Analogy-based on-line reordering approach for machine translation

Hao Wang and Yves Lepage

Graduate School of Information, Production and Systems,
Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka, Japan
{oko_ips@ruri, yves.lepage@}waseda.jp

Abstract

In this paper, we describe a novel approach based on analogy and tree-structure to reordering the translation fragments at run-time. This method is inspired from example-based machine translation (EBMT). It does not require any syntax information in target side and examples are extracted on demand in the stage of preprocessing. During decoding, the translate-engine employs the examples to translate the input and to reorder the output on-the-fly. Our experiments show that it achieved a reasonable result on the translation tasks of English-Japanese and Chinese-Japanese .

1. Introduction

There are many cases in which the natural translation of one language into another results in a very different form than the original (Dorr, 1993), e.g., *kusuri wo nonda* (in Japanese, to drink) is usually translated as *I took medicine* (in English) and *wo chiguo yao le* (in Chinese, to take). The verb *nonda* is translated into *took* and *chiguo le*. The underlying translation unit, usually called phrase, is not easy to handle in the languages with very different orders and prevents the derivation of generalizations from training corpora.

Although Phrase-based Statistical Machine Translation (PB-SMT) is considered as the state-of-the-art, when translating, it may overlook useful linguistic information contained in training data. Moreover, the reordering model, which is contained in SMT systems, increases the complexity of the mathematical model of decoding. It also has been mentioned (Isozaki et al., 2010) that SMT could not outperform rule-based translation for language pairs with highly different word orders.

Since language translation is burdened with so many decisions that are hard to formalize, it may be better to learn how to translate from the past translation examples (Cicekli and Güvenir, 2001). This is the basic idea of Example-Based Machine Translation (EBMT). EBMT is not newborn, a rich diversity of models can be found since it has been proposed. To cite a few, Lepage and Denoual, 2005) present the pure string-based EBMT with no additional information; Langlais and Gotti, 2006) explore the potential of using "tree-phrase" model to combine the source-language treelet with target language phrase. (Liu et al., 2006) propose a translation model based on tree-to-string alignment template (TAT) which describes the alignment between a source parse tree and a target string. This topic still needs further exploration.

Given a set of sentences in the source language (from which one is translating) and their corresponding translations in the target language, EBMT systems use those examples to translate other similar source-language sentences into the target language. The basic premise is that, if a previously translated sentence occurs again, the same translation is likely to be correct again. Moreover, corpus-based machine translation systems should prefer longer units because they naturally convey local context and local reordering.

With respect to machine translation divergences (Dorr, 1993) and data-driven MT (Dolan et al., 2002) , in this paper, we propose a hybrid machine translation approach (analogy-based EBMT, aka ABMT), which borrows ideas from EBMT and SMT and takes the advantages of both them. Our ABMT system will appear to easy understand to those familiar with Syntax-based MT and EBMT systems at the same time. Like all EBMT systems, ABMT performs dynamic matching against the training corpus at runtime (i.e., on-the-fly), rather than pre-producing a static phrase table in advance. Moreover, we equip it with an analogy solver: our system will be able to handle the distinct structure mappings and reorder the words on-line. During the decoding phase, it relies on the structure of the syntactic tree of the source sentence, dynamically and recursively implements a variant of chart parsing.

The main part of this article is devoted to describing the components of decoding, with particular emphasis on those which are unique to ABMT and depart from common practice in data-driven MT systems. We discuss some basic notions and related work in Section 2. We present how to translate using analogy in decoding in Section 3.. In section 4., we describe our experiments, and discuss our conclusions and future work in section 5..

2. Translation using analogy

2.1. The principle

Given a new sentence to translate, an EBMT engine first looks for the sentence in the memory. If the sentence is found in the memory, the translation engine just outputs its translation without any further computation. This is the most felicitous case. However, most sentences do not already exist in the training data.

(Lepage and Denoual, 2005) proposed that computation should be carried out using the principle of corre-

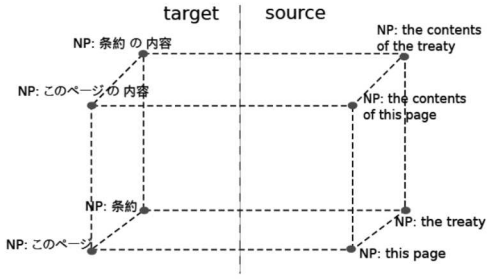

Figure 1: View of the homomorphism parallelopiped: four terms in each language form a monolingual proportional analogy.

sponding proportional analogies between two language domains. To translate a sentence A, the engine basically solves all possible analogical equations of the type:

$$A : B :: C : x$$

where B and C are two source phrases in the memory. If the solution of the equation $x = D$ can be found in the memory (i.e., in the training set), then its translation $\hat{D}$ is known and one tries to solve the analogical equation (in the target language):

$$y : \hat{B} :: \hat{C} : \hat{D}$$

If a solution $y = \hat{A}$ exists, it is a possible translation of A. Figure 1 illustrates the principle of translation using analogy. When no solution at all can be generated by analogy, the engine backs off to the basic behavior of EBMT and composes these translation fragments into the appropriate target text.

