A model based transformation paradigm for cross-language collaborations

Kunmei Wen a,⁎, Suo Tan b, Jie Wang a, Ruixuan Li a, Yuan Gao a

a School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
b Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada H3G 1W8

ARTICLE INFO

Article history:
Received 8 March 2012
Received in revised form 11 October 2012
Accepted 22 October 2012
Available online xxxx

Keywords:
Cross-language transformation
Online collaboration
Recursive object model
Model-based paradigm
CLT-ROM

ABSTRACT

Online collaboration is a big challenge in the field of international product development in a cross-language environment. It serves two purposes: cross-language translation and design requirement clarification. Though many approaches and tools are developed for each of the purposes, not a solution serves both of them well. Especially, the traditional statistical methods for cross-language translation cannot preserve the whole semantic information, which intend to incur misunderstanding and ineffective collaboration. This results in potential problems in clarifying the design requirements. In this paper, we proposed a method to online collaboration, named Cross-Language Transformation based on Recursive Object Model (CLT-ROM). The proposed method consists of two steps. Firstly, a natural language sentence is transformed into a source ROM diagram. Secondly, a corresponding target ROM diagram is generated by a transformation algorithm. The proposed method is a model-based communication tool which facilitates collaborations. Since the ROM has been proven effective in requirements clarification, some examples are given to illustrate that the CLT-ROM has a good capability of semantic preserving in requirement engineering for product development.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

With the gradual integration of the world economy, communication and cooperation are playing an increasing role for enterprisers in achieving their business goals effectively. For those enterprisers who expand their business into global markets through international collaborations in product development, cross-language and cross-cultural challenges emerge. These challenges may prevent the business partners from understanding each other, and result in ineffective collaborations that harm their business. Though many researchers have put much effort into finding solutions [1–3], not a solution satisfied the common needs for international engineering collaborations. Large enterprisers are able to face the challenges since they have enough resources to deal with. But this may not be the case for small and medium enterprisers (SMEs) in developing and underdeveloped countries.

Let us consider such a scenario: an English-speaking company and a Chinese-speaking company are collaborating on rapid product development in multi-disciplinary design situations. Employees from both companies are having difficulties in expressing themselves clearly in a language other than their mother tongues. For such a case, the very basic activities in product development that encounter cross-language barrier can be extracted as real-time tasks (for example, video and telephone conference, instant message, face-to-face online discussion) and static tasks (such as document translation and email correspondence). Obviously, once a solution to resolving the real-time tasks is found, static tasks can be resolved without a doubt. Furthermore, if we could use conference calls over face-to-face discussion, we could save a large amount of money. And thus, the solution must be efficient for the collaboration. If other real-time tasks apart from face-to-face online discussion are renamed as online collaboration, the problems in cross-language collaborations can be simplified as a single critical one – online collaboration.

To break the barrier, enterprisers have to hiring experts as interpreters and/or purchase software kits (language tool). However, for interpreters who well know both languages may do not have domain specific knowledge about the design concepts. Therefore, they cannot perform documents translations and real-time interpretations. Even though perfect interpreters are available, they are always too expensive for SMEs to hire. For software kits, they are less expensive compared to experts in the long run. Unfortunately, current software kits still offer poor translations in engineering applications. A substantial demand of a more reliable solution in engineering applications is increasing for SMEs.

Before giving a solution, we must understand the purpose of online collaboration for cross-language collaborations in product development. Based on the previous analysis, online collaboration serves two objectives: cross-language translation and design requirements clarification. The first purpose is to make a translated sentence readable in semantics. The other one is to ensure the
translated sentences reasonable and meaningful for communication, especially for engineering applications. They are in two different levels. Although human translators are still widely used in cross-language collaborations, in this paper, human translation is not considered.

Computational linguistics is an interdisciplinary field dealing with natural language modeling from a computational perspective. Natural Language Processing (NLP) is also an interdisciplinary field of computer science, artificial intelligence, and linguistics, which focuses on the interactions between computers and human natural languages. Sometimes, computational linguistics and NLP can be treated as a same concept. Machine Translation (MT) is a sub-field of computational linguistics or NLP that investigates the automatic translation of text or speech from one natural language to another. Their common goal is to enhance human–machine communication in natural languages. Language translation, also referred as MT [4], is the most commonly used method to implement cross-language collaborations by transforming a source sentence to a target sentence, for example, transforming an English sentence to a Chinese sentence. However, language translation maintains a challenge despite of the efforts in MT. Semantic integrity is a basic standard for cross-language engineering applications. If the semantic information included in natural languages is lost, the translation results will be incorrect. As a result, the design product cannot satisfy the users’ requirements. Whether designers can clearly understand customers’ requirements or not largely determines the practicability and efficiency of the design results [5]. Therefore, the semantics should be preserved in the process of cross-language translation. There are two important features of semantics: constant and imparity.

