightarrow 0$, then read the fourth Arabic sentence and use it with the second and third Arabic sentences with the first three English sentences to generate the feature vector X.

Continue using this approach until there are no more English-Arabic text pairs.

Table 2 shows the results when we applied this approach on the 1,200 English - Arabic sentence pairs. It is clear from Table 2 that the results have been improved in terms of accuracy over the length based approach. Additionally, we applied the MSVM approach to the entire English-Arabic corpus, which contained 191,623 English sentences and 183,542 Arabic sentences. Then we randomly selected 500 sentence pairs from the sentence aligned output file and manually checked them. The system reported a total error rate of 4.6%.

We decreased the number of sentence pairs used for training the MR model to 4,000 sentence pairs to investigate the effect of the training data size on the total system performance. These 4,000 sentence pairs were randomly selected from the training data. The constructed model was then used to align the entire English-Arabic corpus. Then, we randomly selected 500 sentence pairs from the sentence aligned output file and manually checked them. The system reported a total error rate of 4.8%. The reduction of the training data set does not significantly change total system performance.

5.3. Hidden Markov Model (HMM)

The system extracted features from 7,653 manually aligned English-Arabic sentence pairs and used them to construct a Hidden Markov Model. 1,200 English-Arabic sentence pairs were used as the testing data. Using formula (20) and the 5 steps mentioned in the previous section (using HMM instead of MSVM) the 1,200 English-Arabic sentence pairs were aligned. Table 3 shows the results when we applied this approach. Additionally, we applied the HMM approach to the

entire English-Arabic corpus. Then, we randomly selected 500 sentence pairs from the sentence aligned output and manually checked them. The system reported a total error rate of 4.1%.

We decreased the number of sentence pairs used for training the HMM to 4,000 sentence pairs, as in the previous section, to investigate the effect of the training data size on total system performance. These 4,000 sentence pairs were randomly selected from the training data. The constructed model was then used to align the entire English-Arabic corpus. Then, we randomly selected 500 sentence pairs from the sentence aligned output and manually checked them. The system reported a total error rate of 4.2%. The reduction of the training data set from 7,653 to 4,000 does not significantly change the total system performance.

5.4. Discussion

The length based approach did not give bad results, as shown in Table 1 (the total error rate was 6.4%). This is explained by the fact that there is a good correlation (0.992) and low standard deviation (0.1806) between English and Arabic text lengths. These factors lead to good results as shown in Table 1. The Multi-Class Support Vector Machine approach decreased the total errors by 29.7% and the Hidden Markov Model approach decreased it by 35.9%, as compared to the length based approach as shown in Tables 1, 2 and 3. The Multi-Class Support Vector Machine and Hidden Markov Model approaches are quite flexible and they open the door to many other models that can be used for sentence alignment problems.

Using feature extraction criteria allows researchers to use different feature values when trying to solve natural language processing problems in general.

Table 1. The results from using the length based approach

Category	Frequency	Error	% Error	95% Confidence interval
1-1	1,099	54	4.9%	3.68%, 6.12%
1-0, 0-1	2	2	100%	99.44%, 100%
1-2, 2-1	88	14	15.9%	13.83%, 17.97%
2-2	2	1	50%	47.17%, 52.83%
1-3, 3-1	9	6	66%	63.32%, 68.68%
Total	1,200	77	6.4%	5.02%, 7.78%

Table 2. The results from using the Multi-Class Support Vector Machine approach

Category	Frequency	Error	% Error	95% Confidence interval
1-1	1,099	35	3.2%	2.2%, 4.2%
1-0, 0-1	2	2	100%	99.44%, 100%
1-2, 2-1	88	11	12.5%	10.63%, 14.37%
2-2	2	1	50%	47.17%, 52.83%
1-3, 3-1	9	5	55.5%	52.69%, 58.31%
Total	1,200	54	4.5 %	3.33%, 5.67%

Table 3. The results from using the Multi-Class Support Vector Machine approach

Category	Frequency	Error	% Error	95% Confidence interval
1-1	1,099	31	2.8%	1.87%, 3.73%
1-0, 0-1	2	2	100%	99.44%, 100%
1-2, 2-1	88	10	11.4%	9.6%, 13.2%
2-2	2	1	50%	47.17%, 52.83%
1-3, 3-1	9	5	55.5%	52.69%, 58.31%
Total	1,200	49	4.1%	2.98%, 5.22%

