A Novel Graph-based Compact Representation of Word Alignment

Qun Liu[†‡] **Zhaopeng Tu**[‡] **Shouxun Lin**[‡]

[†]Centre for Next Generation Locolisation
Dublin City University
qliu@computing.dcu.ie

[‡]Key Lab. of Intelligent Info. Processing
Institute of Computing Technology, CAS
{tuzhaopeng,sxlin}@ict.ac.cn

Abstract

In this paper, we propose a novel compact representation called *weighted bipartite hypergraph* to exploit the fertility model, which plays a critical role in word alignment. However, estimating the probabilities of rules extracted from hypergraphs is an NP-complete problem, which is computationally infeasible. Therefore, we propose a divide-and-conquer strategy by decomposing a hypergraph into a set of independent subhypergraphs. The experiments show that our approach outperforms both 1-best and n-best alignments.

1 Introduction

Word alignment is the task of identifying translational relations between words in parallel corpora, in which a word at one language is usually translated into several words at the other language (*fertility model*) (Brown et al., 1993). Given that many-to-many links are common in natural languages (Moore, 2005), it is necessary to pay attention to the relations among alignment links.

In this paper, we have proposed a novel graph-based compact representation of word alignment, which takes into account the joint distribution of alignment links. We first transform each alignment to a bigraph that can be decomposed into a set of subgraphs, where all interrelated links are in the same subgraph (§ 2.1). Then we employ a weighted partite hypergraph to encode multiple bigraphs (§ 2.2).

The main challenge of this research is to efficiently calculate the fractional counts for rules extracted from hypergraphs. This is equivalent to the decision version of set covering problem, which is NP-complete. Observing that most alignments are not connected, we propose a divide-and-conquer strategy by decomposing a hypergraph into a set of independent subhypergraphs, which is computationally feasible in practice (§ 3.2). Experimental results show that our approach significantly improves translation performance by up to 1.3 BLEU points over 1-best alignments (§ 4.3).

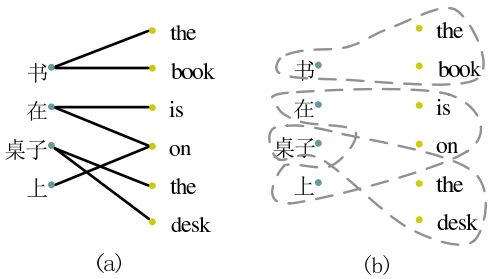

Figure 1: A bigraph constructed from an alignment (a), and its disjoint MCSs (b).

2 Graph-based Compact Representation

2.1 Word Alignment as a Bigraph

Each alignment of a sentence pair can be transformed to a bigraph, in which the two disjoint vertex sets S and T are the source and target words respectively, and the edges are word-by-word links. For example, Figure 1(a) shows the corresponding bigraph of an alignment.

The bigraph usually is not connected. A graph is called connected if there is a path between every pair of distinct vertices. In an alignment, words in a specific portion at the source side (i.e. a verb phrase) usually align to those in the corresponding portion (i.e. the verb phrase at the target side), and would never align to other words; and vice versa. Therefore, there is no edge that connects the words in the portion to those outside the portion.

Therefore, a bigraph can be decomposed into a unique set of *minimum connected subgraphs* (MCSs), where each subgraph is connected and does not contain any other MCSs. For example, the bigraph in Figure 1(a) can be decomposed into

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 358–363,
Sofia, Bulgaria, August 4-9 2013.

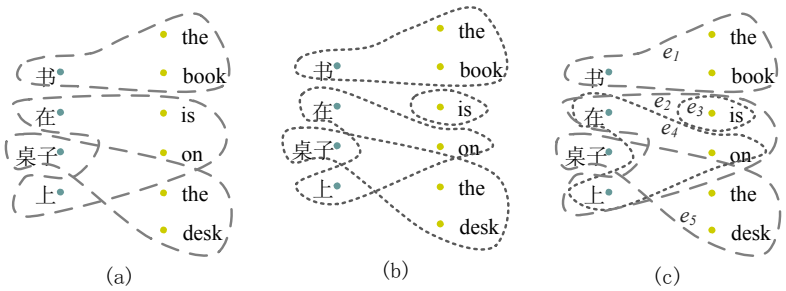

Figure 2: (a) One alignment of a sentence pair; (b) another alignment of the same sentence pair; (c) the resulting hypergraph that takes the two alignments as samples.

the MCSs in Figure 1(b). We can see that all interrelated links are in the same MCS. These MCSs work as fundamental units in our approach to take advantage of the relations among the links. Hereinafter, we use bigraph to denote the alignment of a sentence pair.

