Multi-Word Expression-Sensitive Word Alignment

Tsuyoshi Okita[1], **Alfredo Maldonado Guerra**[2], **Yvette Graham**[3], **Andy Way**[1]
{CNGL[1], NCLT[3]} / School of Computing / Dublin City University,
CNGL / School of Computer Science and Statistics / Trinity College Dublin[2]
{tokita,ygraham,away}@computing.dcu.ie, maldonaa@scss.tcd.ie

Abstract

This paper presents a new word alignment method which incorporates knowledge about Bilingual Multi-Word Expressions (BMWEs). Our method of word alignment first extracts such BMWEs in a bidirectional way for a given corpus and then starts conventional word alignment, considering the properties of BMWEs in their grouping as well as their alignment links. We give partial annotation of alignment links as prior knowledge to the word alignment process; by replacing the maximum likelihood estimate in the M-step of the IBM Models with the Maximum A Posteriori (MAP) estimate, prior knowledge about BMWEs is embedded in the prior in this MAP estimate. In our experiments, we saw an improvement of 0.77 Bleu points absolute in JP–EN. Except for one case, our method gave better results than the method using only BMWEs grouping. Even though this paper does not directly address the issues in Cross-Lingual Information Retrieval (CLIR), it discusses an approach of direct relevance to the field. This approach could be viewed as the opposite of current trends in CLIR on semantic space that incorporate a notion of order in the bag-of-words model (e.g. co-occurences).

1 Introduction

Word alignment (Brown et al., 1993; Vogel et al., 1996; Och and Ney, 2003a; Graca et al., 2007) remains key to providing high-quality translations as all subsequent training stages rely on its performance. It alone does not effectively capture many-to-many word correspondences, but instead relies on the ability of subsequent heuristic phrase extraction algorithms, such as grow-diag-final (Koehn et al., 2003), to resolve them.

Some aligned corpora include implicit partial alignment annotation, while for other corpora a partial alignment can be extracted by state-of-the-art techniques. For example, implicit tags such as reference number within the patent corpus of Fujii et al. (2010) provide (often many-to-many) correspondences between source and target words, while statistical methods for extracting a partial annotation, like Kupiec et al. (1993), extract terminology pairs using linguistically predefined POS patterns. Gale and Church (1991) extract pairs of anchor words, such as numbers, proper nouns (organization, person, title), dates, and monetary information. Resnik and Melamed (1997) automatically extract domain-specific lexica. Moore (2003) extracts named-entities. In Machine Translation, Lambert and Banchs (2006) extract BMWEs from a phrase table, which is an outcome of word alignment followed by phrase extraction; this method does not alter the word alignment process.

This paper introduces a new method of incorporating previously known many-to-many word correspondences into word alignment. A well-known method of incorporating such prior knowledge in Machine Learning is to replace the likelihood maximization in the M-step of the EM algorithm with either the MAP estimate or the Maximum Penalized Likelihood (MPL) estimate (McLach-

Proceedings of the 4th International Workshop on Cross Lingual Information Access at COLING 2010, pages 26–34,
Beijing, August 2010

lan and Krishnan, 1997; Bishop, 2006). Then, the MAP estimate allows us to incorporate the *prior*, a probability used to reflect the degree of prior belief about the occurrences of the events.

A small number of studies have been carried out that use partial alignment annotation for word alignment. Firstly, Graca et al. (2007) introduce a posterior regularization to employ the prior that cannot be easily expressed over model parameters such as stochastic constraints and agreement constraints. These constraints are set in the E-step to discard intractable alignments contradicting these constraints. This mechanism in the E-step is in a similar spirit to that in GIZA++ for IBM Model 3 and 4 which only searches around neighbouring alignments around the Viterbi alignment. For this reason, this algorithm is not intended to be used combined with IBM Models 3 and 4. Although theoretically it is possible to incorporate partial annotation with a small change in its code, Graca et al. do not mention it. Secondly, Talbot (2005) introduces a constrained EM method which constrains the E-step to incorporate partial alignment into word alignment,[1] which is in a similar manner to Graca et al. (2007). He conducted experiments using partial alignment annotation based on cognate relations, a bilingual dictionary, domain-specific bilingual semantic annotation, and numerical pattern matching. He did not incorporate BMWEs. Thirdly, Callison-Burch et al. (2004) replace the likelihood maximization in the M-step with mixed likelihood maximization, which is a convex combination of negative log likelihood of known links and unknown links.

