Two Phase Evaluation for Selecting Machine Translation Services

Chunqi Shi, Donghui Lin, Masahiko Shimada, Toru Ishida

Department of Social Informatics, Kyoto University
Yoshida-Honmachi,Sakyo-Ku, Kyoto, 606-8501, Japan
{shi,mshimada}@ai.soc.i.kyoto-u.ac.jp, {lindh,ishida}@i.kyoto-u.ac.jp

Abstract

An increased number of machine translation services are now available. Unfortunately, none of them can provide adequate translation quality for all input sources. This forces the user to select from among the services according to his needs. However, it is tedious and time consuming to perform this manual selection. Our solution, proposed here, is an automatic mechanism that can select the most appropriate machine translation service. Although evaluation methods are available, such as BLEU, NIST, WER, etc., their evaluation results are not unanimous regardless of the translation sources. We proposed a two-phase architecture for selecting translation services. The first phase uses a data-driven classification to allow the most appropriate evaluation method to be selected according to each translation source. The second phase selects the most appropriate machine translation result by the selected evaluation method. We describe the architecture, detail the algorithm, and construct a prototype. Tests show that the proposal yields better translation quality than employing just one machine translation service.

Keywords: Machine Translation, Evaluation, Service Selection

1. Introduction

Due to online access and instant availability, machine translation (MT) services are becoming more popular. One example is the online Google translation service. These MT services, in most cases, do not provide perfect accuracy or fluency. When multiple MT services are available, the user is confused about which service is more accurate for the task at hand. Manual service selection is tedious and error prone. Thus, it is necessary to create a mechanism that can select the most appropriate MT service.

Many functional equivalent MT services have become available. Language Grid (Ishida, 2011) is a service-oriented intelligence platform for language services. It provides many language translation services by wrapping non-networked language resources and software. With standard interfaces, functional equivalent translation services are formalized and made available for both end-users and community translation developers. The types of language services include machine translation services, dictionary services, parallel text services, and morphological analyzer services. Moreover, composite translation services could be generated based on these types of language services (Murakami et al., 2010). For example, Language Grid provides a multi-hop composite service. It combines a machine translation service and in-domain dictionary services, so as to provide a higher quality MT service for a desired domain. Many composite MT services can be created by generating different combination (Bramantoro et al., 2010). Given this multiplicity of translation services available, it is difficult for the user to select the MT service that best suits the current task.

Several evaluation methods can be used to evaluate translation results automatically, such as BLEU (Papineni et al., 2002), NIST (Doddington, 2002), WER (Nieen et al., 2000), etc. However, their evaluation results are not unanimous (Och, 2003; Cer et al., 2010). Moreover, the efficiency of the evaluation method will affect the selection of the MT service. It must be noted that these evaluation methods have incompatible metrics, and their results can have different distribution ranges.

The purpose of this paper is to provide MT service selection for end-users and community translation developers of Language Grid (Ishida, 2011). Community translation developers as well as end-users require assistance selecting the proper translation according to the translation quality, as well as other properties like time. The selection of a service according to the general quality of service (QoS) properties, such as time, cost, etc, has been well researched (Tian et al., 2004; Serhani et al., 2005). Thus, our research focuses on how to use these evaluation methods to calculate and rank MT services according to the translation quality, a domain-specific QoS property.

For the example of Japanese-English translation, there are two candidate services, Google Translate and J-Server, and three candidate evaluation methods, BLUE, NIST, and WER (see Figure 1). The evaluation results and translation results of two source sentences are given below. For the first sentence, Google gets higher evaluation results than J-Server, thus it will be selected by BLEU or NIST, while WER will select J-Server. The evaluation results of these evaluation methods do not agree with each other. For the second sentence, WER generates same evaluation results for Google and J-Server, BLEU generates only a slight different evaluation result, while NIST indicates a disparity between them. Here, we face the problem of how to make use of these evaluation methods to generate an evaluation result for machine translation service selection.

