Implementation of Word Alignment System for English- Myanmar Machine Translation

Ei Ei Khaing, Dr. Khin Mar Soe
eieikhaing98@gmail.com, kmsucsy@gmail.com
University of Computer Studies, Yangon

Abstract

Statistical word alignment models have been widely used for various Natural Language Processing (NLP) problem. In statistical machine translation, word alignment models are trained on bilingual corpora. To build an SMT system we require bitext and a word alignment of that bitext, as well as language models built from target language data. A word alignment for a parallel sentence pair represents the correspondence between words in a source language and their translations in a target language. This system will use the IBM model which is based on the EM algorithm. This system deals with the step of word alignment. In this paper, C# implementation of a word alignment algorithm is used to testing the source and target sentences. This system also uses a English-Myanmar dictionary to bootstrap the Expectation Maximization (EM) algorithm.

Keywords*: Word alignment, statistical machine translation, IBM model, EM algorithm.*

1. Introduction

Bilingual word alignment is the first step of most current approaches to Statistical Machine Translation or SMT [3]. Most of the SMT systems usually have two stages. The first stage is called language modeling. One simple and very old but still quite useful approach for language modeling is *n*-gram modeling. Separate language models are built for the source language (SL) and the target language (TL). For this stage, monolingual corpora of the SL and the TL are required. The second stage is called translation modeling and it includes the step of finding the word alignments induced over a sentence aligned bilingual (parallel) corpus. This paper deals with the step of word alignment, which is sometimes extended to phrase alignment.

The paper is structured as follows. Section 1 introduces the word alignment system. Section 2 describes the background theory of the system. Section-3 describes some related work. The implementation of EM algorithm with English-Myanmar Word Alignment System is presented in Section-4. In Section-5, experimental results are presented.

2. Theoretical Background

In recent years statistical word alignment models have been widely used for various Natural Language Processing (NLP) problems.

Statistical Machine Translation (SMT) models define a way to explain how sentences get translated. To formalize this one can use Bayes' law to rewrite the task of finding the most probable target language translation given the source language at hand. For the translation model one can assume that word alignments *a* exist between the words of the source and the target sentence *e*. These alignments can be intuitively understood as links between single words of the source and target sentence, which indicate that the linked words are translations of each other. In the general case, a single source or target word can have alignments to more than one word on the opposite side.

By introducing the alignments, one can reformulate:

$$P(f|e) = \sum_{a} P(f, a|e)$$

A fundamental problem in SMT is then word aligning a bitext. In this case, it is not needed to seek for the most probable translation *e* given *f* because we have access to both. It is only to care about aligning them. The IBM Model 1 for SMT defines a simple way to explain how sentences get translated. The parameterization of the model involves estimating translation probabilities of the type P(f|e) for single words f from the source language and e from the target. Getting a Maximum Likelihood estimate of the parameters from a bitext involves the usage of the Expectation-Maximization algorithm. A nice property of IBM Model 1 is that there is only a single maximum of the likelihood function.

2.1. Definition of Model 1 (Translation Parameter)

Model 1 is a probabilistic generative model within a framework that assumes a source sentence S of length l translates as a target sentence T, according to the following stochastic process:

- A length m for sentence T is generated.
- For each target sentence position $j \in \{1, . .,m\}$

-A generating word s_i in S (including a null word s_0) is selected, and

- The target word t_j at position j is generated depending on s_i.

Model 1 estimate for the probability of a target sentence, given a source sentence. An alignment *a* specifies which English word each Myanmar word was generated from. Thus, there are (l *m) possible alignments.

2.2. Model-2: Distortion or Alignment Parameter

Given source and target sentence lengths l and m, probability that j^{th} target word is connected to i^{th} source word, the distortion probability is given as D (i | j, l , m). IBM Model 2 builds up from Model 1 by adding alignment probabilities.

2.3. Model-3: Fertility Parameter

We can generate the target sentence from English sentence with the probability p(h, a | e). In the third model, this probability is calculated using a new parameter called fertility Φ, where $F(e|\Phi)$= probability that e is aligned with target words.

2.4. Problem Statements

Based on IBM model, the problem of word alignment is divided into several different problems.

The first problem: is to find the most likely translations of an SL word, irrespective of positions.

This part is taken care of by the translation model. This model describes the mathematical relationship between two or more languages. The main thing is to predict whether expressions in different languages have equivalent meanings.

The second problem: is to align positions in the SL sentence with positions in the TL sentence. This problem is addressed by the distortion model. It takes care of the differences in word orders of the two languages. A novel metric to measure word order similarity (or difference) between any pair of languages based on word alignments.