Constant: The linguistic relation between words of a sentence tends to be constant. Imparity: The semantics that different words carry are imparity.

Traditional language translation methods cannot resolve the semantic preserving problem, in other words, the constant and imparity semantics are lost. The ambiguity is also unavoidable. Since the traditional methods cannot well preserve the semantics in translation, it motivates us to propose a different paradigm for cross-language collaborations. We attempted to preserve the semantics included in natural language by cross-language model transformation.

In this paper, we propose a method called Cross-Language Transformation based on Recursive Object Model (CLT-ROM) for the cross-language online collaboration issue, in order to improve the effectiveness and efficiency of global collaborations. The contributions are as follows: (1) a model-based transformation paradigm is proposed for cross-language collaborations. (2) A prototype of the model-based transformation method is implemented. (3) Examples are provided to illustrate the feasibility of the method.

The rest of the paper is organized as follows. Section 2 interprets the related work. In Section 3, we present a brief introduction to the ROM. The CLT-ROM method is proposed in Section 4. Case study and experiments are provided in Section 5 to illustrate the effectiveness of the proposed method. Finally, we give the conclusions and the future work in Section 6.

2. Problem analysis and related work

A natural language is a typical tool used for communication on daily basis. However, it is not the only option. As shown in Fig. 1, diverse ways are used to describe the communication universe. For instance, symbol and mathematical expressions are neat in science. A model language, Unified Modeling Language (UML) for example, is used for structural representation of objects in software engineering. Apparently, communication can be natural-language-independent. This inspired us that if an appropriate model with primitive symbols is generated to carry semantics of natural language and to represent the relationships of language elements intuitively, perhaps a novel paradigm could be proposed for cross-language collaboration. The Recursive Object Model (ROM) [6] is such a graph-based approach used to model the syntactic and semantic relations between words in a sentence. It can preserve the semantics of natural languages. The constant feature of the linguistic relation between the words can be kept by the predefined elements. Meanwhile, the different relations defined in the ROM can be well used to distinguish the imparity semantics according to the restricted semantics and non-restricted semantics. Therefore, the correct model of a sentence is unique. And ambiguity issue is reduced or eliminated. A detailed introduction to the ROM is given in Section 3.

As shown in Fig. 2, language translation methods translate a source sentence into a target sentence directly. In comparison, model-based transformation method, shown in Fig. 3, transforms a sentence into a source model first. And then the source model is transformed into a target model. The main differences between the traditional language translation method and our model-based transformation method can be summarized as follows: firstly, traditional translation is based on statistical methods by translating one sentence into another language. Our transformation method transforms one sentence into its ROM diagrams in different languages. It provides a new paradigm. Secondly, since the CLT-ROM method is based on the ROM, it can preserve the semantics, while the traditional method may lose some, especially the relations among words.

The aim of cross-language translation is simple: given a source sentence, its equivalent target sentence is to be generated. However, implementation of language translation is complex as inherent ambiguity. In other words, a single word has different meanings with different contexts and idioms. Therefore, without considering the differences between the source language and the target language, translating a source sentence merely into a word sequence of a target language cannot implement a good translation. The word sequence must be reordered before generating the target language sentence to ensure a high quality translation. As a matter of fact, reordering the translated words to fit the logic of a target language is one of the most challenging problems in MT.

When solving the translation problems, language translation is divided into the two categories: Rule-Based Machine Translation (RBMT) and Corpus-Based Machine Translation (CBMT) [12]. The main ideas of RBMT are based on linking the structure of a given input sentence with the structure of a demanded output sentence, necessarily preserving their unique meaning. The RBMT is a large-scale and time-consuming method. It is hard to improve the translation performance because adding a new rule into a large-scale rule set is difficult to implement. Some linguistic information still needs to be set manually, resulting in a huge and changeful rule base. It is too hard to guarantee the consistency of the rules in...
big systems. CBMT involved to Example-Based Machine Translation (EBMT) and Statistical Machine Translation (SMT). Most of the efforts were made on the SMT methods as the EBMT methods [13] build a base to store translation examples first.