If a source sentence is given, the results of multiple evaluation methods can conflict. It is better to select a proper evaluation method for each source sentence, rather than using the same evaluation method continuously. For certain translation source, if more than one evaluation methods are appropriate, selecting one of them is proper. In the MT services selection, multiple translation evaluation methods are

1) TRANSLATION (Japanese → English)
Sentence: これはよく売れるが，しかしあまり効かない薬だ。
Reference: This is a medicine that sells well but is not very effective.
•Service Result
Google: It sells well, but this is medicine which doesn't work so much.
J-Server: This will sell well, but it very ineffective drugs.
•Evaluation Result
Google: *BLEU:0.21, NIST:1.41, WER:-1.00*
J-Server: *BLEU:0.15, NIST:1.12, WER:-0.75*

2) TRANSLATION (Japanese → English)
Sentence: 水が冷たくて手がちぎれそうだ。
Reference: The water is cold and my hands feel like they are to be torn off.
•Service Result
Google: Yes water is cold and torn hands.
J-Server: Water is cold, and a hand seems to come off.
•Evaluation Result
Google: *BLEU:0.18, NIST:0.29, WER:-0.67*
J-Server: *BLEU:0.15, NIST:0.97, WER:-0.67*

Figure 1: An example of two translation services (*Google* and *J-Server*) evaluated by three evaluation methods (*BLEU*, *NIST*, and *WER*)

available, thus, to achieve the goal of selecting a proper MT service, two main issues should be considered. (1) How to make use of multiple evaluation methods? We provide a two-phase architecture for service selection, which is an extension of the Web service broker for service selection. In the first phase, a proper evaluation method is selected. In the second phase, based on the selected evaluation method, the most appropriate machine translation service is selected. (2) How to realize service selection? To achieve this goal, we introduce a ranking algorithm It dynamically selects the appropriate evaluation method for each input translation source. We use data-driven classification for selecting the evaluation method. The machine translation service with the highest evaluation result as indicated by the selected evaluation is chosen.

2. Translation Service Selection Architecture

We extend the broker for Web service selection (see Figure 2) to create a two-phase architecture for MT service selection. For selecting Web service according to QoS properties, a Web service broker is flexible and trustworthy architecture for realizing the management of QoS properties for providers and users of Web services (Tian et al., 2004; Serhani et al., 2005). It receives request from and makes a response to Web service requestor. Meanwhile, it registers services from Web service providers, and verifies and certificates the properties said by Web services. A broker is also a Web service that can be published to and be found in a Web service registry; this makes it readily available to both end-users and new service developers. The broker architecture usually has a built-in evaluation method for each QoS property.

However, for MT service selection, it is not proper to adopt just one evaluation method to evaluate all translation services. Instead, the most appropriate evaluation method is to be selected for each input translation source. Thus, we propose a two-phase architecture for MT service selection.

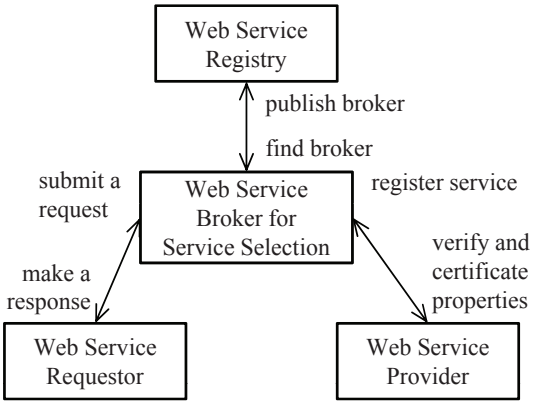

Figure 2: The Web service broker for service selection

2.1. Architecture

Firstly, we present an overview of our two-phase architecture (see Figure 3). The broker for MT service receives a request with translation source from and returns a response (translation result) to the MT service requestor. The extension is to register both MT services and evaluation methods for selection. Meanwhile, in addition to MT service selection, it first has to select the evaluation method. Several considerations are presented below before more details is given.

- Wrapping existent software of evaluation methods into services: Due to ongoing research into MT evaluation, new evaluation methods will emerge, and their software will be published. To provide an open-ended interface for integrating additional evaluation methods, it is useful to provide a self-describing Web service interface, wrapping existent software of evaluation methods into flexible online services (Eck et al., 2006). It is easily realized by the service wrapper function provided by the Language Grid platform.
- Regarding Language Grid as service provider: The Language Grid service-oriented platform successfully solves various service issues, such as creation, registration, and management. Due to the service description profiles of Language Grid, MT services category and evaluation methods category can be registered conveniently. Thus, Language Grid is an excellent provider for MT services and evaluation methods.
- Using data-driven classification to select evaluation method: Classification is necessary to realize evaluation method selection according to translation source.