The remainder of this paper is organized as follows: in Section 2 we define the anchor word alignment problem. In Section 3 we include a review of the EM algorithm with IBM Models 1-5, and the HMM Model. Section 4 describes our own algorithm based on the combination of BMWE extraction and the modified word alignment which incorporates the groupings of BMWEs and enforces their alignment links; we explain the EM algorithm with MAP estimation with three kinds of priors. In Section 5 our experimental results are presented, and we conclude in Section 6.

pair	GIZA++(no prior)			Ours(with prior)		
EN-FR	fin	ini	prior	fin	ini	prior
is *NULL*	1	.25	0	0	.25	.25
rosy *en*	1	.5	0	0	.5	.2
that .	1	.25	0	0	.25	.25
life *la*	1	.25	0	0	.25	0
. *c‘*	1	.25	0	0	.25	.25
that *c‘*	0	.25	0	1	.25	.25
is *est*	0	.25	0	1	.25	.25
life *vie*	0	.5	0	1	.5	1
rosy *rose*	0	.25	0	1	.25	.2

Table 1: The benefit of prior knowledge of anchor words.

2 Anchor Word Alignment Problem

The input to standard methods of word alignment is simply the sentence-aligned corpus, whereas our alignment method takes in additionally a partial alignment. We assume, therefore, the availability of a partial alignment, for example via a MWE extraction tool. Let $\breve{e}$ denote an English sentence, and e denote an English word, throughout this paper. The anchor word alignment problem is defined as follows:

Definition 1 (Anchor Word Alignment Problem) Let $(\breve{e},\breve{f}) = \{(\breve{e}_1,\breve{f}_1),\ldots,(\breve{e}_n,\breve{f}_n)\}$ be a parallel corpus. By prior knowledge we additionally have knowledge of anchor words $(\hat{e},\hat{f}) = \{(sent_i, t_{e_1}, t_{f_1}, pos_{e_1}, pos_{f_1}, length_e, length_f), \ldots, (sent_k, t_{e_n}, t_{f_n}, pos_{e_n}, pos_{f_n}, length_e, length_f)\}$ where $sent_i$ denotes sentence ID, pos_{e_i} denotes the position of t_{e_i} in a sentence $\breve{e}_i$, and $length_e$ (and $length_f$) denotes the sentence length of the original sentence which includes e_i. Under a given $(\breve{e},\breve{f})$ and $(\hat{e},\hat{f})$, our objective is to obtain word alignments. It is noted that an anchor word may include a phrase pair which forms n-to-m mapping objects.

Table 1 shows two example phrase pairs for French to English *c’est la vie* and *that is life*, and *la vie en rose* and *rosy life* with the initial value for the EM algorithm, the prior value and the fi-

[1] Although the code may be similar in practice to our Prior Model I, his explanation to modify the E-step will not be applied to IBM Models 3 and 4. Our view is to modify the M-step due to the same reason above, i.e. GIZA++ searches only over the alignment space around the Viterbi alignment.

<table>
<tr><th>Statistical MWE extraction method</th></tr>
<tr><td>97|||groupe_socialiste|||socialist_group|||26|||26
101|||monsieur_poettering|||mr_poettering|||1|||4
103|||monsieur_poettering|||mr_poettering|||1|||11
110|||monsieur_poettering|||mr_poettering|||1|||9
117|||explication_de_vote|||explanation_of_vote|||28|||26</td></tr>
<tr><th>Heuristic-based MWE extraction method</th></tr>
<tr><td>28|||the_wheel_2|||車輪_２ ||| 25||| 5
28|||the_primary-side_fixed_armature_13||| １_次_側_固定_電機_子_１_３ ||| 13||| 9
28|||the_secondary-side_rotary_magnet_7||| ２_次_側_回転_マグネット_７ ||| 15||| 11</td></tr>
</table>

Table 2: Example of MWE pairs in Europarl corpus (FR-EN) and NTCIR patent corpus (JP-EN). There are 5 columns for each term: sentence number, source term, target term, source position, and target position. The number appended to each term from the patent corpus (lower half) is a reference number. In this corpus, all the important technical terms have been identified and annotated with reference numbers.

nal lexical translation probability for Giza++ IBM Model 4 and that of our modified Giza++. Our modified Giza++ achieves the correct result when anchor words 'life' and '*vie*' are used to assign a value to the prior in our model.

