Learning Phrase Translation using Level of Detail Approach

Hendra Setiawan[12], Haizhou Li[1], Min Zhang[1]
[1]Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
{stuhs,hli,mzhang}@i2r.a-star.edu.sg
[2]School of Computing
National University of Singapore
Singapore 117543
hendrase@comp.nus.edu.sg

Abstract

We propose a simplified Level Of Detail (LOD) algorithm to learn phrase translation for statistical machine translation. In particular, LOD learns unknown phrase translations from parallel texts without linguistic knowledge. LOD uses an agglomerative method to attack the combinatorial explosion that results when generating candidate phrase translations. Although LOD was previously proposed by (Setiawan et al., 2005), we improve the original algorithm in two ways: simplifying the algorithm and using a simpler translation model. Experimental results show that our algorithm provides comparable performance while demonstrating a significant reduction in computation time.

1 Introduction

Many natural language processing applications, such as machine translation, treat words as the primitive unit of processing. These units are often treated as a set, discarding ordering information, and reducing an utterance as a bag of words. However, natural language often exhibits a non-compositional property where an utterance's meaning is a matter of convention rather than the sum of its parts (e.g., "lend an ear").

While it is desirable to extract linguistically-motivated phrases, it is often difficult to do so. The case of statistical machine translation (SMT) is an illustrative application, as researchers in this area often do not make assumptions about the source and target languages. As such, the notion of a phrase in SMT usually connotes a statistically significant grouping of words rather than a grouping that is linguistically significant. In SMT, phrases are learned without distinguishing non-compositional from compositional ones. Despite this problem, SMT has witnessed great benefits from learning such phrasal units.

One method to address this problem is to employ evidence found in parallel corpora. This is not a new idea, as there is a vast amount of literature that directly addresses learning phrase translation from parallel texts. The identification of meaningful phrase translation includes the learning of the translation of non-compositional compounds (Melamed, 1997), "captoids" and name entities (Moore, 2003) and both gapped and rigid multiword sequence (Kaoru et al., 2003) just to name a few. Specifically for phrase-based SMT, there are many approaches proposed for learning phrase translation, such as the joint model (Marcu and Wong, 2002), ISA (Zhang et al., 2003), alignment templates (Och et al., 1999), HMM paths (Vogel et al., 1996) and projection extensions (Tillmann, 2003).

In our previous work (Setiawan et al., 2005), we introduced an agglomerative approach to learn phrase translation for SMT. While the LOD approach works well, a weakness is that it is quite slow, as it suffers from computational inefficiency when calculating translation candidates. In this paper, we modify the original LOD approach by simplifying the learning process and using a simpler translation model while maintaining a comparable level of performance.

In the remainder of this paper, we will describe our simplified Level of Detail (LOD) algorithm. Section 2 discusses related work, including the original LOD algorithm. Section 3 describes the new alternations in the LOD algorithm, with particular attention to simplified learning process. In section 4, we report our experiments on the two LOD algorithms and the use of simpler translation model. Finally, section 5 concludes this paper by providing some discussion.

2 Statistical Machine Translation: A Level of Detail

2.1 Motivation

The problem of learning phrase translation is more complicated than it appears due to the fact that the actual phrase is unknown. This information gap forces the learning algorithm to explore all possibilities in the parallel corpus to search for the interesting candidates.

Most algorithms for learning phrase translations from parallel corpora can be generalized into a pipeline consisting of three steps in series. We classify our discussion of previous work along these main steps:

- **Generation.** This step analyzes the parallel texts and forms a pool of all candidate phrase translations. This step emphasizes recall and attempts to identify all possible candidates, with a minimal amount of filtering, resulting in a noisy pool of translation pairs. Candidate identification algorithms often work directly with the parallel texts or utilize an underlying word alignment.