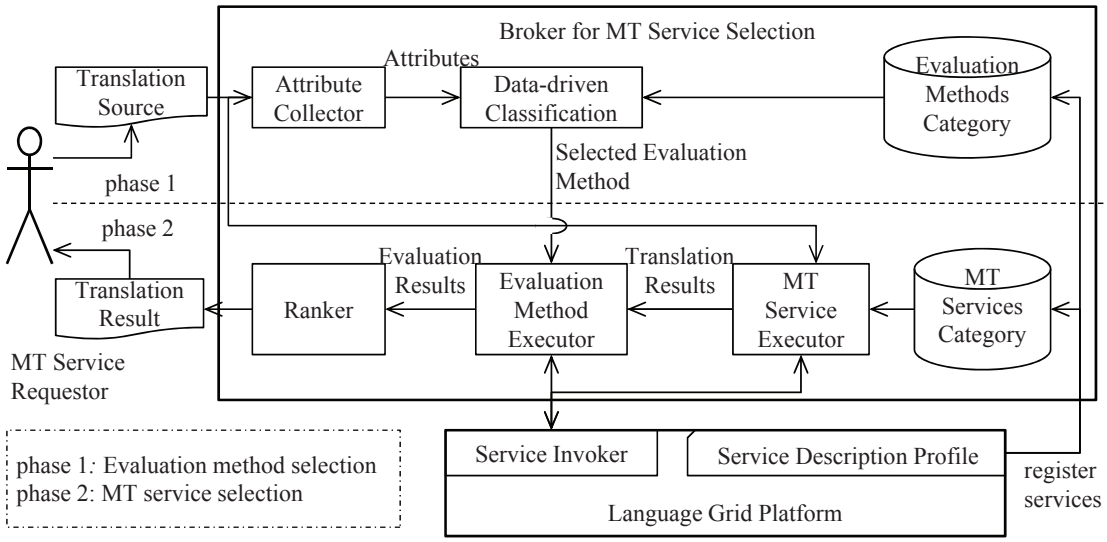

Figure 3: Two-phase architecture for machine translation (MT) service selection

Data-driven classification is suitable for this task. First, no experience is available for such classification. Second, because the attributes of sentences vary dynamically, the data-driven approach is more extensible. Data-driven classification builds a classification function dynamically from a training set. This set can be collected from human selection cases.

To realize data-driven classification, we adopt the decision tree approach. First, it offers quick training and classification, which is very user-friendly. Second, it is easy to transform a decision tree into decision rules, which is well supports manual verification. C4.5 algorithm (Quinlan, 1993), one of the most frequently used decision tree, is used in the evaluation method selection. It has several merits including handling missing values, allowing presence of noise, and realizing the categorization of continuous attributes. It should be noted that we view C4.5 as a 'black box' for the classification task; its original functionality was preserved.

The two-phase architecture of machine translation service selection (see Figure 3) is based on the above considerations. The broker for MT service selection divides its functions into the evaluation method selection phase and the MT service selection phase. Evaluation methods are handled in the former phase, and the output is the appropriate evaluation method. MT services are handled in the second phase, and the result of a translation service is selected in this phase. The main components of the broker include Attribute Collector, Data-driven Classification, Evaluation Methods Category, MT Services Category, MT Service Executor, Evaluation Method Executor, and Ranker (see Figure 3). Then, we describe the processes of two phases below.

- Evaluation method selection phase: A translation source from MT service requestor is analyzed by the attribute collector component, and the analyzed attributes are sent to data-driven classification component. According to the attributes of the translation source, an evaluation method is selected from methods in evaluation methods category component by the data-driven classification.
- MT Service selection phase: The translation source is send to the MT service executor component, which invokes the MT services from services in MT service category component. The translation results are sent to the evaluation method executor component, which invokes the selected evaluation method identified in the earlier phase. The evaluation results of the translation are sent to the ranker component, and the best translation result is send to the MT service requestor.

2.2. Deployment

We realized a prototype that implemented the above component functions as detailed below.