The SMT were evolved in traditional word-and-phrase-based model and syntax model. Brown et al. [14] proposed a word-based SMT method by using a noisy channel model. The method was an important work for the SMT methods. The model was designed to model the lexical dependencies between the words of sentences. However, this method did not resolve the reordering problem. The phrase-based models [15] were introduced to remedy the reordering model. Target language strings were broken into phrases which are independently translated as blocks. These translated phrases were reordered to produce the source sentence according to a distortion probability. Talbot et al. [16] designed a lightweight evaluation framework for language translation reordering. Reordering models involved some different models, such as fundamental distance-based distortion model [16,17], flat reordering model [18,19], lexicalized reordering model [20], hierarchical phrased model [21] and maximum entropy-based phrase reordering model [22]. These methods cannot carry out the complex reordering operations including long distance dependencies and variable contexts. To obtain a global reordering result, the syntax-based SMT methods were developed to capture the syntactic knowledge by simply swapping the children nodes of a parse tree [23]. These methods totally depend on the syntax structure to complete translation [24–26]. These syntax-based SMT methods partly resolve the local reordering problem. However, the reordering results cannot be consistent with syntactic structures. Furthermore, a syntax-based SMT model is more complex than a phrase-based SMT model.

Overall, the methods mentioned above brought some improvements. However, in order to provide a good translation for a source sentence, the existing methods must balance the translated effect and the reordering consequence. It brings a problem: these methods cannot fully preserve the original semantics included in natural language. Based on different classification methods, there are several kinds of semantic information, such as humor semantics [27] and ontology semantics [28]. Correct and complete semantic information plays a significant role in cross-language understanding. Focusing on the constant feature of the linguistic relation between words and the imparity feature of semantics, we proposed the CLT-ROM method. The differences between the traditional methods and the proposed transformation method can be summarized in Table 1, for an English to Chinese translation case.

3. Introduction to the ROM

The ROM [6] is a graphic model that carries the semantic information included in natural language. It is a tool used to analyze natural languages, especially to extract major ontological words. It uses two primitives, object and relation, to capture the linguistic meaning and relation of the words. These primitives are proven sufficient for technical English through enumeration [6]. Every word and phrase in sentences can be treated as an object and a compound object, respectively. Some applications have demonstrated the usefulness of the ROM in engineering design [8,10,11,29,30].

We use the notation R(A, B) to denote the representations of ROM diagrams, where R denotes the relation between two objects A and B. Fig. 4 illustrates the different representations of R(A, B). From the left to the right in Fig. 4, they are constraint, connection, and predicate respectively.

4. The proposed method – CLT-ROM

The proposed method, CLT-ROM, is for cross-language collaborations in this paper. Our focus is on the transformation from English sentences to Chinese ROMs. The inverse transformation can be implemented by using this method under different rules. Fig. 5
illustrates the main process of the CLT-ROM method. Firstly, a source sentence is transformed to a Source ROM (S-ROM) by using mapping rules between the representations of grammar parser and the ROM relations. The mapping rules are discussed later. Then, a Target ROM (T-ROM) is generated by the CLT-ROM algorithm addressed in Section 4.2.

To facilitate the discussion, some notations and concepts are defined in Section 4.1, and an example sentence is given to show the process of transformation: “Design a tool for fixing brake linings onto brake shoes for internal drum brakes.”

### 4.1. Concepts and notations

Syntax is the study of the principles and processes by which sentences are constructed. Semantics is the study of meaning. It focuses on the relations between words or phrases. The lexical of a language is its vocabulary. Therefore, the definition of semantics can explain better through the concepts of restricted semantics and non-restricted semantics. Restricted semantics indicate that there are some relations between objects. On the contrary, non-restricted semantics indicate that there is no any relation between objects.

We assume that there are n words in a sentence $S = W_1, W_2 \cdots W_n$. If $S$ is presented by a ROM diagram with n objects, $S_0 = \{O_1, O_2, \ldots, O_n\}$ is able to denote the sentence. The object $O_j$ is corresponding to the word $W_j$. In addition, there are m ROM relations denoted as $S_0 = \{R_1, R_2, \ldots, R_m\}$. $R_i$ can be expressed as $R_i(O_k, O_j)$, where $O_k, O_j \in S_0$ and $1 \leq k < n, 1 \leq j \leq n$ and $k \neq j$.