2.2 Weighted Bipartite Hypergraph

We believe that offering more alternatives to extracting translation rules could help improve translation quality. We propose a new structure called *weighted bipartite hypergraph* that compactly encodes multiple alignments.

We use an example to illustrate our idea. Figures 2(a) and 2(b) show two bigraphs of the same sentence pair. Intuitively, we can encode the union set of subgraphs in a bipartite hypergraph, in which each MCS serves as a hyperedge, as in Figure 2(c). Accordingly, we can calculate how well a hyperedge is by calculating its relative frequency, which is the probability sum of bigraphs in which the corresponding MCS occurs divided by the probability sum of all possible bigraphs. Suppose that the probabilities of the two bigraphs in Figures 2(a) and 2(b) are 0.7 and 0.3, respectively. Then the weight of e_1 is 1.0 and e_2 is 0.7. Therefore, each hyperedge is associated with a weight to indicate how well it is.

Formally, a *weighted bipartite hypergraph* H is a triple $\langle S, T, E \rangle$ where S and T are two sets of vertices on the source and target sides, and E are hyperedges associated with weights. Currently, we estimate the weights of hyperedges from an n-best list by calculating relative frequencies:

$$w(e_i) = \frac{\sum_{BG \in \mathcal{N}} p(BG) \times \delta(BG, g_i)}{\sum_{BG \in \mathcal{N}} p(BG)}$$

Here $\mathcal{N}$ is an n-best bigraph (i.e., alignment) list, $p(BG)$ is the probability of a bigraph BG in the n-best list, g_i is the MCS that corresponds to e_i, and $\delta(BG, g_i)$ is an indicator function which equals 1 when g_i occurs in BG, and 0 otherwise.

It is worthy mentioning that a hypergraph encodes much more alignments than the input n-best list. For example, we can construct a new alignment by using hyperedges from different bigraphs that cover all vertices.

3 Graph-based Rule Extraction

In this section we describe how to extract translation rules from a hypergraph (§ 3.1) and how to estimate their probabilities (§ 3.2).

3.1 Extraction Algorithm

We extract translation rules from a hypergraph for the hierarchical phrase-based system (Chiang, 2007). Chiang (2007) describes a rule extraction algorithm that involves two steps: (1) extract phrases from 1-best alignments; (2) obtain variable rules by replacing sub-phrase pairs with nonterminals. Our extraction algorithm differs at the first step, in which we extract phrases from hypergraphs instead of 1-best alignments. Rather than restricting ourselves by the alignment consistency in the traditional algorithm, we extract all possible candidate target phrases for each source phrase. To maintain a reasonable rule table size, we filter out less promising candidates that have a *fractional count* lower than a threshold.

3.2 Calculating Fractional Counts

The *fractional count* of a phrase pair is the probability sum of the alignments with which the phrase pair is consistent (§3.2.2), divided by the probability sum of all alignments encoded in a hypergraph (§3.2.1) (Liu et al., 2009).

Intuitively, our approach faces two challenges:

1. How to calculate the probability sum of all alignments encoded in a hypergraph (§3.2.1)?
2. How to efficiently calculate the probability sum of all consistent alignments for each phrase pair (§3.2.2)?

3.2.1 Enumerating All Alignments

In theory, a hypergraph can encode all possible alignments if there are enough hyperedges. However, since a hypergraph is constructed from an n-best list, it can only represent partial space of all alignments ($p(A|H) < 1$) because of the limiting size of hyperedges learned from the list. Therefore, we need to enumerate all possible alignments in a hypergraph to obtain the probability sum $p(A|H)$.

Specifically, generating an alignment from a hypergraph can be modelled as finding a *complete hyperedge matching*, which is a set of hyperedges without common vertices that matches all vertices. The probability of the alignment is the product of hyperedge weights. Thus, enumerating all possible alignments in a hypergraph is reformulated as finding all *complete hypergraph matchings*, which is an NP-complete problem (Valiant, 1979).