3 Word Alignment

We review two models which address the problem of word alignment. The aim of word alignment is to obtain the model parameter t among English and French words, e_i and f_j respectively. We search for this model parameter under some model $\mathcal{M}$ where $\mathcal{M}$ is chosen by IBM Models 1-5 and the HMM model. We introduce the latent variable a, which is an alignment function with the hypothesis that each e and f correspond to this latent variable. (e, f, a) is a complete data set, and (e, f) is an incomplete data set.

3.1 EM Algorithm

We follow the description of the EM algorithm for IBM Models of Brown et al. (1993) but introduce the parameter t explicitly. In this model, the parameter t represents the lexical translation probabilities $t(e_i|f_j)$. It is noted that we use $e|f$ rather than $f|e$ following the notation of Koehn (2010). One important remark is that the Viterbi alignment of the sentence pair $(\breve{e}, \breve{f}) = (e_1^J, f_1^I)$, which is obtained as in (1):

$$\mathbf{E^{viterbi}}: \quad \hat{a_1^J} = \arg\max_{a_1^J} p_{\hat{\theta}}(f, a|e) \quad (1)$$

provides the best alignment for a given log-likelihood distribution $p_{\hat{\theta}}(f, a|e)$. Instead of summing, this step simplifies the E-step. However, under our modification of maximum likelihood estimate with MAP estimate, this simplification is not a correct approximation of the summation since our surface in the E-step is greatly perturbed by the prior. There is no guarantee that the Viterbi alignment is within the proximity of the target alignment (cf. Table 1).

Let z be the latent variable, t be the parameters, and x be the observations. The EM algorithm is an iterative procedure repeating the E-step and the M-step as in (2):

$$\begin{aligned}
\mathbf{E^{EXH}}: &\quad q(z;x) = p(z|x;\theta) \qquad (2)\\
\mathbf{M^{MLE}}: &\quad t' = \arg\max_t Q(t, t^{old})\\
&\qquad = \arg\max_t \sum_{x,z} q(z|x) \log p(x, z; t)
\end{aligned}$$

In the E-step, our knowledge of the values of the latent variables in a is given only by the posterior distribution $p(a|e, f, t)$. Hence, the (negative log)-likelihood of complete data (e, f, a), which we denote by $-\log p(t|e, f, a)$, is obtained over all possible alignments a. We use the current parameter values t^{old} to find the posterior distribution of the latent variables given by $p(a|e, f, t^{old})$. We then use this posterior distribution to find the expectation of the complete data log-likelihood evaluated for parameter value t. This expectation is given by $\sum_a p(a|e, f, t^{old}) \log p(e, f, a|t)$.

In the M-step, we use a maximal likelihood estimation to minimize negative log-likelihood in order to determine the parameter t; note that t is a lexical translation probability. Instead of using the log-likelihood $\log p(a, e, f|t)$, we use the expected complete data log-likelihood over all the possible alignments a that we obtained in the E-step, as in (3):

$$\mathrm{M}^{\mathbf{MLE}}: \quad t' = \arg\max_t Q(t, t^{old}) \qquad (3)$$
$$= \frac{c(f|e; f, e)}{\sum_e c(f|e; f, e)}$$

where an auxiliary function $c(e|f; e, f)$ for IBM Model 1 introduced by Brown et al. is defined as

$$c(f|e; f, e) = \sum_a p(a|e, f) \sum_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j})$$

and where the Kronecker-Delta function $\delta(x, y)$ is 1 if $x = y$ and 0 otherwise. This auxiliary function is convenient since the normalization factor of this count is also required. We note that if we use the MAP estimate, the E-step remains the same as in the maximum likelihood case, whereas in the M-step the quantity to be minimized is given by $Q(t, t^{old}) + \log p(t)$. Hence, we search for the value of t which maximizes the following equation:

$$\mathrm{M}^{\mathbf{MAP}}: \quad t' = \arg\max_t Q(t, t^{old}) + \log p(t)$$

3.2 HMM

A first-order Hidden Markov Model (Vogel et al., 1996) uses the sentence length probability $p(J|I)$, the mixture alignment probability $p(i|j, I)$, and the translation probability, as in (4):

$$p(f|e) = p(J|I) \prod_{j=1}^{J} p(f_j|e_i) \qquad (4)$$

Suppose we have a training set of R observation sequences X_r, where $r = 1, \cdots, R$, each of which is labelled according to its class m, where $m = 1, \cdots, M$, as in (5):

$$p(i|j, I) \quad = \quad \frac{r(i - j\frac{I}{J})}{\sum_{i'=1}^{I} r(i' - j\frac{I}{J})} \qquad (5)$$

