A Rule-Augmented Statistical Phrase-based Translation System

Cong Duy Vu Hoang[†], AiTi Aw[†] and Nhung T. H. Nguyen[‡*]
[†]Human Language Technology Dept.
Institute for Infocomm Research (I^{2}R), A*STAR, Singapore
{cdvhoang, aaiti}@i2r.a-star.edu.sg
[‡]School of Information Science
Japan Advanced Institute of Science and Technology (JAIST), Japan
nthnhung@jaist.ac.jp

Abstract

Interactive or Incremental Statistical Machine Translation (IMT) aims to provide a mechanism that allows the statistical models involved in the translation process to be incrementally updated and improved. The source of knowledge normally comes from users who either post-edit the entire translation or just provide the translations for wrongly translated domain-specific terminologies. Most of the existing work on IMT uses batch learning paradigm which does not allow translation systems to make use of the new input instantaneously. We introduce an adaptive MT framework with a Rule Definition Language (RDL) for users to amend MT results through translation rules or patterns. Experimental results show that our system acknowledges user feedback via RDL which improves the translations of the baseline system on three test sets for Vietnamese to English translation.

1 Introduction

In current Statistical Machine Translation (SMT) framework, users are often seen as passive contributors to MT performance. Even if there is a collaboration between the users and the system, it is carried out in a batch learning paradigm (Ortiz-Martinez et al., 2010), where the training of the SMT system and the collaborative process are carried out in different stages. To increase the productivity of the whole translation process, one has to incorporate human correction activities within the translation process. Barrachina et al. (2009) proposed an iterative process in which the translator activity is used by the system to compute its best (or n-best) translation suffix hypotheses to complete the prefix. Ortiz-Martinez et al. (2011) proposed an IMT framework that includes stochastic error-correction models in its statistical formalization to address the prefix coverage problems in Barrachina et al. (2009). Gonzalez-Rubio et al. (2013) proposed a similar approach with a specific error-correction model based on a statistical interpretation of the Levenshtein distance (Levenshtein, 1966). On the other hand, Ortiz-Martinez et al. (2010) presented an IMT system that is able to learn from user feedback by incrementally updating the statistical models used by the system. The key aspect of this proposed system is the use of HMM-based alignment models trained by an incremental EM algorithm.

Here, we present a system similar to Ortiz-Martinez et al. (2010). Instead of updating the translation model given a new sentence pair, we provide a framework for users to describe translation rules using a Rule Definition Language (RDL). Our RDL borrows the concept of the rule-based method that allows users to control the translation output by writing rules using their linguistic and domain knowledge. Although statistical methods pre-dominate the machine translation research currently, rule-based methods are still promising in improving the translation quality. This approach is especially useful for low resource languages where large training corpus is not always available. The advantage of rule-based methods is that they can well handle particular linguistic phenomena which are peculiar to languages and domains. For example, the TCH MT system at IWSLT 2008 (Wang et al., 2008) used dictionary and hand-crafted rules (e.g. regular expression) to process NEs. Their experiments showed that handling NE separately (e.g., person name, location name, date, time, digit) results in translation quality improvement.

In this paper, we present an adaptive and in-

*Work done during an internship at I^{2}R, A*STAR.

Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 73–78,
Baltimore, Maryland USA, June 23-24, 2014.

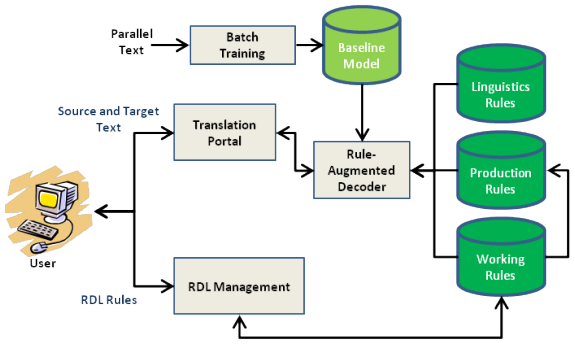

Figure 1: The proposed rule-augmented SMT framework.

teractive MT system that allows users to correct the translation and integrate the adaptation into the next translation cycle. Our experiments show that the system is specifically effective in handling translation errors related to out of vocabulary words (OOVs), language expressions, name entities (NEs), abbreviations, terminologies, idioms, etc. which cannot be easily addressed in the absence of in-domain parallel data.

2 System Overview

Figure 1 shows the translation and interactive process of our system. The system is trained with a batch of parallel texts to create a baseline model. Users improve the translation by adding RDL rules to change or correct the unsatisfactory translation. New RDL rules are tested in a working environment before uploading to the production environment where they would be used by subsequent translation requests.