Similar to the bigraph, a hypergraph is also usually not connected. To make the enumeration practically tractable, we propose a *divide-and-conquer* strategy by decomposing a hypergraph H into a set of independent subhypergraphs $\{h_1, h_2, \ldots, h_n\}$. Intuitively, the probability of an alignment is the product of hyperedge weights. According to the divide-and-conquer strategy, the probability sum of all alignments A encoded in a hypergraph H is:

$$p(A|H) = \prod_{h_i \in H} p(A_i|h_i)$$

Here $p(A_i|h_i)$ is the probability sum of all sub-alignments A_i encoded in the subhypergraph h_i.

3.2.2 Enumerating Consistent Alignments

Since a hypergraph encodes many alignments, it is unrealistic to enumerate all consistent alignments explicitly for each phrase pair.

Recall that a hypergraph can be decomposed to a list of independent subhypergraphs, and an alignment is a combination of the sub-alignments from the decompositions. We observe that a phrase pair is absolutely consistent with the sub-alignments from some subhypergraphs, while possibly consistent with the others. As an example,

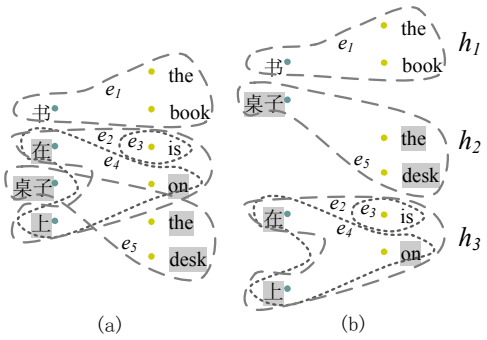

Figure 3: A hypergraph with a candidate phrase in the grey shadow (a), and its independent subhypergraphs $\{h_1, h_2, h_3\}$.

consider the phrase pair in the grey shadow in Figure 3(a), it is consistent with all sub-alignments from both h_1 and h_2 because they are outside and inside the phrase pair respectively, while not consistent with the sub-alignment that contains hyperedge e_2 from $h3$ because it contains an alignment link that crosses the phrase pair.

Therefore, to calculate the probability sum of all consistent alignments, we only need to consider the *overlap subhypergraphs*, which have at least one hyperedge that crosses the phrase pair. Given a overlap subhypergraph, the probability sum of consistent sub-alignments is calculated by subtracting the probability sum of the sub-alignments that contain crossed hyperedges, from the probability sum of all sub-alignments encoded in a hypergraph.

Given a phrase pair P, let OS and NS denotes the sets of overlap and non-overlap subhypergraphs respectively ($NS = H - OS$). Then

$$p(A|H,P) = \prod_{h_i \in OS} p(A_i|h_i,P) \prod_{h_j \in NS} p(A_j|h_j)$$

Here the phrase pair is absolutely consistent with the sub-alignments from non-overlap subhypergraphs (*NS*), and we have $p(A|h,P) = p(A|h)$. Then the fractional count of a phrase pair is:

$$c(P|H) = \frac{p(A|H,P)}{p(A|H)} = \frac{\prod_{h_i \in OS} p(A|h_i,P)}{\prod_{h_i \in OS} p(A|h_i)}$$

After we get the fractional counts of translation rules, we can estimate their *relative frequencies* (Och and Ney, 2004). We follow (Liu et al., 2009; Tu et al., 2011) to learn lexical tables from n-best lists and then calculate the lexical weights.

Rules from...	Rules	MT03	MT04	MT05	Avg.
1-best	257M	33.45	35.25	33.63	34.11
10-best	427M	34.10	35.71	34.04	34.62
Hypergraph	426M	34.71	36.24	34.41	35.12

Table 1: Evaluation of translation quality.

4 Experiments

4.1 Setup

We carry out our experiments on Chinese-English translation tasks using a reimplementation of the hierarchical phrase-based system (Chiang, 2007). Our training data contains 1.5 million sentence pairs from LDC dataset.[1] We train a 4-gram language model on the Xinhua portion of the GIGAWORD corpus using the SRI Language Toolkit (Stolcke, 2002) with modified Kneser-Ney Smoothing (Kneser and Ney, 1995). We use minimum error rate training (Och, 2003) to optimize the feature weights on the MT02 testset, and test on the MT03/04/05 testsets. For evaluation, case-insensitive NIST BLEU (Papineni et al., 2002) is used to measure translation performance.