The three relations defined in the ROM are denoted as $R = \{Predicate\}, \{Constraint\}$, and $\{Connection\}$. They imply different semantics for any $R_i(O_k, O_j)$:

1. If $R_i(O_k, O_j) = Predicate$, it indicates that the object $O_k$ restricts the semantic of the object $O_j$. Meanwhile, $O_j$ restricts the semantic of $O_k$ as well. The Predicate relation is a bidirectional restricted semantic relation.

2. If $R_i(O_k, O_j) = Constraint$, it indicates that the object $O_k$ restricts the semantic of the object $O_j$ whereas $O_j$ does not restrict the semantic of $O_k$. The Constraint relation is a non-bidirectional restricted semantic relation.

3. If $R_i(O_k, O_j) = Connection$, it indicates that the status of the objects $O_k$ and $O_j$ is independent. There is no any restricted semantic between the two objects with connection relation.

Therefore, the objects can be categorized into two types: restricted semantic objects and non-restricted semantic objects. For an object $O_k$, at least, has one object restricting its semantics, it is a restricted semantic object. It can be formally represented as:

For any $O_k \in S_0$,

if $\exists R_i(O_k, O_j) = \{Predicate\}$ or $\exists R_i(O_j, O_k) = \{Predicate\}$, then $O_k \in R_{object}$

On the contrary, if an object $O_k$ has no other object restricting its semantics, it is a non-restricted semantic object. It is formally represented as:

For any $O_k \in S_0$,

if $O_k \in S_0 - R_{object}$ then $O_k \in NR_{object}$

Other than the these two objects, there is another special type of objects – compound objects. It is helpful to reduce the complexity of cross-language transformation. A definition is given to the compound objects as: A compound object is an object which includes two or more objects in it. Compound objects always consist of phrases, such as noun phrases, verb phrases, verbal phrases, and gerund phrases. The structure of a ROM diagram is more clear if compound objects are used. In the CLT-ROM method, determining compound objects is one of the important steps.

### 4.2. CLT-ROM algorithm

The algorithm for the CLT-ROM is shown in Algorithm 1. The input of the CLT-ROM algorithm is an English source sentence, whereas the output is a Chinese Target ROM (T-ROM) diagram. The Chinese T-ROM can be obtained in four steps. Step 1, the S-ROM is generated for the English source sentence $S$. Some preparations are necessary in step 1. We need to record some additional information for the objects of the S-ROM to classify these objects. In step 2, the compound objects are determined. Step 3, the objects of the S-ROM are transformed into the target objects. For the restricted semantic objects and non-restricted semantic objects, different transformational operations are performed. The structure of the S-ROM is updated and the Chinese T-ROM is obtained in step 4.
Fig. 5. Main process of the CLT-ROM method.

Fig. 6. How generating S-ROM.

4.2.1. Generation of S-ROM and preparation

Fig. 6 illustrates how to generate an S-ROM from a source sentence. Where SD (Stanford typed Dependencies) [31] represents sentences by using a grammar parser. It acts as a bridge between a natural language sentence and its ROM diagram. The S-ROM is generated by mapping an SD representation into a ROM representation. The mapping rules preserve the syntax logic relation between two objects. In order to improve the quality of mapping results, specified rules are developed to revise some special SD representations. Table 3 gives some mapping rules. The amod and the conj are mapped into ROM constraint relations. The cc and the csubj are mapped into ROM connection relations. The cop and the csubj are mapped into ROM predicate relations. The complete mapping rules are determined once a source language is selected.

The algorithm for S-ROM generation and preparation is shown in Algorithm 2. Once an S-ROM is generated (line 1), the ROM representation set SROM and the object set SO (line 2) are obtained. By analyzing the objects in the S-ROM, some preparations are then completed, e.g., classifying parts of speech (POS), differentiating tense (present progressive, present perfect, or pluperfect), identifying voice (active or passive), and determining morphology (singular or plural). During the process of preparation, additional information (AI) is also obtained (lines 5 and 6). The information will be utilized in the later object transformation. Object lemmatization is also needed for the object set SO (line 7), which is critical in getting the candidate transformation. Most lemmas in the dictionary are stored in terms of original morphology.