6. CONCLUSIONS

In this paper, we investigated the use of the Multi-Class Support Vector Machine and the Hidden Markov Model for sentence alignment. We have applied our new approaches on a sample of an English - Arabic parallel corpus. Our approach results outperform the length based approach. The proposed approaches have improved the total system performance in terms of effectiveness (accuracy). Our approaches have used the feature extraction criteria, which gives researchers an opportunity to use many varieties of these features based on the language pairs used and their text types (Hanzi characters in Japanese-Chinese texts may be used for instance).

REFERENCES

[1] M. Fattah, F. Ren, S. Kuroiwa, "*Sentence Alignment using P-NNT and GMM*," Computer Speech and Language, Vol.21, No.4, 2007, pp.594-608.

[2] R. Moore, "*Fast and Accurate Sentence Alignment of Bilingual Corpora,*" AMTA, 2002, pp.135-144.

[3] F. Gey, A. Chen, M. Buckland, R. Larson, "*Translingual vocabulary mappings for multilingual information access*," SIGIR, 2002, pp.455-456.

[4] M. Davis, F. Ren, "*Automatic Japanese-Chinese Parallel Text Alignment*," Proceedings of International Conference on Chinese Information Processing, 1998, pp.452-457.

[5] W. Dolan, J. Pinkham, S. Richardson, "MSR-*MT, The Microsoft Research Machine Translation System*," AMTA, 2002, pp.237-239.

[6] M. Simard, "*Text-translation alignment: three languages are better than two*" In Proceedings of EMNLP/VLC- 99, College Park, MD, 1999.

[7] A. Chen, F. Gey, "*Translation Term Weighting and Combining Translation Resources in Cross-Language Retrieval*" TREC 2001.

[8] D. Oard, "*Alternative approaches for cross-language text retrieval*": In D. Hull, & D. Oard (Eds.), AAAI symposium in cross-language text and speech retrieval. American Association for Artificial Intelligence, March, 1997.

[9] C. Christopher, L. Kar, "*Building parallel corpora by automatic title alignment using length-based and text-based approaches*" Information Processing and Management ,Vol.40, 2004, pp.939-955.

[10] M. Fattah, F. Ren, S. Kuroiwa, "*Stemming to Improve Translation Lexicon Creation form Bitexts*" Information Processing & Management, Vol.42 No.4, 2006, pp.1003-1016.

[11] S. Ker, J. Chang, "*A class-based approach to word alignment*", Computational Linguistics, Vol.23, No.2, 1997, pp.313-344.

[12] I. Melamed, "*A portable algorithm for mapping bitext correspondence*" In T*he 35th Conference of the Association for Computational Linguistics* (ACL 1997), Madrid, Spain, 1997.

[13] H. Dejean, É. Gaussier, F. Sadat, "*Bilingual Terminology Extraction: An Approach based on a Multilingual thesaurus Applicable to Comparable Corpora*", Proceedings of the *19th International Conference on Computational Linguistics COLING 2002*, Taipei, Taiwan, 2002, pp.218-224.

[14] C. Thomas, C. Kevin, "*Aligning Parallel Bilingual Corpora Statistically with Punctuation Criteria*", *Computational Linguistics and Chinese Language* Processing , Vol.10, No.1, 2005, pp.95-122.

[15] W. Gale, K. Church, "*A program for aligning sentences in bilingual corpora*" *Computational Linguistics*, Vol.19, 1993, pp.75-102.

[16] P. Brown, J. Lai, R. Mercer, "*Aligning sentences in parallel corpora*" In Proceedings of the *29th annual meeting of the association for computational linguistics*, Berkeley, CA, USA, 1991.

[17] M. Simard, G. Foster, P. Isabelle, "*Using cognates to align sentences in bilingual corpora*", Proceedings of TMI92, Montreal, Canada, 1992, pp.67-81.

[18] I. Melamed, "*Bitext Maps and Alignment via Pattern Recognition*", *Computational Linguistics*, March, Vol.25, No.1, 1999, pp.107-130.