  Algorithms for this step often consider all possible segmentation as possible phrases and all possible pairings as possible translation candidates. This leads to a combinatorially large number of candidates. The size of the candidate pool is even larger when non-contiguous phrases are considered. To alleviate the problem, many limiting assumptions are imposed. Usually, the algorithm imposes a contiguity constraint, as the number of non-contiguous phrases is relatively small in most languages. A maximum phrase length can also be introduced to limit the generation step. This limit retains most of the algorithm's performance since the count of long phrases decreases gradually.

- **Scoring.** This step calculates a significance score that reflects the interestingness of the candidate for each entry in the candidate pool. There have been numerous metrics used to score candidates; specific examples include mixtures of alignment map, word-based lexicon and language-specific measures (Venugopal et al., 2004), block frequency (Tillmann, 2003), relative frequency (Och et al., 1999), lexical weighting (Koehn et al., 2003) and a set of features reflecting word penalty, length penalty and lexicon score (Zens and Ney, 2004).

  The scoring function must accommodate phrases of varying length and allow direct comparison between them. Many methods employ a normalization process over the phrase length to enable the comparison, but such normalization may not reflect the actual distribution of the phrase.

- **Selection.** This final step selects the most probable candidates as phrase translations. The algorithm typically explores all candidates and decides their promotion based on their scores. Rank-based methods, using maximal separation criteria (Venugopal et al., 2004) and frequency filtering (Tillmann, 2003) are common methods employed for this step.

  This step is simple as long as the associated score reflects the true interestingness of the candidates. In such cases, a correctly set score threshold separates the desired phrase translations from noise. In practice, the selection step errs on the side of recall, acquiring as many candidates as possible, even with the cost of having a large number of un-interesting translations. Errors in the phrase translation acquisition are mitigated by the decoding step. The hope is that the decoding process will utilize only the most interesting phrase translations in translating new, unseen sentences.

The majority of approaches try to learn phrase translation directly from the texts in one-step. As such, this unconstrained method leads to a combinatorial explosion in the number of translation candidates. The LOD approach addresses this by stating that a phrase translation may consist of several levels according to its alignment granularity. Traditionally, a phrase translation can be described as a set of word alignments or as a single phrase alignment. The LOD model introduces several levels in between these two extremes, describing certain phrase translations as a cascaded series of sub-phrase alignments, where sub-phrase are multi-word units that are shorter than entire phrase. Phrase translation at its finest level of detail uses simple word alignment, whereas at the coarsest level of detail uses entire phrase alignment. A coarser alignment is a merge between more than one finer alignment. The coarser the level of the phrase translation, the longer the

unit involves. Therefore, our agglomerative approach tries to learn phrase translation starting from the word level alignment through intermediate, sub-phrase alignment levels.

Under agglomerative framework, the LOD approach addresses the issues that arise from the unknown phrase problem. In the generation step, LOD only considers generating candidates that are one level more coarse, which significantly limits the amount of candidates generated per iteration. In the scoring step, LOD estimates the probability of the candidates using soft-counting as described in (Setiawan et al., 2005). In the selection step, LOD applies a decoding-like algorithm to select the best set of phrase translation describing the sentence pair rather than ranking all the candidates.

2.2 Formulation

The LOD algorithm is language-independent, but for concreteness we introduce it using English as the source language and French as the target language. Let $\langle \mathbf{e}, \mathbf{f} \rangle$ be a sentence pair of English sentence $\mathbf{e}$ with its translation in French $\mathbf{f}$. Let $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle$ represent the same sentence pair but using phrases as atomic units rather than words. Words are thus phrases of length one. Therefore, we arrive at the following notation $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(0)}$ for $\langle \mathbf{e}, \mathbf{f} \rangle$ and $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(N)}$ for $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle$. The superscript in the notation denotes level of detail with 0 for the finest and N for the coarsest. Let $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(n)}$ be a sentence pair at an intermediate level between 0 and N and $\tilde{\mathbf{e}}^{(n)} = \{\tilde{e}_0^{(n)}, \tilde{e}_1^{(n)}, ...\tilde{e}_i^{(n)}, ...\tilde{e}_{l'}^{(n)}\}$ and $\tilde{\mathbf{f}}^{(n)} = \{\tilde{f}_0^{(n)}, \tilde{f}_1^{(n)}, ...\tilde{f}_j^{(n)}, ...\tilde{f}_{m'}^{(n)}\}$ be its tuples with $\tilde{e}_0^{(n)}$ and $\tilde{f}_0^{(n)}$ represent the special token *NULL* as suggested in (Brown et al., 1993) and $l^{(n)}, m^{(n)}$ represent the length of the corresponding sentence. Let $T^{(n)}$ be a set of alignments defined over the sentence pair with $\tilde{t}_{ij}^{(n)}=[\tilde{e}_i^{(n)}, \tilde{f}_j^{(n)}]$ as its member.