In our system, RDL Management checks, validates and indexes the translation rules. The Rule-Augmented Decoder has two components: (1) the RDL Matcher to find applicable RDL rules for a given source text to create dynamic translation hypotheses; and (2) the Augmented Decoder to produce the final consensus translation using both dynamic hypotheses and static hypotheses from the baseline model.

3 Rule Definition Language (RDL)

The Rule Definition Language (RDL) comprises a RDL grammar, a RDL parser and a RDL matching algorithm.

3.1 RDL Grammar

Our RDL grammar is represented with a Backus-Naur Form (BNF)s syntax. The major feature of

Node Type	Description
Token	Any string of characters in the defined basic processing unit of the language.
String	A constant string of characters.
Identifier	A term represents a pre-defined role (e.g. integer, date, sequence, . . .).
Meta-node	A term executes a specific function (e.g. casing, selection/option, connection).
Context cue	A term describes source context's existence.
Function	A term executes a pre-defined task.

Table 1: A brief description of RDL nodes.

```
rule DATE5_1 //rule info
{
#s [!"Vào"] (!"Năm tài khoá"|!" Năm tài chính") @Num //source
#c ~inRangeOf(@Num, "1000", "9999") //condition
#t [!"Vào"] -> ["in"] //target
#r [["in"] "the fiscal year" @Num] //reordering
#cf false //user confidence
}
```

Figure 2: An Example of RDL Rule.

RDL grammar is the support of pre-defined identifiers and meta-operators which go beyond the normal framework of regular expression. We also included a set of pre-defined functions to further constraint the application and realization of the rules. This framework allows us to incorporate semantic information into the rule definition and derive translation hypotheses using both semantic and lexical information. A RDL rule is identified by a unique rule ID and five constituents, including **S**ource pattern, rule **C**ondition, **T**arget translation, **R**eordering rule and user **C**on**F**idence. The source pattern and target translation can be constructed using different combination of node types as described in Table 1. The rules can be further conditioned by using some pre-defined functions and the system allows users to reorder the translation of the target node. Figure 2 gives an example of a RDL rule where identifier @Num is used.

3.2 RDL Parsing and Indexing

The RDL Parser checks the syntax of the rules before indexing and storing them into the rule database. We utilize the compiler generator (WoB et al., 2003) to generate a RDL template parser and then embed all semantic parsing components into the template to form our RDL Parser.

As rule matching is performed during translation, searching of the relevant rules have to be very fast and efficient. We employed the modified version of an inverted index scheme (Zobel and Moffat, 2006) for our rule indexing. The algorithm is

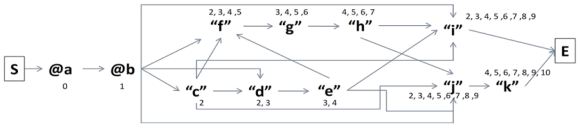

Figure 3: A linked item chain for a rule source (@a @b [c] [“d e”] [“f g h”] (“i” | “j k”)).

represented in Algorithm 1.

Data: ruleID & srcPatn
Result: idxTbl
// To build data structure − Forward Step
doForward(srcPatn, linkedItmChain);
// To create index table − Backward Step
doBackward(linkedItmChain, ruleID, idxTbl);
Algorithm 1: Algorithm for RDL rule indexing.

The main idea of the rule indexing algorithm is to index all string-based nodes in the source pattern of the RDL rule. Each node is represented using 3-tuple. They are ruleID, number of nodes in source pattern and all plausible positions of the node during rule matching. The indexing is carried out via a Forward Step and Backward Step. The Forward Step builds a linked item chain which traverses all possible position transitions from one node to another as illustrated in Figure 3. Note that S and E are the Start and End Node. The link indicates the order of transition from a node to another. The numbers refer to the possible positions of an item in source. The Backward Step starts at the end of the source pattern; traverses back the link to index each node using the 3-tuple constructed in the Forward Step. This data structure allows us to retrieve, add or update RDL rules efficiently and incrementally without re-indexing.

3.3 RDL Matching Algorithm

Each word in the source string will be matched against the index table to retrieve relevant RDL rules during decoding. The aim is to retrieve all RDL rules in which the word is used as part of the context in the source pattern. We sort all the rules based on the word positions recorded during indexing, match their source patterns against the input string within the given span, check the conditions and generate the hypotheses if the rules fulfill all the constraints.