The HMM alignment probabilities $p(i|i', I)$ depend only on the jump width $(i - i')$. Using a set of non-negative parameters $s(i - i')$, we have (6):

$$p(i|i', I) \quad = \quad \frac{s(i - i')}{\sum_{l=1}^{I} s(l - i')} \qquad (6)$$

4 Our Approach

Algorithm 1 Overall Algorithm

```
Given: a parallel corpus,
1. Extract MWEs by Algorithm 2.
2. Based on the results of Step 1, specify a set of anchor word alignment links in the format of anchor word alignment problem (cf. Definition 1 and Table 2).
3. Group MWEs in source and target text.
4. Calculate the prior in order to embed knowledge about anchor words.
5. Calculate lexical translation probabilities with the prior.
6. Obtain alignment probabilities.
7. Ungroup of MWEs in source and target text.
```

Algorithm 1 consists of seven steps. We use the Model I prior for the case where our prior knowledge is sparse and evenly distributed throughout the corpus, whereas we use the Model II prior when our prior knowledge is dense in a partial corpus. A typical example of the former case is when we use partial alignment annotation extracted throughout a corpus for bilingual terminology. A typical example of the latter case is when a sample of only a few hundred lines from the corpus have been hand-annotated.

4.1 MWE Extraction

Our algorithm of extracting MWEs is a statistical method which is a bidirectional version of Kupiec (1993). Firstly, Kupiec presents a method to extract bilingual MWE pairs in a unidirectional manner based on the knowledge about typical POS patterns of noun phrases, which is language-dependent but can be written down with some ease by a linguistic expert. For example in French they are N N, N prep N, and N Adj. Secondly, we take the intersection (or union) of extracted bilingual MWE pairs.[2]

[2] In word alignment, bidirectional word alignment by taking the intersection or union is a standard method which improves its quality compared to unidirectional word alignment.

Algorithm 2 MWE Extraction Algorithm

Given: a parallel corpus and a set of anchor word alignment links:

1. We use a POS tagger (Part-Of-Speech Tagger) to tag a sentence on the SL side.
2. Based on the typical POS patterns for the SL, extract noun phrases on the SL side.
3. Count n-gram statistics (typically $n = 1, \cdots, 5$ are used) on the TL side which jointly occur with each source noun phrase extracted in Step 2.
4. Obtain the maximum likelihood counts of joint phrases, i.e. noun phrases on the SL side and n-gram phrases on the TL side.
5. Repeat the same procedure from Step 1 to 4 reversing the SL and TL.
6. Intersect (or union) the results in both directions.

Let SL be the source language side and TL be the target language side. The procedure is shown in Algorithm 2. We informally evaluated the MWE extraction tool following Kupiec (1993) by manually inspecting the mapping of the 100 most frequent terms. For example, we found that 93 of the 100 most frequent English terms in the patent corpus were correctly mapped to their Japanese translation.

Depending on the corpus, we can use more prior knowledge about implicit alignment links. For example in some categories of patent and technical documents corpora,[3] we can use heuristics to extract the "noun phrase" + "reference number" from both sides. This is due to the fact that terminology is often labelled with a unique reference number, which is labelled on both the SL and TL sides.

4.2 Prior Model I

Algorithm 3 Prior Model I for IBM Model 1

```
Given: parallel corpus $\breve{e}$, $\breve{f}$,
    anchor words $biTerm$
initialize t($e|f$) uniformly
do until convergence
 set count($e|f$) to 0 for all e,f
 set total(f) to 0 for all f
 for all sentence pairs ($\breve{e}_s$,$\breve{f}_s$)
    prior($e|f$)$_s$ = getPriorModelI($\breve{e}, \breve{f}, biTerm$)
 for all words e in $\breve{e}_s$
    $total_s$(e) = 0
    for all words f in $\breve{f}_s$
       $total_s$(e) += t($e|f$)
 for all words e in $\breve{e}_s$
    for all words f in $\breve{f}_s$
       count($e|f$)+=$t(e|f)/total_s(e)\times prior(e|f)_s$
       total(f) += $t(e|f)/total_s(e) \times prior(e|f)_s$
 for all f
    for all e
       t($e|f$) = $count(e|f)/total(f)$
```