As in our previous work, LOD algorithm transforms $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(0)}$ to $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(N)}$ iteratively. In every iteration, LOD performs a series of steps, similar to the pipeline followed by many other algorithms, to learn phrase translation at one level more coarse. In the generation step, LOD algorithm forms $\mathcal{B}^{(n)}$, a pool of sub-phrase alignments, as the basis for the generation of phrase alignment candidate. LOD generates all possible candidates from $\mathcal{B}^{(n)}$ and forms a pool of phrase alignment candidates, $\mathcal{C}^{(n)}$. $\mathcal{B}^{(n)}$ and $\mathcal{C}^{(n)}$ together form a pool of phrase translation at one level more coarse. In the scoring step, LOD estimates the probability for all entries in $\mathcal{B}^{(n)}$ and $\mathcal{C}^{(n)}$. In the selection step, LOD selects a set of most probable phrase translations from $\mathcal{B}^{(n)}$ and $\mathcal{C}^{(n)}$ to cover each sentence pair. In another words, LOD algorithm re-aligns the sentence pair using the alignments from $\mathcal{B}^{(n)}$ and $\mathcal{C}^{(n)}$.

Algorithm 1. An algorithmic sketch of the simplified LOD approach. It takes a sentence pair at its word level (the finest level of detail) as its input, and learns the phrase-level alignment iteratively and outputs the same sentence pair in its coarsest level of detail, along with the generated phrase translation table.

input $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(0)}$
for $n = 0$ **to** $N - 1$ **do**
  if n=0 **then**
    - Generate $\mathcal{B}^{(0)}$ from $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(0)}$.
  else
    - Generate $\mathcal{B}^{(n)}$ from $T^{(n)}$.
  - Identify $\mathcal{C}^{(n)}$ from $\mathcal{B}^{(n)}$.
  - Estimate the probabilities of $\mathcal{B}^{(n)}$ and $\mathcal{C}^{(n)}$.
  - Learn $\tilde{T}^{(n+1)}$ from $\mathcal{B}^{(n)}$ and $\mathcal{C}^{(n)}$ and construct $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(n+1)}$ according to $\tilde{T}^{(n+1)}$.
output $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(N)}$ and $\tilde{T}^{(N)}$

In the generation step, LOD algorithm uses the bi-directional word alignments from the sentence pair $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(n)}$ to form $\mathcal{B}^{(n)}$, resulting a considerably high collection of sub-phrase alignments with high coverage over the sentence pair. In practice, the production of the bi-directional word alignments requires the estimation of IBM translation model over the whole corpus which is well known to be expensive. In the original formulation, LOD algorithm is required to perform the estimation for every iterations, taking up a major portion in computation time. In this paper, we simplify the algorithm by modifying the sub-phrase level alignments generation step and keeping the other steps the same. The simplification is reflected in Algorithm 1 where LOD algorithm distinguishes the first iteration from the rest with respect to the generation of $\mathcal{B}^{(n)}$. The algorithm suggests that instead of relying on the high recall alignments, the simplified algorithm relies on the high precision alignment learnt from the previous iteration to form the pool of sub-phrase level alignments.