1) Evaluation method selection phase:

- Evaluation Methods Category: It is a simple MySQL[1] database holding stored service name, the URL, operation names and types, parameter names and types, and preset parameter values. There are three evaluation methods, *BLEU*, *NIST*, and *WER* methods, which are from Stanford Phrasal Evaluation project (Cer et al., 2010), and are wrapped into services by Language Grid platform.
- Attribute Collector: Two simple attributes are collected, the length of translation source and the source and target languages. The length of translation source is calculated as the number of words in the translation source.
- Data-driven Classification: J48 software, a Java implementation of C4.5 algorithm from Weka data mining tool [2], is used for classification. Its input is the attribute-value pairs output by the attribute

[1] http://www.mysql.com/
[2] http://weka.sourceforge.net/

collector component, and its output is the name of evaluation method, according to which, the details of evaluation method can be retrieved from the evaluation methods category component.

2) MT service selection phase:

- MT Services Category: It is similar to evaluation methods category component. Three services, Google, J-Server, and Translution services from Language Grid platform, are registered.
- MT Service Executor and Evaluation Method Executor: They are implemented based on JAX-RPC [3] service client, which makes it easy to invoke a Web service according to the name space, operation name and type, and parameter name and type.
- Ranker: A ranking algorithm is designed and implemented in Java. The input of this algorithm is the evaluation results of the selected evaluation method. The output of the algorithm is a selected translation result of highest QoS value of translation quality. The detail of this new algorithm will be explained in the following section.

The above two-phase architecture and components deployment, makes it convenient to realize the proposed broker for MT service selection.

3. Translation Selection Algorithm

After that, we illustrate the strategy of selecting the most appropriate MT service. To explain the selection algorithm in detail, a formal description is given as follows.

3.1. Selection

For a translation user, n translation services S=$\{s_1, s_2, \ldots, s_n\}$ are available, along with m evaluation methods E=$\{e_1, e_2, \ldots, e_m\}$. For each request translation source r, a proper evaluation method e_k is to be selected. According to this selected evaluation method e_k, a QoS value of translation quality $qos(e_k, s_i)$ is to be generated for each service s_i. Ranking these QoS values will determine the translation service s_{select} to be selected.

While q_{ji}, an evaluation result, is the result of applying the jth evaluation method e_j to the translation result of ith MT service s_i. However, these evaluation results are likely to conflict with each other, since they are generated by different evaluation methods. We need to select an evaluation method before we can select the service.

We use a decision tree to select the target evaluation method, e_k. For the request translation source r, the attribute collector collects c attribute values by the set of functions F=$\{f_1, f_2, \ldots, f_c\}$. If the decision tree is not trained, the decision rules are not generated. First, a training set is required, which are a set of translation sources, and for each translation source, a proper evaluation method is given. The attributes of these translation sources will be analyzed and used for training. Once the decision tree is trained, it easily generates the decision rules. Each decision rule can be described as follows:

$$(\theta_1^{low} < f_1(r) < \theta_1^{up}) \wedge \ldots \wedge (\theta_t^{low} < f_t(r) < \theta_t^{up}) \wedge \ldots \wedge (\theta_c^{low} < f_c(r) < \theta_c^{up}) \rightarrow e_k \quad (1)$$

Here, θ_t^{low} and θ_t^{up} are the lower and upper boundaries of tth collected attribute value $f_t(r)$ $(1 \leq t \leq c)$. When attributes are collected from the request translation source, each decision rule is test until the target evaluation method e_k is satisfied. Then, it will be sent to the next phase for execution.

After appropriate evaluation method e_k is selected, the translation quality of a service s_i is $qos(e_k, s_i) = q_{ki}$. The translation quality values of all the MT services can then be ranked, and target service s_{select} can be selected as follows.

$$s_{select} = \arg\max_{s_i} qos(e_k, s_i) \quad (2)$$

Thus, the algorithm will select an evaluation method e_k in the first. Then, based on the evaluation values, it will get QoS value (translation quality) for each service s_i. Finally, the QoS values will be ranked, and the target MT service s_{select} will be selected.