2.2. Overview of analogy-based EBMT

As mentioned above, the method involves two basic operations: searching for possible analogical equations, and solving them. The latter of these is relatively straightforward, among the implementations, (Lepage, 1998) has been generally accepted as the most efficient. It comes with an efficient algorithm to solve analogical equations between sentences according to the edit distances under the constraint of $d(A, B) = d(C, D)$ and $d(A, C) = d(B, D)$. However, more problematic in this approach is, given A, how to find the proper triple of (B, C, D) which satisfies the analogical equation $[A : B :: C : D]$. Commentators (Somers et al., 2009) agree that the search is intractable or unmanageable, except for "toy problems". Even if the analogy solver is quite efficient, it is obvious that some heuristics are needed to reduce the search space.

The first ALEPH system described in (Lepage and Denoual, 2005) uses very simple data (e.g., the average length of the Chinese sentences in characters is only 9.4). Since the examples are "quite short", it has not been compared with the state-of-art systems. After that, the ALEPH system evolved into a new system, named GREYC (Lepage and Lardilleux, 2007), for the 2007 and 2008 IWSLT campaigns. The main change was the addition of a preprocessing stage which adds subsentential (word) alignments to the example base.

3. Proposed method

3.1. Analogy-based decoding

Because capturing the hierarchical structure in the target side remains hard, using the existing hierarchical structure of the source language seems more reliable.

We first parse a source sentence into a parse tree and then decompose the source tree into treelets. Based on the tree structure obtained by a consistent parser and the alignment between words in the parallel sentences, we create the correspondences between treelets (source) to phrases (target) across the two languages. Figure 2 shows an example for our tree-to-string model. In order to translate a

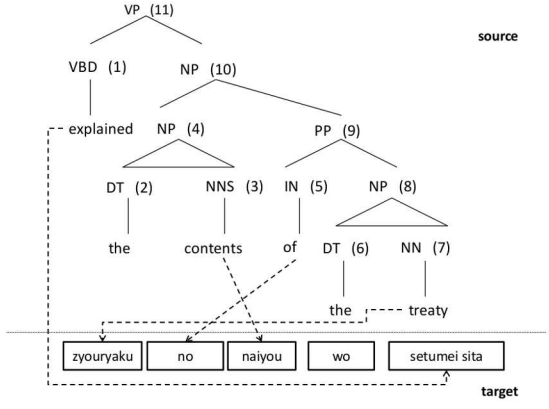

Figure 2: Example of tree-to-string matching from a English phrase to the translation in Japanese.

given phrase $A :=$ [*the contents of the treaty*] and the corresponding treelet T_A, we first apply approximate pattern matching to find a similar example-candidate $B :=$ [*the contents of this page*] in the training set. There have been various of methods (e.g., approximate pattern matching) to solve this problem. The parsed tree-structure T_B of B is obtained at the same time. After analysis T_A with T_B, it is easy to extract the different children T_C and T_D. If we keep the string of leaves instead of the four elements in the above equation, so the $C :=$ [*the treaty*] and $D :=$ [*this page*]. The equation $[A : B :: C : D]$ reduces as:

$$\begin{matrix} \textit{the contents} \\ \textit{of the treaty} \end{matrix} : \begin{matrix} \textit{the contents} \\ \textit{of this page} \end{matrix} :: \textit{the treaty} : \textit{this page}$$

In the phrase-based model, translations are built from left to right. However, in tree transfer models, the fundamental element of the decoding process has changed. To translate a given source sentence, we employ a parser to produce a parse tree. Moving bottom-up through the source parse tree, we traverse the pointed input treelet rooted at each node with a post-order transversal as shown in Figure 2. Our bottom-up decoding traverses (as Table 5) the given treelet in post-order. Algorithm 1 shows the details of performing recursive translation. In the experiment we found that it is more effective if the examples were sampled according to the similarity in structure or words. Figure 3 shows the distribution of the example used in decoding for the given phrase with different length. As Figure 3