We first follow Venugopal et al. (2008) to produce n-best lists via GIZA++. We produce 10-best lists in two translation directions, and use "grow-diag-final-and" strategy (Koehn et al., 2003) to generate the final n-best lists by selecting the top n alignments. We re-estimated the probability of each alignment in the n-best list using renormalization (Venugopal et al., 2008). Finally we construct weighted alignment hypergraphs from these n-best lists.[2] When extracting rules from hypergraphs, we set the pruning threshold $t = 0.5$.

4.2 Tractability of Divide-and-Conquer Strategy

Figure 4 shows the distribution of vertices (hyperedges) number of the subhypergraphs. We can see that most of the subhypergraphs have just less than two vertices and hyperedges.[3] Specifically, each subhypergraph has 2.0 vertices and 1.4 hyperedges on average. This suggests that the divide-and-conquer strategy makes the extraction computationally tractable, because it greatly reduces the number of vertices and hyperedges. For computational tractability, we only allow a subhypergraph has at most 5 hyperedges. [4]

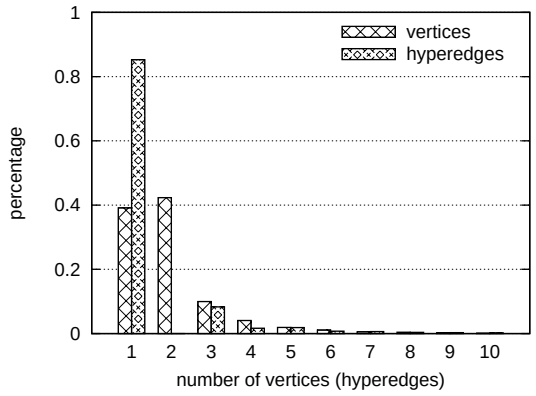

Figure 4: The distribution of vertices (hyperedges) number of the subhypergraphs.

4.3 Translation Performance

Table 1 shows the rule table size and translation quality. Using n-best lists slightly improves the BLEU score over 1-best alignments, but at the cost of a larger rule table. This is in accord with intuition, because all possible translation rules would be extracted from different alignments in n-best lists without pruning. This larger rule table indeed leads to a high rule coverage, but in the meanwhile, introduces translation errors because of the low-quality rules (i.e., rules extracted only from low-quality alignments in n-best lists). By contrast, our approach not only significantly improves the translation performance over 1-best alignments, but also outperforms n-best lists with a similar-scale rule table. The absolute improvements of 1.0 BLEU points on average over 1-best alignments are statistically significant at $p < 0.01$ using *sign-test* (Collins et al., 2005).

[1] The corpus includes LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

[2] Here we only use 10-best lists, because the alignments beyond top 10 have very small probabilities, thus have negligible influence on the hypergraphs.

[3] It's interesting that there are few subhypergraphs that have exactly 2 hyperedges. In this case, the only two hyperedges fully cover the vertices and they differ at the word-by-word links, which is uncommon in n-best lists.

[4] If a subhypergraph has more than 5 hyperedges, we forcibly partition it into small subhypergraphs by iteratively removing lowest-probability hyperedges.

Rules from...	Shared		Non-shared		All	
	Rules	BLEU	Rules	BLEU	Rules	BLEU
10-best	1.83M	32.75	2.81M	30.71	4.64M	34.62
Hypergraph	1.83M	33.24	2.89M	31.12	4.72M	35.12

Table 2: Comparison of rule tables learned from n-best lists and hypergraphs. "All" denotes the full rule table, "Shared" denotes the intersection of two tables, and "Non-shared" denotes the complement. Note that the probabilities of "Shared" rules are different for the two approaches.

Why our approach outperforms n-best lists? In theory, the rule table extracted from n-best lists is a subset of that from hypergraphs. In practice, however, this is not true because we pruned the rules that have fractional counts lower than a threshold. Therefore, the question arises as to how many rules are shared by n-best and hypergraph-based extractions. We try to answer this question by comparing the different rule tables (filtered on the test sets) learned from n-best lists and hypergraphs. Table 2 gives some statistics. "All" denotes the full rule table, "Shared" denotes the intersection of two tables, and "Non-shared" denotes the complement. Note that the probabilities of "Shared" rules are different for the two approaches. We can see that both the "Shared" and "Non-shared" rules learned from hypergraphs outperform n-best lists, indicating: (1) our approach has a better estimation of rule probabilities because we estimate the probabilities from a much larger alignment space that can not be represented by n-best lists, (2) our approach can extract good rules that cannot be extracted from any single alignments in the n-best lists.