3 Learning Phrase Translation

Each iteration of the LOD process generates a new pool of (sub-)phrase level alignments from

the previous, finer-grained ones. At each iteration LOD follows four simple steps:

1. Generation of sub-phrase level alignments
2. Identification of phrase level alignment candidates
3. Estimation of alignment probability
4. Learning of coarser level alignment

3.1 Generation of sub-phrase level alignments

We distinguish the generation of sub-phrase level alignment between the first iteration and the rest. In the first iteration, LOD collects the asymmetrical word alignment from both translation directions as the basis from which phrase alignment candidates are generated. For the sake of clarity, we define the following notation for these alignments:

Let $\Gamma^{(0)}_{ef} : \tilde{e}^{(0)}_i \rightarrow \tilde{f}^{(0)}_j$ be an alignment function that represents all alignments from translating English sentence to French, and $\Gamma^{(0)}_{fe} : \tilde{f}^{(0)}_j \rightarrow \tilde{e}^{(0)}_i$ be the same but for reverse translation direction. Then, sub-phrase alignment $\mathcal{B}^{(0)}$ for the first iteration includes all possible alignments defined by both functions:

$$\begin{aligned}\mathcal{B}^{(0)} =&\{[\tilde{e}^{(0)}_i, \tilde{f}^{(0)}_j]|\Gamma^{(0)}_{ef}(\tilde{e}^{(0)}_i) = \tilde{f}^{(0)}_j\}\cup \\ &\{[\tilde{e}^{(0)}_i, \tilde{f}^{(0)}_j]|\Gamma^{(0)}_{fe}(\tilde{f}^{(0)}_j) = \tilde{e}^{(0)}_i\}\end{aligned}$$

For the rest of the iterations, the algorithm takes the phrase alignment learnt from previous iteration $T^{(n)}$ to form $\mathcal{B}^{(n)}$. In this way, LOD algorithm gains a significant reduction in computation time by avoiding the generation of bi-directional word alignment. Therefore, at any iteration n except the initial one, LOD takes all phrase translation $T^{(n)}$ learnt from the previous iteration to form $\mathcal{B}^{(n)}$. Then, $\mathcal{B}^{(n)}$ is defined as:

$$\mathcal{B}^{(n)} = \{[\tilde{e}^{(n)}_i, \tilde{f}^{(n)}_j]|[\tilde{e}^{(n)}_i, \tilde{f}^{(n)}_j] \in T^{(n)}\}$$

3.2 Identification of Phrase Alignment Candidates

LOD applies a simple heuristic to identify possible phrase alignments. First, LOD considers every combination of two distinct sub-phrase alignments, where a phrase alignment candidate $\langle t^{(n)}_{ij}, t^{(n)}_{i'j'}\rangle \in \mathcal{C}$ is defined as follows:

Let $\langle t^{(n)}_{ij}, t^{(n)}_{i'j'}\rangle$ be a set of two tuples, where $t^{(n)}_{ij} \in \mathcal{B}^{(n)}$ and $t^{(n)}_{i'j'} \in \mathcal{B}^{(n)}$. Then $\langle t^{(n)}_{ij}, t^{(n)}_{i'j'}\rangle$ is a phrase alignment candidate **iff**:

- $\neg\left((i, i') \neq 0\right)$ **or** $|i - i'| = 1$
- $\neg\left((j, j') \neq 0\right)$ **or** $|j - j'| = 1$
- $\neg((t^{(n)}_{ij} \in N^{(n)})$ **and** $(t^{(n)}_{i'j'} \in N^{(n)}))$

The first and second clauses define a candidate as a set of two sub-phrase alignments that are adjacent to each other while the third clause forbids candidates of two $NULL$ alignments.

The LOD heuristic generates candidates from the combination of only two alignments at a time and hands over the candidate generation of more coarse alignment to the subsequent iteration. By considering only two alignments, the LOD model opens the opportunity for non-contiguous phrase translation without the disadvantage of combinatorial explosion in the number of candidates.