4 Rule-Augmented Decoder

The rule-augmented decoder integrates the dynamic hypotheses generated during rule matching with the baseline hypotheses during decoding. Given a sentence *f* from a source language *F*, the fundamental equation of SMT (Brown et al., 1993) to translate it into a target sentence *e* of a target language *E* is stated in Equation 1.

$$\begin{aligned} e_{best} &= argmax_e P_r(e|f) \\ &= argmax_e P_r(f|e) P_r(e) \\ &= argmax_e \sum_{n=1}^{N} \lambda_n h_n(e, f) \end{aligned} \tag{1}$$

Here, $P_r(f|e)$ is approximated by a translation model that represents the correlation between the source and the target sentence and $P_r(e)$ is approximated by a language model presenting the well-formedness of the candidate translation e. Most of the SMT systems follow a log-linear approach (Och and Ney, 2002), where direct modelling of the posterior probability $P_r(f|e)$ of Equation 1 is used. The decoder searches for the best translation given a set of model $h_m(e, f)$ by maximizing the log-linear feature score (Och and Ney, 2004) as in Equation 1.

For each hypothesis generated by the RDL rule, an appropriate feature vector score is needed to ensure that it will not disturb the probability distribution of each model and contributes to hypothesis selection process of SMT decoder.

4.1 Model Score Estimation

The aim of the RDL implementation is to address the translation of language-specific expressions (such as date-time, number, title, etc.) and domain-specific terminologies. Sometimes, translation rules and bilingual phrases can be easily observed and obtained from experienced translators or linguists. However, it is difficult to estimate the probability of the RDL rules manually to reflect the correct word or phrase distribution in real data. Many approaches have been proposed to solve the OOV problem and estimate word translation probabilities without using parallel data. Koehn et al. (2000) estimated word translation probabilities from unrelated monolingual corpora using the EM algorithm. Habash et al. (2008) presented different techniques to extend the phrase table for online handling of OOV. In their approach, the extended phrases are added to the baseline phrase

table with a default weight. Arora et al. (2008) extended the phrase table by adding new phrase translations for all source language words that do not have a single-word entry in the original phrase-table, but appear in the context of larger phrases. They adjusted the probabilities of each entry in the extended phase table.

We performed different experiments to estimate the lexical translation feature vector for each dynamic hypothesis generated by our RDL rules. We obtain the best performance by estimating the feature vector score using the baseline phrase table through context approximation. For each hypothesis generated by the RDL rule, we retrieve entries from the phrase table which have at least one similar word with the source of the generated hypothesis. We sort the entries based on the similarities between the generated and retrieved hypotheses using both source and target phrase. The medium score of the sorted list is assigned to the generated hypothesis.

5 System Features

The main features of our system are (1) the flexibilities provided to the user to create different levels of translation rules, from simple one-to-one bilingual phrases to complex generalization rules for capturing the translation of specific linguistic phenomena; and (2) the ability to validate and manage translation rules online and incrementally.

5.1 RDL Rule Management

Our system framework is language independent and has been implemented on a Vietnamese to English translation project. Figure 4 shows the RDL Management Screen where a user can add, modify or delete a translation rule using RDL. A RDL rule can be created using nodes. Each node can be defined using string or system predefined meta-identifiers with or without meta-operators as described in Table 1. Based on the node type selected by the user, the system further restricts the user to appropriate conditions and translation functions. The user can define the order of the translation output of each node and at the same time, inform the system whether to use a specific RDL exclusively during decoding, in which any phrases from the baseline phrase table overlapping with that span will be ignored[1]. The system also provides an editor for expert users to code the rules using the RDL controlled language. Each rule is validated by the RDL parser (discussed in section 3.2), which will display errors or warning messages when an invalid syntax is encountered.

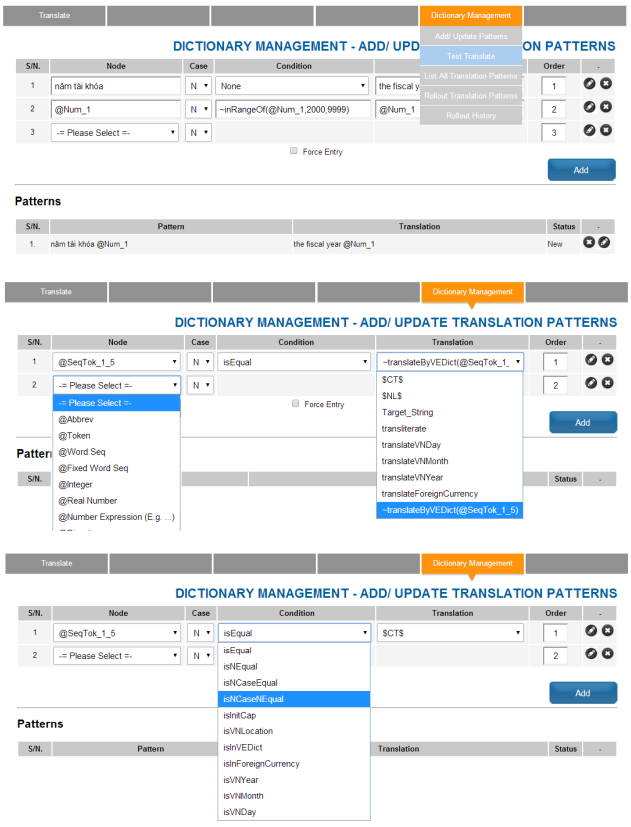

Figure 4: RDL Management screen with identifiers & meta-functions supported.