In addition, we design a normalization of evaluation result q_{ki}, which are probably in different metrics, due to different evaluation methods. The result of normalization q'_{ki} is given below, which is a relative value of the average translation quality values of whole MT services and average of MT services except s_i.

$$q'_{ki} = \frac{\sum_j q_{kj}/n}{(\sum_j q_{kj} - q_{ki})/(n-1)} \quad (3)$$

If evaluation results are positive, normalization of translation quality $qos'(e_k, s_i) = q'_{ki}$, otherwise, $qos'(e_k, s_i) = 1/q'_{ki}$. Getting a unitary measure is required for community translation developers to aggregate translation quality with other QoS properties such as time and cost.

3.2. Algorithm

We describe the algorithm that works in the broker for MT service selection, see Algorithm 1. It includes two-phase execution. In the first phase, if no decision rules exist, we need to train the decision tree, and generate decision rules. Next, we calculate attributes $\{f_1(r), f_2(r), \ldots, f_c(r)\}$ from request translation source r by attribute collector functions, then the attributes values are checked by decision rules. If decision rules exists, we can select a target evaluation method *selected_evaluation*, which completes the first phase.

In the second phase, it invokes the MT services S for translation results, evaluate translation results by the evaluation method *selected_evaluation* for evaluation results, and get evaluation scores $q(e_k, s_i)$ from evaluation results. Then it is easy to rank for the target result s_{select}.

There are one more issue need be mentioned here, training the J48 decision tree. We need human-generated translation selection data for training. To prepare each training data, we need to prepare several MT service results,

[3] http://java.net/projects/jax-rpc/

manually rank them, evaluate them by the multiple evaluation methods, and choose the evaluation method which best matches manually-generated ranking. With the target evaluation method, we calculate the attributes, train J48 with these attribute-data pairs, and generate decision rules from trained J48.

We use the example in Figure 1 to explain the *MT-Service-Select* algorithm. The input evaluation methods are *BLEU*, *NIST*, and *WER*. The input MT services are *Google* and *J-Server*. There are two Japanese sentences are the request translation sources. The attribute collector has one function, which counts the *translation-length*, the number of words in the translation source.

Algorithm 1: MT-Service-Select(E,S,r,F)

Input: E=$\{e_1, e_2, \ldots, e_m\}$: the m evaluation methods;
S=$\{s_1, s_2, \ldots, s_n\}$: the n MT services;
r: the request translation source ;
F=$\{f_1, f_2, \ldots, f_c\}$: the c attribute collectors ;

/* phase 1: Evaluation method selection */
if decision rules *not exist* **then**
    train decision tree by J48, and generate decision rules.
/* collect attribute values */
process translation source r by $\{f_1, f_2, \ldots, f_c\}$, and get $\{f_1(r), f_2(r), \ldots, f_c(r)\}$;
/* check decision rules, and select evaluation method */
selected_evaluation$\leftarrow \{e_k | (\theta_1^{low} < f_1(r) < \theta_1^{up}) \wedge \ldots \wedge (\theta_c^{low} < f_c(r) < \theta_c^{up}) \rightarrow e_k\}$;
/* phase 2: MT service selection */
$max \leftarrow 0$;
/* evaluate MT results */
foreach $s_i \in S$ **do**
    translate r by execute service s_i, and get translation result;
    evaluate translation result by *selected_evaluation*, and get q_{ki};
    $qos(e_k, s_i) \leftarrow q_{ki}$;
/* rank best service */
foreach $i \in \{1, 2, \ldots, n\}$ **do**
    /* select max quality score */
    if $max < qos(e_k, s_i)$ **then**
        $max \leftarrow qos(e_k, s_i)$ $s_{select} \leftarrow s_i$;
return s_{select};

In the first phase, it is assumed that the decision rules exist (see Section 4.1.). The *translation-length* of the first sentence is 21. Then each decision rule is checked and the last decision rule, *translation-length* $> 20 \rightarrow$ *BLEU*, matches. Thus, the *BLEU* evaluation method is selected for the first sentence. While the *translation-length* of the second sentence is 14, so the the *NIST* evaluation method is selected for it. Thus, for the first sentence, the *BLUE* is sent to the next phase, while for the second sentence, *NIST* is sent to the next phase.

In the second phase, for the first sentence, the MT services *Google* and *J-Server* are executed, and the service results are generated. Then the selected evaluation method *BLEU* is executed to evaluate the translation results, and the scores are 9.21 for *Google*, while 0.15 for *J-Server*. The results are compared, and the maximum is selected. Thus, for the first sentence, the translation result of *Google* is selected. For the second sentence, the translation result of *J-Server* is selected as per *NIST*.