Prior for Exhaustive Alignment Space IBM Models 1 and 2 implement a prior for all possible alignments exhaustively. Such a prior requires the following two conditions. Firstly, partial knowledge about the prior that we use in our context is defined as follows. Let us denote a bilingual term list $T = \{(s_1, t_1), \ldots, (s_m, t_m)\}$. For example with IBM Model 1: Let us define the following prior $p(e|f, e, f; T)$ from Equation (4):

$$p(e|f,e,f;T) \quad = \quad \begin{cases} 1 & (e_i = s_i, f_j = t_j) \\ 0 & (e_i = s_i, f_j \neq t_j) \\ 0 & (e_i \neq s_i, f_j = t_j) \\ \text{uniform} & (e_i \neq s_i, f_j \neq t_j) \end{cases}$$

Secondly, this prior should be proper for the exhaustive case and non-proper for the sampled alignment space where by proper we mean that the probability is normalized to 1. Algorithm 3 shows the pseudo-code for Prior Model I. Note that if the prior is uniform in the MAP estimation, this is equivalent to maximum likelihood estimation.

Prior for Sampled Alignment (Function) Space Due to the exponential costs introduced by fertility, null token insertion, and distortion probability, IBM Models 3 and 4 do not consider all $(I + 1)^J$ alignments exhaustively, but rather a small subset in the E-step. Each iteration only uses the subset of all the alignment functions: this sampling

[3] Unlike other language pairs, the availability of Japanese–English parallel corpora is quite limited: the NTCIR patent corpus (Fujii et al., 2010) of 3 million sentence pairs (the latest NTCIR-8 version) for the patent domain and JENAAD corpus (Utiyama and Isahara, 2003) of 150k sentence pairs for the news domain. In this regard, the patent domain is particularly important for this particular language pair.

is not uniform, as it only includes the best possible alignment with all its neighbouring alignments which differ from the best alignment by one word (this can be corrected by a move operation) or two words (this can be corrected by a swap operation).

If we consider the neighbouring alignment via a move or a swap operation, two issues arise. Firstly, the fact that these two neighbouring alignments are drawn from different underlying distributions needs to be taken into account, and secondly, that the application of a move and a swap operation alters a row or column of a prior matrix (or indices of the prior) since either operation involves the manipulation of links.

Algorithm 4 Pseudo-code for Prior Model II Exhaustive Alignment Space

```
def getPriorModelII($\breve{e}$,$\breve{f}$,biTerm):
for i in sentence:
    for e in $\breve{e}_i$:
        allWords$_i$ = length of sentence $\breve{e}$
        for f in $\breve{f}_i$:
            if $(e, f)$ in biTerm:
                n= num of anchor words in $i$
                uni$(e|f)_i$ = $\frac{\text{allWords}_i - \text{n}}{\text{allWords}_i}$
                expSum$(e|f)$ += uni$(e|f)_i \times$ n
            else:
                countSum$(e|f)_i$ += n
countSum$(e|f)$ += count$(e|f)_i$
for e in $all_e$:
    for f in $all_f$:
        $prior(e|f)$ = expSum$(e|f)$ + countSum$(e|f)$
return $prior(e|f)$
```

Prior for Jump Width i' One implementation of HMM is to use the forward-backward algorithm. A prior should be embedded within the forward-backward algorithm. From Equation (6), there are three cases which depend on whether a_i and its neighbouring alignment a_{i-1} are determined by our prior knowledge about anchor words or not. When both a_i and a_j are determined, this probability is expressed as in (7):

$$p(i-i';I) = \begin{cases} 0 & (else) \\ 1 & (e_i = s_i, f_j = t_j \text{ for } a_i) \text{ and } \\ & (e'_i = s'_i, f'_j = t'_j \text{ for } a_j) \end{cases} \tag{7}$$

When either a_i or a_j is determined, this probability is expressed as in (8):[4]

$$p(i-i';I) = \begin{cases} 0 & (\text{condition } 1) \\ 1 & (\text{condition } 2) \\ \frac{1}{(m - \#e_{a_i} - \cdots - \#e_{a_i+m})} & (else) \\ (\text{uniform distribution}) & \end{cases} \tag{8}$$

When neither a_i nor a_j is determined, this probability is expressed as in (9): [5]

$$p(i-i';I) = \begin{cases} 0 & (\text{condition } 3) \\ 1 & (\text{condition } 4) \\ \frac{m-i'}{(m - \#e_{a_i} - \cdots - \#e_{a_i+m})^2} & (else) \\ (\text{Pascal's triangle distribution}) & \end{cases} \tag{9}$$

4.3 Prior Model II

Prior Model II assumes that we have prior knowledge only in some part of the training corpus. A typical example is when a small part of the corpus has a hand-crafted 'gold standard' annotation.