5.2 RDL Rule Validation

Our decoder manages two types of phrase table. One is the static phrase-table obtained through the SMT training in parallel texts; the other is the dynamic table that comprises of the hypotheses generated on-the-fly during RDL rule matching. To ensure only fully tested rules are used in the production environment, the system supports two types of dynamic phrase table. The working phrase-table holds the latest updates made by the users. The users can test the translation with these latest modifications using a specific translation protocol. When users are satisfied with these modifications, they can perform an operation to upload the RDL rules to the production phrase-table, where the RDLs are used for all translation

[1]Similar to Moses XML markup exclusive feature http://www.statmt.org/moses/?n=Moses.AdvancedFeatures#ntoc9

Named Entity Category	Number of Rules
Date-time	120
Measurement	92
Title	13
Designation	12
Number	19
Terminology	178
Location	13
Organization	48
Total	**495**

Table 2: Statistics of created RDL rules for Vietnamese-to-English NE Translation.

requests. Uploaded rules can be deleted, modified and tested again in the working environment before updated to the production environment. Figure 5b and Figure 5c show the differences in translation output before and after applied the RDL rule in Figure 5a.

6 A Case Study for Vietnamese–English Translation

We performed an experiment using the proposed RDL framework for a Vietnamese to English translation system. As named entity (NE) contributes to most of the OOV occurrences and impacts the system performance for out-of-domain test data in our system, we studied the NE usage in a large Vietnamese monolingual corpus comprising 50M words to extract RDL rules. We created RDL rules for 8 popular NE types including title, designation, date-time, measurement, location, organization, number and terminology. We made use of a list of anchor words for each NE category and compiled our RDL rules based on these anchor words. As a result, we compiled a total of 495 rules for 8 categories and it took about 3 months for the rule creation. Table 2 shows the coverage of our compiled rules.

6.1 Experiment & Results

Our experiments were performed on a training set of about 875K parallel sentences extracted from web news and revised by native linguists over 2 years. The corpus has 401K and 225K unique English and Vietnamese tokens. We developed 1008 and 2548 parallel sentences, each with 4 references, for development and testing, respectively. All the reference sentences are created and revised by different native linguists at different times. We also trained a very large English language model using data from Gigaword, Europarl and English

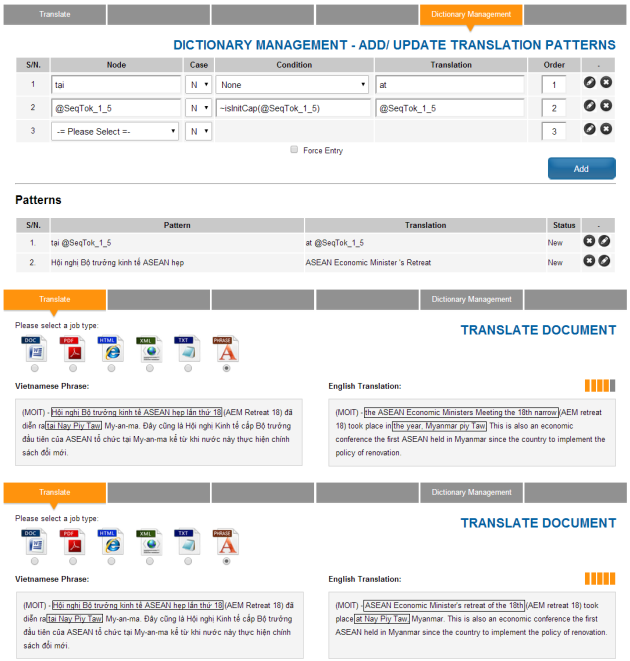

Figure 5: Translation Demo with RDL rules.