Algorithm 1 Analogy-based Decoding

procedure RECURSIVE–TRANSLATE($T_A, \hat{e}$)
  ▷ $\hat{e}$ set of the examples
  for $\tau \in T_A.children$ && $\notin$ translated **do**
    recursive-translate(τ)
  $\theta \leftarrow \emptyset$ ▷ Initial list.
  for $example(B, T_B) \in \hat{e}$ in memory **do**:
    $T_C, T_D \leftarrow \emptyset$; ▷ initial list.
    for $t_1 \in \{T_A.children - T_B.children\}$ **do**
      $T_C \leftarrow T_C \cup t_1$; ▷ Add element.
    for $t_2 \in \{T_B.children - T_A.children\}$ **do**
      $T_D \leftarrow T_D \cup t_2$; ▷ Add element.
    $[A : B :: C : D]$; ▷ Validate analogy equation.
    $\hat{B}, \hat{C}, \hat{D} = \varphi(B), \varphi(C), \varphi(D)$; ▷ Translate.
    $[y : \hat{B} :: \hat{C} : \hat{D}]$; ▷ Solve analogy equation.
    $\theta \leftarrow \theta \cup y$
  $\hat{y} = \arg\max_{y \in \theta} Score(y)$ ▷ Reranking function.
  return $Top_k(\hat{y})$ ▷ Select k-best list.

indicates, when translating, the system prefers to use the examples in which the lengths of words are similar to the input phrase. The length of most of the examples is less than 10 (in words).

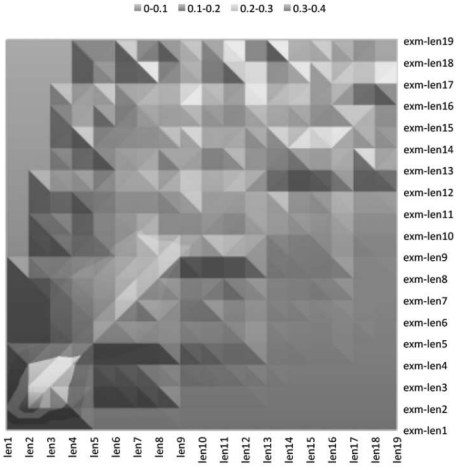

Figure 3: Distribution of the lengths of the used examples at run-time and the corresponding phrases in the source language.

3.2. On-line reordering

Reordering models for phrase-based translation are typically part of the log-linear framework which forms the basis of many statistical machine translation systems. It really complicates the mathematical model of decoding. In this section, we propose a novel method to build an analogy-based MT system which reorders the subtranslations with an analogy solver recursively on-line.

Given a solved analogy equation in source language:

$$\begin{matrix}\textit{the contents}\\ \textit{of the treaty}\end{matrix} : \begin{matrix}\textit{the contents}\\ \textit{of this page}\end{matrix} :: \textit{the treaty} : \textit{this page}$$

Source	Target
the treaty	zyouryaku
explained	setumei sita
the contents	naiyou
of	no

Table 1: An example of phrase table for analogy used in the decoding, from English to Japanese

	Input	Example
NP:	the treaty	the law / houritu
NP:	the contents of the treat	the contents of this page/ kono peizi no naiyou
VP:	explained the contents of the treaty	explained the situation/ zyouryaku wo setumei sita

Table 2: an example of example table for analogy used in the decoding, from English to Japanese

On the target side, we employ proportional analogy to generate the translation using the following equation:

$$y : \begin{matrix}\textit{kono peizi no}\\ \textit{naiyou}\end{matrix} :: \textit{zyouryaku} : \textit{kono peizi}$$

In the above equation, given the known phrase translations in order, analogy can automatically generate the solution.

$$y = \textit{zyouryaku no naiyou}$$

One advantage of our model is that analogy can be automatically acquired to capture linguistically motivated reordering at both low level (character, if we allow analogy on characters) and high levels (phrase/chunk).

3.3. Pruning as reranking

We make beam as follows in the source tree, we build a beam. In other words, hypotheses covering the same source words (e) of a treelet (T_e) are grouped in a beam. There are 2 cases in translation. The first one is when examples in memory are found and the analogy solver outputs the solution. The second one is when, in some cases, no example can be used or no examples are found. We handle the first case by just building a new entry for the translation in the chart. The following features of the entry are computed: language model P_{LM}, inverse lexical weighting $\phi(f|e)$, direct lexical weighting $\phi'(e|f)$ and word penalty ρ. To enable more meaningful comparisons, we define a heuristic. We sort the entries $\tilde{e}$ for a given tuple $(\tilde{e}, T_{\tilde{e}})$ with the formula:

$$h(\tilde{e}) = \log[P_{LM}(\tilde{e})^{\lambda_i} \phi(\tilde{e}|f)^{\lambda_j} \phi'(f|\tilde{e})^{\lambda_k} \rho^{\lambda_m}] \quad (1)$$

When comparing items for pruning (and only for pruning), we add this heuristic function to the score of each item.