The third problem: is to find out how many TL words are generated by one SL word. Note that an SL word may sometimes generate no TL word, or a TL word may be generated by no SL word (NULL insertion). The fertility model is supposed to account for this. The first three models corresponding to these problems form the core of the IBM model based generate SMT.

Most of the SMT systems usually have two stages based on the models. The first stage is called language modeling. One simple and very old but still quite useful approach for language modeling is *n*-gram modeling. Separate language models are built for the source language (SL) and the target language (TL). For this stage, monolingual corpora of the SL and the TL are required. The second stage is called translation modeling and it includes the step of finding the word alignments induced over a sentence aligned bilingual (parallel) corpus. The Expectation-Maximization (EM) algorithm is used to iteratively estimate alignment model probabilities according to the likelihood of the model on a parallel corpus. In the Expectation step, alignment probabilities are computed from the model parameters and in the Maximization step, parameter values are re-estimated based on the alignment probabilities and the corpus. The iterative process is started by initializing parameter values with uniform probabilities for IBM Model 1. The EM algorithm is only guaranteed to find a local maximum which makes the result depend on the starting point of the estimation process. This system is implemented EM algorithm and deals with problem statements.

2.4.1 The Advantages of the Models

Model 1 and 2 simplifies how the source string was generated from the target string, while model 3 adds several new parameters to the alignment model. In the simplest of the IBM-models, Model 1 only depends on one parameter, the translation probability that the target word aligned to the source word at position j is a translation of the words.

Model 2 includes a parameter for alignment positions where the position of the target word depends on the position of the source word, the length of the target sentence and the length of the source sentence. In this model, the alignment depends on the source and target words as well as the absolute position of the source word.

Model 3 assumes that source words can be generated from a NULL word token at each position in the target sentence. The probability of generating such a NULL word is also used.

3. Related Work

In 1991, Gale and Church [4] introduced the idea of using measures of association for finding translations of words based on information in parallel text. They begin by carrying out sentence alignment, which is the problem of determining which sentences are translations of each other. In fact this is a much simpler problem than finding the translations of words, since long sentences in one language tend to translate as long sentences in another language, and the order in which sentences appear doesn't usually change radically in a translation. The original K-vec algorithm proposed by Fung and Church [2] works only for parallel corpus and makes use of the word position and frequency feature to find word correspondences. K-vec uses tests of association as a

similarity measure, while the 1995 approach of Fung [1] uses Euclidean distance. Like K-vec this approach is also language independent and works for different language pairs. Fung and Yee [3] also proposed an IR approach for translating new words from non-parallel comparable texts. Ittycheriah and Roukos [5] proposed a maximum entropy word aligner for Arabic-English machine translation. Malouf [6] compared several algorithms for maximum entropy parameter estimation. Martin et al. [7] have discussed word alignment for languages with scarce resources. Moore et al. [8] proposed a discriminative framework for bilingual word alignment.

4. Design and Implementation

For calculating the parameters mentioned in session 2 (translation, distortion and fertility) it can be used a generative algorithm called Expectation Maximization (EM) for training [9].

The EM algorithm guarantees an increase in likelihood of the model in each iteration, i.e., it is guaranteed to converge to a maximum likelihood estimate. A set of sentence aligned parallel corpus is used as the training data. Let the number of sentence pairs in the training data be N and the lengths of the source and target sentences be *s* and *t*, respectively. Iterative EM algorithm corresponding to the translation problem can be described as:

Step-1: Collect all word types from the source and target corpora.
For each source word e collect all target words *m* that co-occur at least once with *e*.

Step-2: Initialize the translation parameter uniformly (uniform probability distribution), i.e., any target word probably can be the translation of a source word e. In this system, there are three main steps for aligning the source and target sentences. The detail algorithm can be seen as shown in below. Figure 4.1 shows the algorithm for pre-processing phase. Figure 4.2 is the algorithm for problem statement 1 and Figure 4.3 is the algorithm for problem statement 2 while aligning the source and target words.

```
Pre-processing Phase
 Accept Source Sentence;
  Accept Target Sentence;
    Remove Stop Word in Source Words (e)
  For each Source Sentence S do
    Separate into words;
    Store Source Words Indexes;
  End For
 For each Target Sentence T do
    Separate into words;
    Store Target Words Indexes;
 End For
```