Data Set	nS	nT	nMR
TrainFull (VN)	875,579	28,251,775	627,125
TrainFull (EN)	875,579	20,191,526	-
Test1 (VN)	1009	34,717	737
Test1 (4 refs) (EN)	1009	≈25,713	-
Test2 (VN)	1033	29,546	603
Test2 (4 refs) (EN)	1033	≈22,717	-
Test3 (VN)	506	16,817	344
Test3 (4 refs) (EN)	506	≈12,601	-
Dev (VN)	1008	34,803	-
Dev (4 refs) (EN)	1008	≈25,631	-

Table 3: Statistics of Vietnamese-to-English parallel data. nS, nT, and nMR are number of sentence pairs and tokens, and count of matched rules, respectively.

web texts of Vietnamese authors to validate the impact of RDL rules on large-scale and domain-rich corpus. The experimental results show that created RDL rules improve the translation performance on all 3 test sets. Table 3 and Table 4 show respective data statistics and results of our evaluation. More specifically, the BLEU scores increase 3%, 3.6% and 1.4% on the three sets, respectively.

7 Conclusion

We have presented a system that provides a control language (Kuhn, 2013) specialized for MT for users to create translation rules. Our RDL differs from Moses's XML mark-up in that it offers fea-

Data Set	System	BLEU	NIST	METEOR
Set 1	Baseline	39.21	9.2323	37.81
	+RDL (all)	39.51	9.2658	37.98
Set 2	Baseline	40.25	9.5174	38.24
	+RDL (all)	40.61	9.6092	38.84
Set 3	Baseline	36.77	8.6953	37.65
	+RDL (all)	36.91	8.7062	37.69

Table 4: Experimental results with RDL rules.

tures that go beyond the popular regular expression framework. Without restricting the mark-up on the source text, we allow multiple translations to be specified for the same span or overlapping span.

Our experimental results show that RDL rules improve the overall performance of the Vietnamese-to-English translation system. The framework will be tested for other language pairs (e.g. Chinese-to-English, Malay-to-English) in the near future. We also plan to explore advanced methods to identify and score *"good"* dynamic hypotheses on-the-fly and integrate them into current SMT translation system (Simard and Foster, 2013).

Acknowledgments

We would like to thank the reviewers of the paper for their helpful comments.

References

Paul M. Sumita E. Arora, K. 2008. Translation of unknown words in phrase-based statistical machine translation for languages of rich morphology. In *In Proceedings of the Workshop on Spoken Language Technologies for Under-Resourced Languages, SLTU 2008*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Comput. Linguist.*, 35(1):3–28, March.

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

Jesús González-Rubio, Daniel Ortíz-Martinez, José-Miguel Benedí, and Francisco Casacuberta. 2013. Interactive machine translation using hierarchical translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 244–254, Seattle, Washington, USA, October.

Nizar Habash. 2008. Four techniques for online handling of out-of-vocabulary words in arabic-english statistical machine translation. In *Proceedings of ACL: Short Papers*, HLT-Short '08, pages 57–60, Stroudsburg, PA, USA.

Philipp Koehn and Kevin Knight. 2000. Estimating word translation probabilities from unrelated monolingual corpora using the em algorithm. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 711–715. AAAI Press.

Tobias Kuhn. 2013. A survey and classification of controlled natural languages. *Computational Linguistics*.

VI Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *In Proceedings of ACL*, pages 295–302, Stroudsburg, PA, USA.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30(4):417–449, December.

Daniel Ortiz-Martinez, Ismael Garcia-Varea, and Francisco Casacuberta. 2010. Online learning for interactive statistical machine translation. In *In Proceedings of NAACL*, HLT '10, pages 546–554, Stroudsburg, PA, USA.

Daniel Ortiz-Martínez, Luis A. Leiva, Vicent Alabau, Ismael García-Varea, and Francisco Casacuberta. 2011. An interactive machine translation system with online learning. In *In Proceedings of ACL: Systems Demonstrations*, HLT '11, pages 68–73, Stroudsburg, PA, USA.

Michel Simard and George Foster. 2013. Pepr: Post-edit propagation using phrase-based statistical machine translation. *Proceedings of the XIV Machine Translation Summit*, pages 191–198.

Haifeng Wang, Hua Wu, Xiaoguang Hu, Zhanyi Liu, Jianfeng Li, Dengjun Ren, and Zhengyu Niu. 2008. The tch machine translation system for iwslt 2008. In *In Proceedings of IWSLT 2008, Hawaii, USA*.

Albrecht WoB, Markus Loberbauer, and Hanspeter Mossenbock. 2003. Ll(1) conflict resolution in a recursive descent compiler generator. In *Modular Programming Languages*, volume 2789 of *Lecture Notes in Computer Science*, pages 192–201.

Justin Zobel and Alistair Moffat. 2006. Inverted files for text search engines. *ACM Comput. Surv.*, 38, July.