A special case occurs when translating (C, T_C) or (D, T_D). Because T_C is a sequence of children-nodes for the current node, the combination of sub-translations is complex. Since the situation is very similar to k-best list

		en-ja		zh-ja	
train	lines	329,874		618,184	
	words	11.5M	11.9M	18.3M	21.9M
	mean	26.21	27.07	27.44	32.74
	± std.dev	19.27	19.79	14.02	16.26
tune	lines	500		500	
	words	13.9K	15.6K	14.2K	16.2K
	mean	27.92	31.36	28.33	32.32
	± std.dev	21.83	24.78	16.20	18.05
test	lines	200		200	
	words	3.2K	3.6K	5.6K	6.2K
	mean	16.35	18.16	28.04	31.15
	± std.dev	14.69	17.87	31.15	16.67

Table 3: Statistics on the parallel corpus used in experiments

generation, we employ beam search to expand a hypothesis in a beam from left to right. Here we only take inverse lexical weighting and direct lexical weighting into consideration and add the new hypothesis to the corresponding beam monotonically as $h_1(\cdot)$.

$$h_1(e) = \log[\phi(e|f)^{\lambda_j}\phi'(e|f)^{\lambda_k}] \quad (2)$$

For the second case, when combining two entries in the chart as a monotonic process, cube pruning (Chiang, 2007) is used to adjust the trade-off between search speed and translation accuracy. If i is the index of the last word in e_1 and j is the index of the start word in e_2, the following heuristic is used for combination:

$$h_2(e_1, e_2) = h(e_1)+h(e_2)+P_{LM}(e_1[i-1, i], e_2[j, j+1]) \quad (3)$$

At each node, this reranks the translation candidates and imposes two restrictions in pruning: histogram pruning and threshold pruning, to reduce the search space.

4. Experiment results

In order to evaluate our system, we conducted translation experiments on two language pairs: English-Japanese (en-ja) and Chinese-Japanese (zh-ja). For en-ja, we evaluated on the KFTT[1] and compared our system with a baseline system. For zh-ja, we used parallel scientific paper excerpts from the ASPEC[2] corpus and compared against a similar baseline system. Table 3 summarizes some statistics on the experiment data. To translate a source sentence, we employ a parser to produce a parse tree. The Berkeley Parser (Petrov et al., 2006) is used to parse the text. The source side of the corpus is stored in a suffix-array using the DC3 / skew algorithm proposed by (Kärkkäinen and Sanders, 2003). The language model storage of target language uses the implementation in KenLM (Heafield, 2011). For tuning, the optimal weights for each feature are estimated using the minimum error rate training (MERT) algorithm (Och, 2003) and parameter optimization with Z-MERT (Zaidan, 2009).

[1]http://www.phontron.com/kftt/

[2]http://lotus.kuee.kyoto-u.ac.jp/ASPEC/

Method	Acceptable	Smoothing	Near-natural
ABMT	49%	32%	58%
MOSES	53%	35%	55%

Table 4: Human evaluation for 100 samples from the subtask of zh-ja translation.

To assess the contribution of the analogy in decoding, we propose to compare two systems: our analogy-based machine translation (ABMT) and a system built using Moses. The two baseline systems for the two language pairs are based on the open-source GIZA++/Moses pipeline without the reordering models (setting the features weights as 0 when decoding).

The first system is trained using the KFTT parallel corpus. The second one is trained on the ASPEC-JC. For Japanese, the segmentation tool used is kytea[3]. GIZA++ (Och and Ney, 2003) is used to generate word alignments. An analysis of human evaluation is shown in Table 4 and a series of automatic metrics are listed in Table 6. Though in BLEU, ABMT is behind both in en-ja and zh-ja, in human evaluation, there is not too much difference.

5. Conclusion

In this paper, we presented a hierarchical tree-to-string model for analogy-based machine translation which can be seen as a compromise between SMT and EBMT. With the help of syntax tree structure, our model learns rules automatically using analogy from the extracted examples. It searches for the best derivation with a heuristic function. Our model employs syntax for decoding directly. We investigated the distribution of examples usable in the decoding for building a hybrid MT system. We also conducted experiments on two language pairs. To our disappointment, even though we tried to catch up with the most state-of-art MT system MOSES, the results still lie behind. The advantages of our system, which is an EBMT system, should be seen when the sentences in the test set are more similar to the sentences in the training set. There is a positive feature that the decoder in our system is much simpler than that in MOSES, because the ordering stage is accomplished by comparing the input with examples and learning the order of words. Further efforts and experiments will be put and conducted in the future so as to investigate the issue of tree binarization to reduce the complexity in the decoding and to improve the tuning method in the future.