3.3 Estimation of Phrase Alignment Candidate Probability

Joining the alignment set $\mathcal{B}^{(n)}$ derived in Section 3.1 and the coarser level alignment $\mathcal{C}^{(n)}$ derived in Section 3.2, we form a candidate alignment set $\mathcal{B}^{(n)} \cup \mathcal{C}^{(n)}$. LOD utilizes information theoretic measure as suggested by (Church and Hanks, 1989) to assess the candidacy of each entry in $\mathcal{C}^{(n)}$. Assuming that there are two alignments $x \in \mathcal{B}^{(n)}$,$y \in \mathcal{B}^{(n)}$ and a candidate alignment $\langle x, y\rangle \in \mathcal{C}^{(n)}$, we derive the probability $p(x)$ and $p(y)$ from the statistics as the count of x and y normalized by the number of sentence pairs in the corpus, and derive probability $p(\langle x, y\rangle)$ in a similar way.

If there is a genuine association between x and y, then we expect the joint probability $p(\langle x, y\rangle) \gg p(x)p(y)$. If there is no interesting relationship between x and y, then $p(\langle x, y\rangle) \approx p(x)p(y)$ where we say that x and y are independent. If x and y are in a complementary relationship, then we expect to see that $p(\langle x, y\rangle) \ll p(x)p(y)$.

3.4 Learning Coarser Level Alignment

From section 3.1 to 3.3, we have prepared all the necessary alignments with their probability estimates.

The final step is to re-align $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}}\rangle^{(n)}$ into $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}}\rangle^{(n+1)}$ using alignments in $\mathcal{B}^{(n)} \cup \mathcal{C}^{(n)}$. We consider re-alignment as constrained search

problem. Let $p(\tilde{t}_{ij}^{(n)})$ be the probability of a phrase alignment $\tilde{t}_{ij}^{(n)} \in (\mathcal{B}^{(n)} \cup \mathcal{C}^{(n)})$ as defined in Section 3.3 and $T^{(n)}$ be the potential new alignment sequence for $\langle \tilde{\mathbf{e}}, \tilde{\mathbf{f}} \rangle^{(n)}$, then we have the likelihood for $T^{(n)}$ as:

$$\log P(< \tilde{\mathbf{e}}, \tilde{\mathbf{f}} >^{(n)} | T^{(n)}) = \sum_{t_{ij}^{(n)} \in T^{(n)}} \log p(t_{ij}^{(n)}) \quad (1)$$

The constrained search decodes an alignment sequence that produces the highest likelihood estimate in the current iteration, subject to the following constraints:

- to preserve the phrase ordering of the source and target languages
- to preserve the completeness of word or phrase coverage in the sentence pair
- to ensure the mutual exclusion between alignments (except special $NULL$ tokens)

Then, the constrained search process can be formulated as:

$$T^{(n+1)} = \underset{\forall T^{(n)}}{\text{argmax}} \; P(\langle \tilde{e}, \tilde{f} \rangle^{(n)} | T^{(n)}) \quad (2)$$

In Equation 2, we have $T^{(n+1)}$ as the best alignment sequence to re-align sentence pair $\langle \tilde{e}, \tilde{f} \rangle^{(n)}$ to $\langle \tilde{e}, \tilde{f} \rangle^{(n+1)}$. Details on how to obtain this alignment sequence are identical to the original LOD algorithm, as discussed in (Setiawan et al., 2005). From this alignment, we form a new pool of sub-phrase alignments $\mathcal{B}^{(n+1)}$ for next iteration.

4 Experiments

Since there is no gold standard to evaluate the quality of phrase translation, we can not directly validate the quality of phrase translation resulting from applying the LOD algorithm. However, we can evaluate the performance of LOD approach from an SMT perspective. Our previous work demonstrated that LOD improves the performance of word-based SMT significantly. The current evaluation has two objectives:

1. To assess the performance impact of our simplified algorithm versus the original LOD approach. We would like to see the impact of using phrase translation for generating coarser candidates.
2. To analyze the effect of using a simpler translation model on translation performance. Simpler translation models are easier to estimate, but have reduced expressiveness in modeling. We would to assess the trade-off between using our faster but lower quality alignment on the end translation. The objective is thus to discover how susceptible the LOD algorithm is to the quality of the input word alignment.