Figure 4.1 Algorithm for Pre-processing

```
Align Text Problem (1) Phase
 Step-1:  Collect all word types from the source and target
          corpora.
          For each source word s collect all target words t
           that co occur at least once with e.
//Example, source words(e) I = ကျွန်တော်  Target words (m)
 Step-2:  Any target word (m) probably can be the translation of
         a source word (e).
         Initialize the expected translation count tc to 0
 Step-3:  Iteratively refine the translation probabilities.
          For i=1 to s do
            Select Target Words FROM Dictionary Table
              WHERE   Source Equals e_i
            For j=1 to t do
              If T(m_j) is similar to Target Word in Dictionary
              Store Target Word Index from Target Words (m)
               Assign current Target Word into null
               Store Source Word Index from Source Words (s)
               Assign current Source Word into null
              Insert Source Sentence, Target Sentence and Word
              Indexes into Training Table
             End If
          End For
          Calculate Probability T
       End For
```

Figure 4.2 Algorithm for Problem Statement 1

```
Align Text Problem (2) Phase
Step-1:  Accept Input Source Sentence and Source Sentence
        in Training Table
Step-2:  Initialize matching probability Count for
        Training data is 0
Step-3:  For each Source Words (e) and Target Words (m) in
Input Sentences
              If match with Words in Training Sentence
                Count ++;
              End If
               Probability = Count / Total Number Words;
         If (probability >= 0.7) Retrieve Aligned Training Data
        End If
      Store  Input Source Sentence into the Training Table
   End For
```

Figure 4.3 Algorithm for Problem Statement 2

4.1. System Design

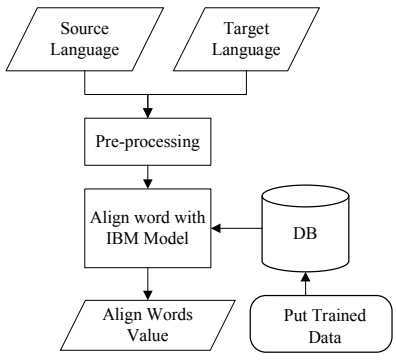

Figure 4.4 System Overview

The overview of the system is depicted in Figure 4.4. The source and target language are inputs of this system. The preprocessing step is used to segment the words and the alignment step is used with corpus database. The system flowchart is depicted as shown in Figure 4.5.

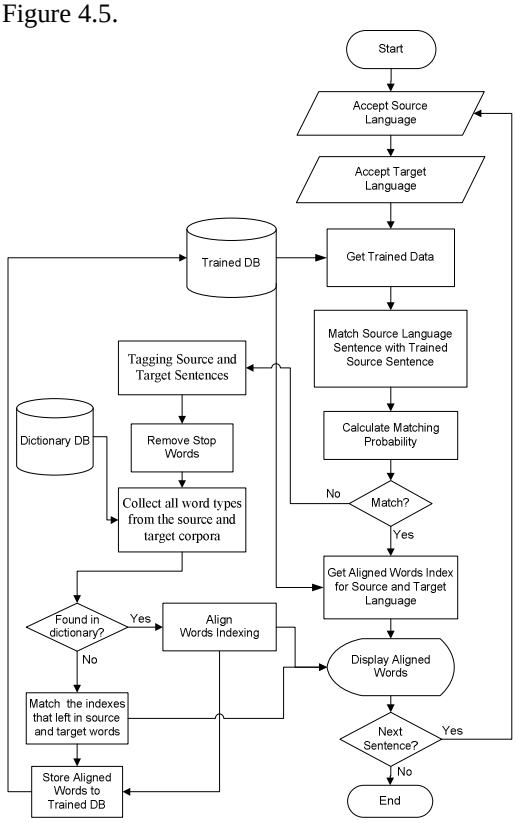

Figure 4.5 System Flowchart

5. Testing Results

This system is tested based on the dictionary and training data set as well as unknown word occurrence. For predefined part of speech are used for test cases are S+V+Adj, S+V(be)+N, S+V+N, S+V, There+V(be)+N, S+V+IO+DO, S+V(be)+ Complement , and from Grade 2 English Text Book. Firstly, when it is used with training dataset, it can be seen the align word results as shown in below:

Align words:

My father is a doctor.

u:ey\ zcion q&m0e wpa,mu |zpon/

Secondly, when it is used with dictionary, the align word results as shown in below:

Align words:

The birds sing sweetly.

Xurm;on omom,m,mweu&;on/

Thirdly, when it is used with dictionary and unknown word ‘blackboard’ is not included in dictionary, the align word results as shown in below and unknown word can be saved in the dictionary table.

Align words:

Look at the blackboard.

aumuoiye;u |un!y#/

The resulted alignment words and their links (indexes) are stored in training table for further usage of checking with training data. For example, the align table for checking with dictionary as shown in Table [5.1].