Algorithm 2. Generating S-ROM and preparation

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Generating the S-ROM</td>
</tr>
<tr>
<td>2</td>
<td>Initial SROM, SO</td>
</tr>
<tr>
<td>3</td>
<td>for all O ∈ SO do</td>
</tr>
<tr>
<td>4</td>
<td>O = Lemmatize(O)</td>
</tr>
<tr>
<td>5</td>
<td>end for</td>
</tr>
<tr>
<td>9</td>
<td>for all R(Oi, Oj) ∈ SROM do</td>
</tr>
<tr>
<td>10</td>
<td>if R = Predicate then</td>
</tr>
<tr>
<td>11</td>
<td>AOC[Io] = AOC[Io] + 1</td>
</tr>
<tr>
<td>12</td>
<td>AOC[Io] = AOC[Io] + 1</td>
</tr>
<tr>
<td>13</td>
<td>end if</td>
</tr>
<tr>
<td>14</td>
<td>if R = Constraint then</td>
</tr>
<tr>
<td>15</td>
<td>AOC[Io] = AOC[Io] + 1</td>
</tr>
<tr>
<td>16</td>
<td>end if</td>
</tr>
<tr>
<td>18</td>
<td>for all O ∈ SO do</td>
</tr>
<tr>
<td>19</td>
<td>if AOC[Ij] = 0 or AI(O).POS = preposition then</td>
</tr>
<tr>
<td>20</td>
<td>Sn = Sn ∪ O</td>
</tr>
<tr>
<td>21</td>
<td>else</td>
</tr>
<tr>
<td>22</td>
<td>St = St ∪ O</td>
</tr>
<tr>
<td>23</td>
<td>end if</td>
</tr>
<tr>
<td>24</td>
<td>end for</td>
</tr>
</tbody>
</table>

All of the elements in the set SROM are traversed. The restricted semantic relation and non-semantic relation are obtained from SROM. Concurrently, the number of restricted semantic relations is recorded for each object (lines 9–17). According to the definition in Section 3.1, classifying parts of speech (POS), as one kind of AI information, is useful to categorize the objects into R_object or NR_object (line 18–23). We used an AOC (the count of an object) array to store the pairs (O, count). The O is corresponding to the object whereas count is corresponding to the number of the object’s restricted semantics. The elements of AOC are resorted in a descending order according to the value of count. To illustrated, the generated S-ROM for the example sentence is shown in Fig. 7.

4.2.2. Determination of compound objects

As aforementioned, the complexity of cross-language transformation can be reduced by using compound objects. Following principles are followed to determine compound objects.

Table 3
A part of the mapping rules.

<table>
<thead>
<tr>
<th>SD</th>
<th>amod</th>
<th>cc</th>
<th>conj</th>
<th>cop</th>
<th>csubj</th>
<th>det</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>Constraint</td>
<td>Connection</td>
<td>Connection</td>
<td>Predicate</td>
<td>Predicate</td>
<td>Constraint</td>
</tr>
</tbody>
</table>
Firstly, all of the objects in Algorithm 3. Determining compound objects for the example sentence. The variable $x$ indicates the position of is seen as the last object of a compound object (line 3). The variable $y$ records the position of one of the case matches the dictionary phrase we record it. If the first object of the compound object, and $y$ records the position of $Oe$ which has the least ROM relation connections with $Oe$ is regarded as the first object of the compound object, and $y$ records the position of $StartP(Oe)$ (line 5). A recursion is used to identify the first object of a compound object. Then the current objects generated by composing all the objects between $Oe$ and $Oe$ is supposed to be a compound object. For every compound object $Oe = \{O_{1}, O_{2}, \ldots, O_{n}\}$, $O_{i}(i = 1, 2, \ldots, n)$ is a simple object and its stem is stem ($O_{i}$). We traverse all possibilities that $O_{i}$ can reach by getHead ($O_{i}$) or its stem ($O_{i}$). Once one of the case matches the dictionary phrase we record it. If the current objects cannot be matched in the dictionary, they will be discarded. At the same time, the transformation of a determined compound object is obtained from the dictionary. The current objects are removed from the queue thereafter (line 7). This process is repeated until all the compound objects are determined (line 8). Finally, after all the compound objects are determined, the set of ROM representations $S_{ROM}$ and the set of objects $S_{0}$ (line 10) are updated. Fig. 8 shows the S-ROM with determined compound objects for the example sentence. **Algorithm 3.** Determining compound objects