[19] P. Danielsson, K. Mühlenbock, "*The Misconception of High-Frequency Words in Scandinavian*

Translation", AMTA, 2000, pp.158-168.
[20] A. Ribeiro, G. Dias, G. Lopes, J. Mexia, "*Cognates Alignment: In Bente Maegaard (ed.)*", Proceedings of the Machine Translation Summit VIII (MT Summit VIII) - Machine Translation in the Information Age, Santiago de Compostela, Spain, 2001, pp.287-292.
[21] A. Ceauşu, D. Ştefănescu, D. Tufiş, "*Acquis communautaire sentence alignment using support vector machines*", Proceedings of the *Fifth Language Resources and Evaluation Conference*, 2006.
[22] S. Vogel, H. Ney, C. Tillmann, "*HMM-Based Word Alignment in Statistical Translation*", Proceedings of the *16th International Conference on Computational Linguistics, Copenhagen*, Denmark, 1996, pp.836-841.
[23] N. Collier, K. Ono, H. Hirakawa, "*An Experiment in Hybrid Dictionary and Statistical Sentence Alignment*" COLING-ACL, 1998, pp.268-274.
[24] K. Chen, H. Chen, "*A Part-of-Speech-Based Alignment Algorithm*", Proceedings of *15th International Conference on Computational Linguistics*, Kyoto, 1994, pp.166-171.
[25] S. Chen, "*Aligning Sentences in Bilingual Corpora Using Lexical Information*", Proceedings of *ACL-93, Columbus OH*, 1993, pp.9-16.
[26] S. Mukherjee, E. Osuna, F. Girosi, "*Nonlinear prediction of chaotic time series using support vector machine*", In proceedings of the *IEEE Workshop on Neural Networks for Signal Processing 7*, Amerlia Island, FL, 997, pp.511-519
[27] E. Osuna, R. Freund, F. Girosi, "*An improved training algorithm for support vector machines*", In Proc. of *the IEEE Workshop on Neural Networks for Signal Processing VII*, New York, 1997, pp.276-285.
[28] M. Brown, H. Lewis, S. Gunn, "*Linear Spectral Mixture Models and Support Vector Machines for Remote Sensing*", *IEEE Transactions On Geoscience And Remote Sensing*, Vol.38, No.5, 2000, September.
[29] G. Foody, A. Mathur, "*A Relative Evaluation of Multiclass Image Classification by Support Vector Machines*", *IEEE Transactions On Geoscience And Remote Sensing*, Vol.42, No.6, 2004, June.
[30] Q. She, H. Su, L. Dong, J. Chu, "*Support vector machine with adaptive parameters in image coding*", *Int. J. Innovative Computing, Information and Control*, Vol.4, No.2, 2008, pp.359-368.
[31] R. Chen, S. Chen, "*Intrusion detection using a hybrid support vector machine based on entropy and TF-IDF*", *Int. J. Innovative Computing, Information and Control*, Vol.4, No.2, 2008, pp.413-424.
[32] X. Song, , W. Chen, B. Jiang, "*Sample Reducing Method in Support Vector Machine Based on K-Closest Sub-Clusters*", *Int. J. Innovative Computing, Information and Control*, Vol.4, No.7, 2008, pp.1751-1760.
[33] L. Rabiner, "*A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*", *Proceddings of the IEEE*, Vol.77, No.2, 1989, pp.257-286.
[34] P. Resnik, N. Smith, "*The Web as a Parallel Corpus*", *Computational Linguistics,* Vol.29, No.3, 2003, pp.349-380.
[35] W. Kraaij, J. Nie, M. Simard, "*Embedding Web-Based Statistical Translation Models in Cross-Language Information Retrieval*", *Computational Linguistics, Vol.29*, No.3, 2003, pp.381-419.

Mohamed Abdel Fattah
received his B.E. degree in 1994 and M.E. degree in 2003 from the Department of Electronics & Communications in the Faculty of Engineering at Cairo University in Egypt. He received his Ph.D. degree in 2007 from the Department of Information Science and Intelligent Systems in the Faculty of Engineering at the University of Tokushima, Japan. Now he is a member of the Faculty of Industrial Education at Helwan University in Cairo, Egypt. His current research interests include Natural Language Processing, Information Retrieval, Speech Recognition, and Speaker Recognition.