Acknowledgments

This work is supported in part by China Scholarship Council (CSC) under the CSC Grant No.201406890026 is acknowledged. We also thank the anonymous reviewers for their insightful comments.

6. References

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

[3]http://www.phontron.com/kytea/

Step	Action	Example (B)	A	C, D	Chart Table
0	s		[.]	[.], [.]	$S \leftarrow \emptyset$
1	c		[explained]	[.], [.]	$S \cup$ [100000]
2	s		[the]	[.], [.]	
3	s		[contents]	[.], [.]	
4	c		[the contents]	[.], [.]	$S \cup$ [011000]
5	c		[of]	[.], [.]	$S \cup$ [000100]
6	s		[the]	[.], [.]	
7	s		[treaty]	[.], [.]	
8	c		[the treaty]	[.], [.]	$S \cup$ [000011]
8$\star$	a	the law	[the treaty]	[law], [treaty]	$S \cup$ [000011]
9	s		[of the treaty]	[.], [.]	
10	a	the contents of this page	[the contents of the treaty]	[this page], [the treaty]	$S \cup$ [001111]
11	a	explained the situation	[explained the contents of the treaty]	[the situation], [the contents of the treaty]	$S \cup$ [111111]

Table 5: Simulation of bottom-up translation process for the derivation of a long phrase: *explains the contents of the treaty* in Figure 2. Actions: s, scan; a, analogy; c, copy translation from phrase table. The column of [] gives the position of source words the entry covers. $\star$ means that there are multi-operations in this step.

	en-ja					zh-ja				
MOSES	WER	BLEU	NIST	TER	RIBES	WER	BLEU	NIST	TER	RIBES
Baseline	0.7406	16.19	4.3516	0.7501	0.6762	0.6401	20.91	5.7588	0.6543	0.7222
+reordering model	0.7114	22.47	5.8200	0.7154	0.6870	0.5681	31.28	6.2936	0.5701	0.7396
ABMT	WER	BLEU	NIST	TER	RIBES	WER	BLEU	NIST	TER	RIBES
word-to-word	0.7225	8.74	3.3503	0.7652	0.6150	0.5818	17.01	5.0360	0.6173	0.7334
+examples	0.9375	16.14	4.1246	0.8403	0.6629	0.5535	23.37	5.8683	0.5878	0.7433
++analogy	0.7350	18.25	4.5831	0.7553	0.6575	0.5404	25.65	6.0270	0.5814	0.7604

Table 6: Evaluation results.

Cicekli, Ilyas and H Altay Güvenir, 2001. Learning translation templates from bilingual translation examples. *Applied Intelligence*, 15(1):57–76.

Dolan, William B, Jessie Pinkham, and Stephen D Richardson, 2002. *MSR-MT: The Microsoft Research machine translation system*. Springer.

Dorr, Bonnie Jean, 1993. *Machine translation: a view from the Lexicon*. MIT press.

Heafield, Kenneth, 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics.

Isozaki, Hideki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh, 2010. Head finalization: A simple reordering rule for sov languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Association for Computational Linguistics.

Kärkkäinen, Juha and Peter Sanders, 2003. Simple linear work suffix array construction. In *Automata, Languages and Programming*. Springer, pages 943–955.

Langlais, Philippe and Fabrizio Gotti, 2006. EBMT by tree-phrasing. *Machine Translation*, 20(1):1–23.

Lepage, Yves, 1998. Solving analogies on words: an algorithm. In *Proceedings of the 36th ACL and 17th COLING-Volume 1*. Association for Computational Linguistics.

Lepage, Yves and Etienne Denoual, 2005. The purest EBMT system ever built: no variables, no templates, no training, examples, just examples, only examples. In *Proceedings of the MT Summit X, Second Workshop on Example-Based Machine Translation*.

Lepage, Yves and Adrien Lardilleux, 2007. The GREYC machine translation system for the iwslt 2007 evaluation campaign. In *IWSLT 2007*.

Liu, Yang, Qun Liu, and Shouxun Lin, 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st COLING and the 44th ACL*. Association for Computational Linguistics.

Och, Franz Josef, 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics.

Och, Franz Josef and Hermann Ney, 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Somers, Harold, Sandipan Dandapat, and Sudip Kumar Naskar, 2009. A review of EBMT using proportional analogies.

Zaidan, Omar, 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.