1: $Q_{0} \rightarrow S_{0}$  
2: while $Q_{0} \neq \text{empty do}$  
3: $O_{x} \leftarrow \text{getHead}(Q_{0})$  
4: $x \leftarrow l_{0}$  
5: $y \leftarrow \text{StartP}(O_{x})$  
6: $z \leftarrow \text{Determine}(O_{x}, O_{y})$  
7: $Q_{0} \leftarrow Q_{0} \cup \{O_{x} \ldots O_{n}\}$  
8: $S_{CO} \leftarrow S_{CO} \cup \{O_{2} \ldots O_{n}\}$  
9: end while  
10: Update $S_{ROM}$, $S_{0}$, $AOC$  

Secondly, an effort is made to find as many objects as possible to compose a compound object. The first object $O_{i}$ of the queue $Q_{0}$ is seen as the last object of a compound object (line 3). The variable $x$ indicates the position of the current objects generated by composing all the objects between $O_{i}$ and $O_{i}$ is supposed to be a compound object. For every compound object $O_{i} = \{O_{1}, O_{2}, \ldots, O_{n}\}$, $O_{i}(i = 1, 2, \ldots, n)$ is a simple object and its stem is stem ($O_{i}$). We traverse all possibilities that $O_{i}$ can reach by getHead ($O_{i}$) or its stem ($O_{i}$). Once one of the case matches the dictionary phrase we record it. If the current objects cannot be matched in the dictionary, they will be discarded. At the same time, the transformation of a determined compound object is obtained from the dictionary. The current objects are removed from the queue thereafter (line 7). This process is repeated until all the compound objects are determined (line 8). Finally, after all the compound objects are determined, the set of ROM representations $S_{ROM}$ and the set of objects $S_{0}$ (line 10) are updated. Fig. 8 shows the S-ROM with determined compound objects for the example sentence. **Algorithm 4.** Object transformation

1: $S = O_{1}, O_{2}, \ldots, O_{n}$  
2: while $Q \neq \text{empty do}$  
3: if $O_{i} \in S \cap AO[C[O_{i}]]$ is the maximum then  
4: $T \leftarrow O_{i}$  
5: end if  
6: if $O_{i} \in StopList then$  
7: record the translation of $O_{i}$  
8: continue  
9: end if  
10: for each $O_{x} \in S$ do  
11: if $O_{x}$ restrict $O_{i}$ and $O_{x} \notin StopList then$  
12: $T \leftarrow T \cup O_{i}$  
13: end if  
14: end for  
15: $O_{i} \leftarrow \text{Trans}(T)$  
16: $S \leftarrow S \setminus O_{i}$  
17: $T \leftarrow \text{NULL}$  
18: end while

The third step of the CLT-ROM is object transformation, shown in Algorithm 4. Different transformational operations are carried out for the restricted semantic objects and non-restricted semantic objects. Usually, an object with more restricted semantics is the focus in understanding a sentence. Therefore, the transformation is conducted in a descending order according to the number of the object’s restricted semantics (line 3). In the algorithm, $AOC[O_{i}]$ stores the number of an object’s restricted semantics. The object with the largest restricted semantic number is transformed prior to the others. **Algorithm 4.** Object transformation

For a restricted semantic object, the transformation is directly obtained by querying the dictionary (line 11). Multiple candidate transformations may exist. In this case, other related objects make contributions to reduce the ambiguity. Several methods were proposed to reduce or eliminate the ambiguity. The first method is calculating the co-occurrence probability for the related objects to confirm the transformation. The second method is computing the semantic similarity of the related objects to eliminate the ambiguity. The semantics of a word in HowNet [32] can be used to compute the semantic similarity. Meanwhile, the AI information, such as POS, is also useful to optimize the transformation by reducing the variation of candidate transformations.
For the non-restricted semantic objects, a Stop_List is created to collect them. They are used to describe the statement between the restricted semantic objects. The corresponding candidate transformations of the Stop_List are also stored in the dictionary. Meanwhile, rules for the non-restricted semantic object transformation are developed. For example, some non-restricted semantic preposition objects, such as for, on and to, are transformed by combining with the related verbs. Some non-restricted semantic object transformations are combined with the voice and tense. Some non-restricted semantic objects have no meanings. After the object transformation is completed, the structure of ROM is updated, and the T-ROM is output. Fig. 9 shows the T-ROM for the example sentence.