We evaluate our approach through several experiments using English and French language pair from the Hansard corpus. We restrict the sentence length to be at most 20 words and obtain around 110K sentence pairs. Then we randomly select around 10K sentence pair as our testing set. In total, the French corpus consists of 994,564 words and 29,360 unique words; while the English corpus consists of 1,055,167 words and 20,138 unique words. Our experiment is conducted on English-to-French tasks on open testing set-up. We use GIZA++[1] as the implementation for the word-based IBM 4 model training and ISI ReWrite[2] to translate sentences in testing set. Translation performance is reported as BLEU scores, with appropriate confidence intervals computed.

4.1 Performance versus Original LOD

Figure 1 shows the performance of the LOD approach using the simplified algorithm in each iteration and juxtaposes it with that of original algorithm in the English-to-French task. We apply a *paired* $t-test$ (Koehn, 2004) to examine whether the performance difference is statistically significant.

The result of our experiments shows that the performance of our simplified algorithm is comparable and slightly above that of original algorithm in some iterations, although the *paired* $t-test$ shows that the performance difference is not statistically significant.

Initially, we expect to see a reduction in translation performance by simplifying the algorithm since the algorithm operates on a smaller set of alignments sacrificing the recall in generating translation candidates. The results suggest that the simplified algorithm of learning phrase translation works without sacrificing performance with the computational advantage.

In terms of computation, the simplified and

[1] http://www.fjoch.com/

[2] http://www.isi.edu/licensed-sw/rewrite-decoder/

the original algorithm differ only in the effort of translation model estimation. In the simplified algorithm, we estimate the translation model once while in the original algorithm, we have to estimate the translation in every iteration.

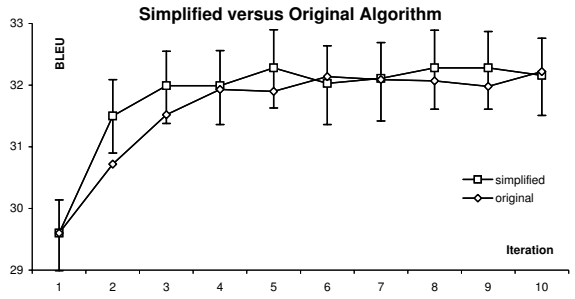

Figure 1. Performance comparison between the original and improved LOD algorithms. BLEU scores and confidence intervals with 95% statistical significance are shown.

4.2 Impact of Translation Model on Sensitivity

The original LOD algorithm uses the initial word-alignment produced by IBM model 4 translation model to learn phrase translations. IBM model 4 builds up the translation model from a set of sub-models: lexical, fertility, distortion and null model. The estimation of such a model is computationally expensive. Furthermore, finding an optimal word alignment efficiently is improbable, and approximations must be used. IBM model 3 represents a similar model, as complex as IBM model 4, but with a simpler distortion model. Exact solutions can be computed for simpler models such as IBM models 1 and 2.

In this experiment, we evaluate the performance of LOD approach using word alignments from different translation models. Figure 2 shows the result of this experiment in the English-to-French task for each iteration. The figure indicates that the LOD algorithm produces a significant improvement over a word-based approach even when using word alignment computed by simpler translation models.

Across the IBM translation model, we expect to see a positive correlation between the quality of underlying word alignment and the translation performance. The experimental results show that IBM model 4 gives the highest performance improvement and IBM model 1 gives the lowest performance gain which conforms with the intuition. However, the experimental results on IBM model 2 and IBM model 3 shows a contradictory result, with the performance improvement by IBM model 2 outperforms that by IBM model 3. The behavior confirms the same finding on similar task by (Koehn et al., 2003). To understand this behavior better, in Table 1, we report the percentage of $NULL$ alignment generated by each translation model in the first iteration of LOD algorithm.

IBM Model	$NULL$ (%)
1	27.6
2	21.9
3	38.0
4	35.7

Table 1. The percentage of $NULL$ alignment generated by IBM translation model

From Table 1, we observe that IBM model 3 generates a significantly higher percentage of $NULL$ alignment compared to any other model and IBM model 2 generates a relatively small percentage of $NULL$ alignment. We suspect that higher percentage of $NULL$ alignments leads to more undesired phrases in the generation step.