Table [5.1] Align Table

Source Words	Word Index	Target Words	Word Index
the	0	ဌက်များသည်	0
birds	1	သာသာယာယာ	1
sing	2	တွန်ကျူးသည်	2
sweetly	3		

T(m | e) = 1/(number of co-occurring target words)

Where T is the translation probability Myanmar given English.

T(ဌက်များ | birds) = 1/1 =1

T(တွန်ကျူးသည် | sing) = ½ =0.5

There are two meanings for the word ‘sing’ which are "သီချင်းဆိုသည် , တွန်ကျူးသည် ". Then, alignment process iteratively refines the translation probabilities until values are good enough. The alignment values can be calculated by looking at the individual translation probability values. The best alignment can be calculated in a quadratic number of steps equal to *(sl+1)×tl*. 1 is used to add for the NULL value. For example for the above sentence pairs,

sl=4, *tl*=3; (*sl* + 1) x *tl* = (4+1) x 3 =15 steps,

where *sl=source sentence's length, tl=target sentence's length.*

For the training table matching, the probability is calculated using the input sentences and training datasets. The probability of both source and target values are calculated. For the training table matching with “My father is a teacher” over “My father is a doctor”, both of the probability values for source and target sentences get 0.8. This system only covers with one unknown occurrence with dynamic positions.

$$P(e) = \frac{4}{5} = 0.8 \qquad P(m) = \frac{4}{5} = 0.8$$

Where P(e) is the probability of English words calculated over input sentence and Training Dataset, P(m) is the probability of Myanmar words from input sentence and Training Dataset.

5.1. API for Word Alignment System

The API for this system is as shown in Figure 5.1. This system can result the aligned word pairs with respect to their word segmentations in source and target sentence.

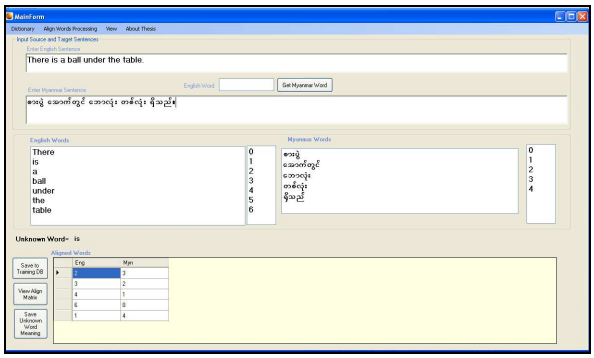

Figure 5.1 system API

6. Conclusion

Describing word alignment is one of the fundamental goals of Statistical Machine Translation (SMT). Alignment specifies the word orders when a sentence is translated into another language. Parameters of a statistical word alignment models are estimated from bitext, and the model is used to generate word alignments over the same bitext. Word alignments can have a strong influence on phrase-based SMT system performance. Most current SMT systems use a generative model for word alignment such as IBM word alignment models. Based on the training table and input dictionary table, this system generates correct alignment words. This system can be extended as phrase alignment. This system limits the predefined seven POS. The POS can be extended as a future work.

References

[1] Pascale Fung. 1995. "Compiling Bilingual Lexicon Entries from a Non-Parallel English-Chinese Corpus", In Third Annual Workshop on Very Large Corpora, Boston, Massachusetts: Jun. 1995. Pages 173-183.

[2] Pascale Fung and Kenneth Ward Church. 1994. K-vec: a new approach for aligning parallel texts. In Proceedings of the 15th conference on Computational linguistics. Pages 1096-1102. Kyoto, Japan.

[3] P. Fung and L. Yee. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*. Pages 414–420.

[4] W. Gale and K. Church. Identifying word correspondences in parallel texts. 1991. In *Proceedings of the Fourth DARPA Speech and Natural Language Workshop*. Pages 152–157, Pacific Grove, CA,

[5] Ittycheriah and S. Roukos. 2005. A maximum entropy word aligner for Arabic-English machine translation. In Proceedings of HLT-EMNLP. Vancouver, Canada. Pages 89–96.

[6] R. Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In Proceedings of CoNLL. Taipei, Taiwan. Pages 49–55.

[7] J. Martin, R. Mihalcea, and T. Pedersen. 2005. Word alignment for languages with scarce resources. In Proceedings of the ACL Workshop on Building and Using Parallel Texts. Ann Arbor, USA. Pages 65–74.

[8] R. C. Moore. 2005. A discriminative framework for bilingual word alignment. In Proceedings of HLT-EMNLP. Vancouver, Canada. Pages 81–88.

[9] G. Chinnappa and Anil Kumar Singh. A Java Implementation of an Extended Word Alignment Algorithm Based on the IBM Models, Language Technologies Research Centre International Institute of Information Technology Hyderabad, India.