5. Case study and evaluation

5.1. Case study

A prototype system, ROM-based Communication Tool (RCT), was developed to support online cross-language communication based on the CLT-ROM method. It is programmed and run in Windows XP laptop (1.83 GHZ Inter Core 2 Due CPU and 2 GB RAM) with a Java environment (MyEclipse 6.5 and JDK version 1.6.0_32). The RCT includes two clients: an English end and a Chinese one, for demonstration purpose. Each of them can take a text sentence as an input and output a ROM diagram, i.e. an English sentence for the English client and a Chinese sentence for the Chinese one. When a user submits an English sentence at the English interface, the system outputs the corresponding English ROM diagram on the Chinese client. It shares the same principle for a Chinese-to-English process. In this way, users from both clients can communicate with each other, sharing their ideas, and discussing the design problems. Just like popular instant messaging tools, people can use RCT to share their design and discussing the design problems. Just like popular instant messaging tools, people can use RCT to share their design

5.2. Evaluation

Since the transformation results of the RCT are graphs not sentences, the evaluation criterion of language translation cannot be directly applied. We define a criterion for transformation based on ROM evaluation, named TRE (Transformation based on ROM evaluation), which is a variation of BLEU (Bilingual evaluation understudy) [33].

5.2.1. BLEU (bilingual evaluation understudy)

Before giving an introduction to the TRE, some concepts about BLEU is addressed here. The BLEU is an evaluation criterion of language translation. The evaluation requires two ingredients, a numerical metric BLEU and a corpus of high quality reference translations. The BLEU can be computed by:

\[
\text{BLEU} = \text{BP} \cdot \exp \left( \sum_{n} w_n \log P_n \right)
\]

where \( P_n \) shown in Eq. (2) is a modified n-gram precision to capture two aspects of translation: adequacy and fluency. A brevity penalty factor BP is introduced to penalize the short candidate sentence. A translation using the same words as in the reference translations tends to satisfy the adequacy.

\[
P_n = \sum_{C \in \text{(candidates)}} \frac{\sum_{n \text{-gram} \in C} \text{Count}_{n \text{-gram}}(n \text{-gram})}{\sum_{n \text{-gram}} \text{Count}_{n \text{-gram}}(n \text{-gram})}
\]

Considering the recall of language translation, the length of a candidate sentence should match the effective length of a reference sentence. The BP can be calculated by:

\[
BP = \begin{cases} 
1 & \text{if } c > r \\
\exp(1-r/c) & \text{if } c \leq r
\end{cases}
\]

where \( c \) is the length of a candidate sentence. \( r \) is the length of a reference sentence.

5.2.2. TRE (transformation based on ROM evaluation)

Accordingly, the evaluation of cross-language transformation also requires two ingredients, a numerical metric TRE and a corpus of high quality reference transformations. TRE can be computed by the Eq. (4) which is similar to BLEU.

\[
\text{TRE} = \text{BP} \cdot \text{P}
\]

For the criterion TRE, since we transform a source sentence to a target ROM graph, \( P_1 \) in BLEU can be directly used in the TRE. A ROM based precision \( P_1 \) is defined, and can be computed by:

\[
\text{Pr} = \frac{\text{Count}(RR \cap TR)}{\text{Count}(TR)}
\]

where \( RR \) is a reference ROM representation. The candidate ROM representation is denoted by TR. First, we add up the clipped ROM representation counts. Then, the result is divided by the total number of TR. Hence, the ratio \( P \) of the criterion TRE can be computed by:

\[
P = \exp(w_1 \log P_1 + w_2 \log P_2)
\]

The score of \( P \) indicates the capability of semantic preserving. \( w_1 \) and \( w_2 \) are corresponding to the weight of \( P_1 \) and \( P_2 \). Here, \( w_1 = w_2 \). Similarly, considering the recall, we also introduce the brevity penalty BP to penalize the candidate ROM representation when the number of a candidate ROM representation is smaller.
than the number of a reference ROM representation. $BP$ can be computed by the same method that is defined in the BLEU. Therefore, we get the evaluation criterion $TRE$ shown in Eq. (7) by using the log form to facilitate the following comparison.

$$\log TRE = \min(1 - r/c, 0) + \log P$$

(7)

where $r$ is the effective number of the reference ROM representation, $c$ is the number of the candidate ROM representation.