Prior for Exhaustive Alignment Space Prior Model II is used to obtain the prior probability $p(e|f)$ over all possible combinations of e and f. In contrast to Prior Model I, which computes the prior probability $p(e|f)$ for each sentence, Prior Model II computes the prior probability globally for all sentences in the corpus. Algorithm 4 shows the pseudo-code for Prior Model II Exhaustive Alignment Space.

[4] condition 1 is as follows:

$((e_i \neq s_i, f_j \neq t_j \text{ for } a_i)$ and $(e'_i = s'_i, f'_j = t'_j \text{ for } a_j))$ or
$((e_i \neq s_i, f_j \neq t_j \text{ for } a_i)$ and $(e'_i = s'_i, f'_j = t'_j \text{ for } a_j))$ or
$((e_i = s_i, f_j = t_j \text{ for } a_i)$ and $(e'_i \neq s'_i, f'_j \neq t'_j \text{ for } a_j))$ or
$((e_i = s_i, f_j = t_j \text{ for } a_i)$ and $(e'_i \neq s'_i, f'_j \neq t'_j \text{ for } a_j))$

'condition 2' is as follows:

$((e_i = s_i, f_j \neq t_j \text{ for } a_i)$ and $(e'_i = s'_i, f'_j = t'_j \text{ for } a_j))$ or
$((e_i \neq s_i, f_j = t_j \text{ for } a_i)$ and $(e'_i = s'_i, f'_j = t'_j \text{ for } a_j))$ or
$((e_i = s_i, f_j = t_j \text{ for } a_i)$ and $(e'_i \neq s'_i, f'_j = t'_j \text{ for } a_j))$ or
$((e_i = s_i, f_j = t_j \text{ for } a_i)$ and $(e'_i = s'_i, f'_j \neq t'_j \text{ for } a_j))$

[5] 'condition 3' is as follows:
$((e_i \neq s_i, f_j \neq t_j \text{ for } a_i)$ and $(e'_i \neq s'_i, f'_j \neq t'_j \text{ for } a_j))$

'condition 4' is as follows:
$((e_i \neq s_i, f_j \neq t_j \text{ for } a_i)$ and $(e'_i \neq s'_i, f'_j = t'_j \text{ for } a_j))$ or
$((e_i \neq s_i, f_j \neq t_j \text{ for } a_i)$ and $(e'_i = s'_i, f'_j \neq t'_j \text{ for } a_j))$ or
$((e_i \neq s_i, f_j = t_j \text{ for } a_i)$ and $(e'_i \neq s'_i, f'_j \neq t'_j \text{ for } a_j))$ or
$((e_i = s_i, f_j \neq t_j \text{ for } a_i)$ and $(e'_i \neq s'_i, f'_j \neq t'_j \text{ for } a_j))$

Prior for Sampled Alignment (Function) Space This is identical to that of the Prior Model II exhaustive alignment space with only a difference in the normalization process.

Prior for Jump Width i' This categorization of Prior Model II is the same as that of Prior Model I for for Jump Width i' (see Section 4.2). Note that Prior Model II requires more memory compared to the Prior Model I.[6]

5 Experimental Settings

The baseline in our experiments is a standard log-linear phrase-based MT system based on Moses. The GIZA++ implementation (Och and Ney, 2003a) of IBM Model 4 is used as the baseline for word alignment, which we compare to our modified GIZA++. Model 4 is incrementally trained by performing 5 iterations of Model 1, 5 iterations of HMM, 5 iterations of Model 3, and 5 iterations of Model 4. For phrase extraction the grow-diag-final heuristics are used to derive the refined alignment from bidirectional alignments. We then perform MERT while a 5-gram language model is trained with SRILM. Our implementation is based on a modified version of GIZA++ (Och and Ney, 2003a). This modification is on the function that reads a bilingual terminology file, the function that calculates priors, the M-step in IBM Models 1-5, and the forward-backward algorithm in the HMM Model. Other related software tools are written in Python and Perl: terminology concatenation, terminology numbering, and so forth.