5 Related Work

Our research builds on previous work in the field of graph models and compact representations. Graph models have been used before in word alignment: the search space of word alignment can be structured as a graph and the search problem can be reformulated as finding the optimal path though this graph (e.g., (Och and Ney, 2004; Liu et al., 2010)). In addition, Kumar and Byrne (2002) define a graph distance as a loss function for minimum Bayes-risk word alignment, Riesa and Marcu (2010) open up the word alignment task to advances in hypergraph algorithms currently used in parsing. As opposed to the search problem, we propose a graph-based compact representation that encodes multiple alignments for machine translation.

Previous research has demonstrated that compact representations can produce improved results by offering more alternatives, e.g., using forests over 1-best trees (Mi and Huang, 2008; Tu et al., 2010; Tu et al., 2012a), word lattices over 1-best segmentations (Dyer et al., 2008), and weighted alignment matrices over 1-best word alignments (Liu et al., 2009; Tu et al., 2011; Tu et al., 2012b). Liu et al., (2009) estimate the link probabilities from n-best lists, while Gispert et al., (2010) learn the alignment posterior probabilities directly from IBM models. However, both of them ignore the relations among alignment links. By contrast, our approach takes into account the joint distribution of alignment links and explores the fertility model past the link level.

6 Conclusion

We have presented a novel compact representation of word alignment, named weighted bipartite hypergraph, to exploit the relations among alignment links. Since estimating the probabilities of rules extracted from hypergraphs is an NP-complete problem, we propose a computationally tractable divide-and-conquer strategy by decomposing a hypergraph into a set of independent subhypergraphs. Experimental results show that our approach outperforms both 1-best and n-best alignments.

Acknowledgement

The authors are supported by 863 State Key Project No. 2011AA01A207, National Key Technology R&D Program No. 2012BAH39B03 and National Natural Science Foundation of China (Contracts 61202216). Qun Liu's work is partially supported by Science Foundation Ireland (Grant No.07/CE/I1142) as part of the CNGL at Dublin City University. We thank Junhui Li, Yifan He and the anonymous reviewers for their insightful comments.

References

Peter E. 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.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

M. Collins, P. Koehn, and I. Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 531–540.

Adrià de Gispert, Juan Pino, and William Byrne. 2010. Hierarchical phrase-based translation grammars extracted from alignment posterior probabilities. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 545–554.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54.

Shankar Kumar and William Byrne. 2002. Minimum Bayes-risk word alignments of bilingual texts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 140–147.

Yang Liu, Tian Xia, Xinyan Xiao, and Qun Liu. 2009. Weighted alignment matrices for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1017–1026.

Yang Liu, Qun Liu, and Shouxun Lin. 2010. Discriminative word alignment by linear modeling. *Computational Linguistics*, 36(3):303–339.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 206–214.

Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 81–88, October.

Franz J. Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Jason Riesa and Daniel Marcu. 2010. Hierarchical search for word alignment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 157–166.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of Seventh International Conference on Spoken Language Processing*, volume 3, pages 901–904. Citeseer.

Zhaopeng Tu, Yang Liu, Young-Sook Hwang, Qun Liu, and Shouxun Lin. 2010. Dependency forest for statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1092–1100.

Zhaopeng Tu, Yang Liu, Qun Liu, and Shouxun Lin. 2011. Extracting hierarchical rules from a weighted alignment matrix. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1294–1303.

Zhaopeng Tu, Wenbin Jiang, Qun Liu, and Shouxun Lin. 2012a. Dependency forest for sentiment analysis. In *Springer-Verlag Berlin Heidelberg*, pages 69–77.

Zhaopeng Tu, Yang Liu, Yifan He, Josef van Genabith, Qun Liu, and Shouxun Lin. 2012b. Combining multiple alignments to improve machine translation. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1249–1260.

Leslie G Valiant. 1979. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2008. Wider pipelines: n-best alignments and parses in mt training. In *Proceedings of AMTA*, pages 192–201.