5.2.3. Results of evaluations

5.2.3.1. Comparing with Google Translate in terms of semantic preserving. In order to indicate the capability of semantic preserving, we compared the results of CLT-ROM method with the results of Google Translate. It does not apply grammatical rules, since its algorithms are based on statistical analysis rather than rule-based analysis. The transformed ROM representation of a Google translated sentence is denoted by $GR$. Eq. (8) was used to calculate the precision of the $GR$, denoted by $P_r$.

$$P_r = \frac{\text{Count}(RR \cap GR)}{\text{Count}(GR)}$$

(8)

By using the criterion $TRE$, the respective score of semantic preserving were derived for the CLT-ROM method and Google Translate. The following eight sample sentences adopted from the book by Hubka et al. [34] are used as the test corpus. For each sentence, the scores $P_1$ and $P_r$ corresponding to both the CLT-ROM method and the Google Translate, are showed in Figs. 12 and 13, respectively.

(1) Design a tool for fixing brake linings onto brake shoes for internal drum brakes.

(2) The user of this tool is a car mechanic.

(3) The working height of the user should follow ergonomic standards.

(4) The use of this tool should conform to the related industry safety standards.

(5) The service life of this tool should be around 5 years.

(6) The tool should be easy for transportation and maintenance.

(7) It will be manufactured in a specific workshop, which has specified equipments.

(8) The cost of this tool cannot be over $190.00.

For the score of $P_1$, the Google Translate can make the right translations for part of the words in the sentences. The results from the CLT-ROM method are slightly better than from the Google Translate. However, for the score of $P_r$, which implies more semantic information, a significant difference exists between the two methods. The CLT-ROM method has a higher $P_r$ score than the Google Translate does. The results indicate that the Google Translate lost some semantics during the process of translation, while the CLT-ROM had a better capability of semantic preserving. The
results of comparative evaluations for total test corpus are shown in Table 4. According to the table, the CLT-ROM method has a higher score of $P_r$ and $TRE$ than that of the Google Translate. It also indicates that the proposed method has a better performance than the Google Translate in the capability of semantic preserving for cross-language transformation.

5.2.3.2. More on correctness tests. To evaluate the correctness of the transformation from English sentences to Chinese ROM diagrams, 50 sentences from all five basic sentence patterns in English are involved. The sentences chosen here, as listed as follows, are focusing on industrial requirements sharing and design problems discussing.

Pattern 1. Subject + intransitive verb
The use of this tool should conform to the related industry safety standards.

Pattern 2. Subject + link verb + subject Complement
Ultrasonic motors are of great interest due to the flexibility of...
minimization in comparison with conventional electromagnetic motors whose efficiency decreases significantly.

Pattern 3. Subject + transitive verb + object
The EEG can detect changes in electrical activity in the brain on a millisecond level.

Pattern 4. Subject + transitive verb + indirect object + direct object
The approach in designing these piezoelectric motors will bring the system more reliability.

Pattern 5. Subject + transitive verb + object + object complement
I think it impossible for us to finish the design task in such a short time.

We used two measures to evaluate the transformed Chinese ROM diagrams generated by the CLT-ROM method: the correctness of the objects and the correctness of the relations. For each sentence in the dataset, we draw a correct Chinese ROM diagram for the comparing purpose. After compared all the experimental results with the correct ones for the 50 sentences, we have achieved 81.7% for the object correctness and 82.7% for the relation correctness in the dataset. The future work includes three aspects. Firstly, the current experiments did not use a large-scale data set. The data set is to be improved to cover all types of grammatical structures. A more in-depth test is going to be conducted thereafter. Secondly, a ROM-net-based dictionary is to be built. With the dictionary, the transformation results will be improved. Thirdly, since user feedback is valuable for the quality of language transformation, we plan to use the feedback recursively to make the prototype adaptable. Translating of target ROM diagrams to a target sentences with ROM merge functions will be implemented in the next version of the prototype system as well.

Acknowledgments

This paper is supported by National Natural Science Foundation of China under Grant Nos. 61173170, 60873225 and 70771043, CCF Opening Project of Chinese Information Processing, central university basic research special funding (Innovation Fund of Huazhong University of Science and Technology under Grant Nos. 2011TS135 and 2010MS008). The authors would like to extend their heartfelt gratitude to Prof. Yong Zeng for his knowledge and valuable suggestions. The authors are also grateful to anonymous reviewers for their constructive comments on this paper.

References