6 Experimental Results

We conduct an experimental evaluation on the NTCIR-8 corpus (Fujii et al., 2010) and on Europarl (Koehn, 2005). Firstly, MWEs are extracted from both corpora, as shown in Table 3. In the second step, we apply our modified version of GIZA++ in which we incorporate the results of MWE extraction. Secondly, in order to incorporate the extracted MWEs, they are reformatted as shown in Table 2. Thirdly, we convert all MWEs into a single token, i.e. we concatenate them with an underscore character. We then run the modified version of GIZA++ and obtain a phrase and reordering table. In the fourth step, we split the concatenated MWEs embedded in the third step. Finally, in the fifth step, we run MERT, and proceed with decoding before automatically evaluating the translations.

corpus	language	size	#unique MWEs	#all MWEs
statistical method				
NTCIR	EN-JP	200k	1,121	120,070
europarl	EN-FR	200k	312	22,001
europarl	EN-ES	200k	406	16,350
heuristic method				
NTCIR	EN-JP	200k	50,613	114,373

Table 3: Statistics of our MWE extraction method. The numbers of MWEs are from 0.08 to 0.6 MWE / sentence pair in our statistical MWE extraction methods.

Table 4 shows the results where 'baseline' indicates no BMWE grouping nor prior, and 'baseline2' represents a BMWE grouping but without the prior. Although 'baseline2' (BMWE grouping) shows a drop in performance in the JP–EN / EN–JP 50k sentence pair setting, Prior Model I results in an increase in performance in the same setting. Except for EN–ES 200k, our Prior Model I was better than 'baseline2'. For EN–JP NTCIR using 200k sentence pairs, we obtained an absolute improvement of 0.77 Bleu points compared to the 'baseline'; for EN–JP using 50k sentence pairs, 0.75 Bleu points; and for ES–EN Europarl corpus using 200k sentence pairs, 0.63 Bleu points. In contrast, Prior Model II did not work well. The possible reason for this is the misspecification, i.e. the modelling by IBM Model 4 was wrong in terms of the given data. One piece of evidence for this is that most of the enforced alignments were found correct in a manual inspection.

For EN–JP NTCIR using the same corpus of 200k, although the number of unique MWEs ex-

[6] This is because it needs to maintain potentially an $\ell \times m$ matrix, where ℓ denotes the number of English tokens in the corpus and m denotes the number of foreign tokens, even if the matrix is sparse. Prior Model I only requires an $\hat{\ell} \times \hat{m}$ matrix where $\hat{\ell}$ is the number of English tokens in a sentence and $\hat{m}$ is the number of foreign tokens in a sentence, which is only needed until this information is incorporated in a posterior probability during the iterative process.

size	EN-JP	Bleu	JP-EN	Bleu
50k	baseline	16.33	baseline	22.01
50k	baseline2	16.10	baseline2	21.71
50k	prior I	17.08	prior I	22.11
50k	prior II	16.02	prior II	20.02
200k	baseline	23.42	baseline	21.68
200k	baseline2	24.10	baseline2	22.32
200k	prior I	24.22	prior I	22.45
200k	prior II	23.22	prior II	21.00
size	FR-EN	Bleu	EN-FR	Bleu
50k	baseline	17.68	baseline	17.80
50k	baseline2	17.76	baseline2	18.00
50k	prior I	17.81	prior I	18.02
50k	prior II	17.01	prior II	17.30
200k	baseline	18.40	baseline	18.20
200k	baseline2	18.80	baseline2	18.50
200k	prior I	18.99	prior I	18.60
200k	prior II	18.20	prior II	17.50
size	ES-EN	Bleu	EN-ES	Bleu
50k	baseline	16.21	baseline	15.17
50k	baseline2	16.61	baseline2	15.60
50k	prior I	16.91	prior I	15.87
50k	prior II	16.15	prior II	14.60
200k	baseline	16.87	baseline	17.62
200k	baseline2	17.40	baseline2	18.21
200k	prior I	17.50	prior I	18.20
200k	prior II	16.50	prior II	17.10

Table 4: Results. Baseline is plain GIZA++ / Moses (without BMWE grouping / prior), baseline2 is with BMWE grouping, prior I / II are with BMWE grouping and prior.

tracted by the statistical method and the heuristic method varies significantly, the total number of MWEs by each method becomes comparable. The resulting Bleu score for the heuristic method (24.24 / 22.48 Blue points for 200k EN–JP / JP–EN) is slightly better than that of the statistical method. The possible reason for this is related to the way the heuristic method groups terms including reference numbers, while the statistical method does not. As a result, the complexity of the alignment model simplifies slightly in the case of the heuristic method.

7 Conclusion

This paper presents a new method of incorporating BMWEs into word alignment. We first detect BMWEs in a bidirectional way and then use this information to do groupings and to enforce already known alignment links. For the latter process, we replace the maximum likelihood estimate in the M-step of the EM algorithm with the MAP estimate; this replacement allows the incorporation of the prior in the M-step of the EM algorithm. We include an experimental investigation into incorporating extracted BMWEs into a word aligner. Although there is some work which incorporates BMWEs in groupings, they do not enforce alignment links.