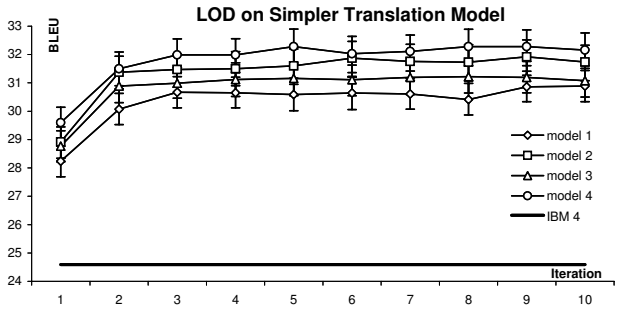

Figure 2. The translation performance behavior of LOD approach with respect to word alignment from different underlying translation model. IBM 4 produces the best performance, followed by IBM2 and IBM3. IBM 1 produces the worst result. The straight line represents the performance of word-based SMT as the baseline

5 Discussion

We propose a simple algorithm to learn phrase translation from parallel texts. In particular, we introduce a simplified level-of-detail (LOD) algorithm that achieves significant efficiency gains with while maintaining a high level of translation fidelity as scored by the BLEU metric.

The original LOD algorithm follows an iterative, agglomerative framework to overcome the issue of the unknown phrase problem and diffi-

culties with normalization of phrases of varying length. The original algorithm uses word alignment information from the IBM model 4 translation model during each iteration, although the estimation of this translation model is computationally expensive.

In this paper, we have simplified the original LOD algorithm by working with a smaller set of sub-phrase alignment from the previous iteration. This modification removes the necessity to re-estimate word alignments from each sentence pair. As this modification is introduced for all iterations aside from the initial iteration, it gives a significant computational advantage. Experimental results on English to French tasks using the Canadian Hansards show that results are comparable to the original algorithm. Both LOD algorithsm produce a significant improvement in the BLEU translation quality metric over word-based SMT.

We also experiment the use of word alignments resulting from employing simpler translation model than IBM Model 4. In particular we employed IBM Models 1 through 3 as alternatives to Model 4. Our results show that using IBM 2 produces a slightly lower but comparable performance, and has better performance than Model 3. The result is encouraging, since IBM 2 is simpler, faster and more efficient than IBM 4 and an exact solution can be obtained for the word alignment.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, Jun.

Kenneth W. Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th. Annual Meeting of the Association for Computational Linguistics*, pages 76–83.

Yamamoto Kaoru, Kudo Taku, Tsuboi Yuta, and Matsumoto Yuji. 2003. Learning sequence-to-sequence correspondences from parallel corpora via sequential pattern mining. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 73–80, May.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference*, pages 127–133, May/June.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–139, july.

I. Dan Melamed. 1997. Automatic discovery of non-compositional compounds in parallel data. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 97–108.

Robert C. Moore. 2003. Learning translations of named-entity phrases from parallel corpora. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Franz J. Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, Jun.

Hendra Setiawan, Haizhou Li, Min Zhang, and Beng Chin Ooi. 2005. Phrase-based statistical machine translation: A level of detail approach. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP-05)*, Oct.

Christoph Tillmann. 2003. A projection extension algorithm for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Ashish Venugopal, Stephan Vogel, and Alex Waibel. 2004. Effective phrase translation extraction from alignment models. In *Proceedings of 41st Annual Meeting of Association of Computational Linguistics*, pages 319–326, July.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of COLING '96: The 16th International Conference of Computational Linguistics*, pages 836–841.

Richard Zens and Hermann Ney. 2004. Im-

provements in phrase-based statistical machine translation. In *Proceedings of Conference on Human Language Technology*, pages 257–264, July.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2003. Integrated phrase segmentation and alignment algorithm for statistical machine translation. In *Proceedings of the Conference on Natural Language Processing and Knowledge Engineering*.