There are several ways in which this work can be extended. Firstly, although we assume that our a priori partial annotation is reliable, if we extract such MWEs automatically, we cannot avoid erroneous pairs. Secondly, we assume that the reason why our Prior Model II did not work was due to the misspecification (or wrong modelling). We would like to check this by discriminative modelling. Thirdly, although here we extract BMWEs, we can extend this to extract paraphrases and non-literal expressions.

8 Acknowledgments

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (`http://www.cngl.ie`) at Dublin City University and Trinity College Dublin. We would also like to thank the Irish Centre for High-End Computing.

References

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Springer. Cambridge, UK

Brown, Peter F., Vincent .J.D Pietra, Stephen A.D.Pietra, Robert L. Mercer. 1993. *The Mathematics of Statistical Machine Translation: Parameter Estimation*. Computational Linguistics. 19(2), pp. 263–311.

Callison-Burch, Chris, David Talbot and Miles Osborne. 2004. *Statistical Machine Translation with*

Word- and Sentence-Aligned Parallel Corpora. Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), Main Volume. Barcelona, Spain, pp. 175–182.

Fujii, Atsushi, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro, Terumasa Ehara, Hiroshi Echizenya, Sayori Shimohata. 2010. *Overview of the Patent Translation Task at the NTCIR-8 Workshop*. Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access, pp. 293–302.

Graca, Joao de Almeida Varelas, Kuzman Ganchev, Ben Taskar. 2007. *Expectation Maximization and Posterior Constraints*. In Neural Information Processing Systems Conference (NIPS), Vancouver, BC, Canada, pp. 569–576.

Gale, William, and Ken Church. 1991. *A Program for Aligning Sentences in Bilingual Corpora*. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics. Berkeley CA, pp. 177–184.

Koehn, Philipp, Franz Och, Daniel Marcu. 2003. *Statistical Phrase-Based Translation*. In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics. Edmonton, Canada. pp. 115–124.

Koehn, Philipp. 2005. *Europarl: A Parallel Corpus for Statistical Machine Translation*. In Conference Proceedings: the tenth Machine Translation Summit. Phuket, Thailand, pp.79-86.

Koehn, Philipp, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, 2007. *Moses: Open source toolkit for Statistical Machine Translation*. Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, Prague, Czech Republic, pp. 177–180.

Koehn, Philipp. 2010. *Statistical Machine Translation*. Cambridge University Press. Cambridge, UK.

Kupiec, Julian. 1993. *An Algorithm for finding Noun Phrase Correspondences in Bilingual Corpora.* In Proceedings of the 31st Annual Meeting of Association for Computational Linguistics. Columbus. OH. pp. 17–22.

Lambert, Patrik and Rafael Banchs. 2006. *Grouping Multi-word Expressions According to Part-Of-Speech in Statistical Machine Translation*. In Proceedings of the EACL Workshop on Multi-Word-Expressions in a Multilingual Context. Trento, Italy, pp. 9–16.

McLachlan, Geoffrey J. and Thriyambakam Krishnan, 1997. *The EM Algorithm and Extensions*. Wiley Series in probability and statistics. New York, NY.

Moore, Robert C.. 2003. *Learning Translations of Named-Entity Phrases from Parallel Corpora*. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics. Budapest, Hungary. pp. 259–266.

Moore, Robert C.. 2004. *On Log-Likelihood-Ratios and the Significance of Rare Events*. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP). Barcelona, Spain, pp. 333–340.

Och, Franz and Herman Ney. 2003. *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics. 29(1), pp. 19–51.

Resnik, Philip and I. Dan Melamed, 1997. *Semi-Automatic Acquisition of Domain-Specific Translation Lexicons*. Proceedings of the 5th Applied Natural Language Processing Conference. Washington, DC., pp. 340–347.

Talbot, David. 2005. *Constrained EM for parallel text alignment*, Natural Language Engineering, 11(3): pp. 263–277.

Utiyama, Masao and Hitoshi Isahara. 2003. *Reliable Measures for Aligning Japanese-English News Articles and Sentences*, In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. Sapporo, Japan, pp. 72–79.

Vogel, Stephan, Hermann Ney, Christoph Tillmann 1996. *HMM-Based Word Alignment in Statistical Translation*. In Proceedings of the 16th International Conference on Computational Linguistics. Copenhagen, Denmark, pp. 836–841.