Thus, our algorithm selects *Google* for the first sentence and *J-Server* for the second sentence.

4. Experiment

The experience analyzed the increase in translation quality and the efficiency of service selection offered by our proposal.

4.1. Preparation

The prototype was tested on three Japanese-English parallel text corpus, a NTT Communication Science Lab corpus (NTT), a medical corpus is used (Medical), and Tanaka corpus[4] (Tanaka). From 3,715 NTT corpus, 2,001 Medical corpus, and 150,127 Tanaka corpus. We sampled out 100 sentence pairs from NTT, Medical, and Tanaka, each, separately. The request data tested consisted of 300 sentences.

We randomly divided 300 sentences into six groups, each with 50 pairs. We trained the J48 decision tree using 60 additional pairs, that were sampled out in a similar manner. The training sets were selected through manual MT service results assessment. Only *translation-length* was gathered by the attribute collector. This length impacts evaluation method selection according to Och (Och, 2003). Finally, the generated decision rules were generated as following.

- *translation-length* $< 12 \rightarrow$ *WER*
- $12 <$ *translation-length* $< 20 \rightarrow$ *NIST*
- *translation-length* $> 20 \rightarrow$ *BLEU*

Two considerations of this experiment are given below.

- Parallel texts are used as translation source. One sentence of a parallel text pair is used as translation source, and the other is used as standard reference for evaluation. Evaluation methods, such as *BLEU*, *NIST* and *WER*, can generate more accurate evaluation results from the standard reference, so that the evaluation result will not be affected by reference quality.
- Human assessment following the manual method from DARPA TIDES projects [5] at University of Pennsylvania were used as the standard quality. It yields five-level scores for fluency and adequacy, {5:All , 4:Most , 3:Much, 2:Little 1:None}. The mean of fluency and adequacy score is used as the human assessment score of translation quality, which is used to assess the translation quality of selected translation results.

4.2. Analysis

Once the translation sources are submitted, the translation result of MT services is selected. Bases on the hu-

[4] http://www.edrdg.org/wiki/index.php/Tanaka_Corpus
[5] http://projects.ldc.upenn.edu/tides/translation/transassess04.pdf

man assessment, a *Hit Rate* is used to evaluate how well the output of the proposed mechanism matches the manual selection. *Average Score* is used to evaluate the translation quality of the output of the proposed mechanism. We explain this by the following example; a user submits 2 translation sources $\{r_1, r_2\}$, which are translated by two MT services $\{s_a, s_b\}$. The corresponding human assessment scores are $\{score(r_1, s_a)$:1, $score(r_1, s_b)$:4, $score(r_2, s_a)$:2, $score(r_2, s_b)$:5$\}$. Because service s_b gets larger scores for both r_1 and r_2, it is selected. Assuming that the proposed service selection works as intended, service s_a for source r_1 and s_b for r_2 are selected. Their human assessment scores are $\{score(r_1, s_a), score(r_2, s_b)\}$, as the *Average Score* of the proposed mechanism is $average\_score = (score(r_1, s_a) + score(r_2, s_b))/2 = (1+5)/2 = 3$. The *Hit Rate* of the proposed mechanism is $hit\_rate = (0+1)/2 = 50\%$, because for the first source r_1, the proposed mechanism selects s_a while the human selects s_b, which are different. For the second source r_2, they both select s_b. Thus, they have one common selection for two translation sources. The hit rate represents how well the proposed service selection follows manual selection.

The results achieved when no service selection is performed are shown in Table 1. Google received average human score of 3.37, J-Server got 3.43, and Translution got 3.06. Manual selection on the three sets of translation results yielded 116 sentence by Google, 143 by J-Server, and 41 by Translution. Compared to manual selection, the hit rate of Google is 62.8%, J-Server is 67.5%, and Translution is 54.0%. J-Server has highest average and hit rate for this Japanese-English translation task. From the hit rate, we find that no MT service dominates the other services (otherwise its hit rate will be 100%).

Table 1: Average score and hit rate of MT services

Service	Average Score	Hit Rate
Google (G)	3.29	62.8%
J-Server (J)	3.43	67.5%
Translution (T)	3.06	54.0%

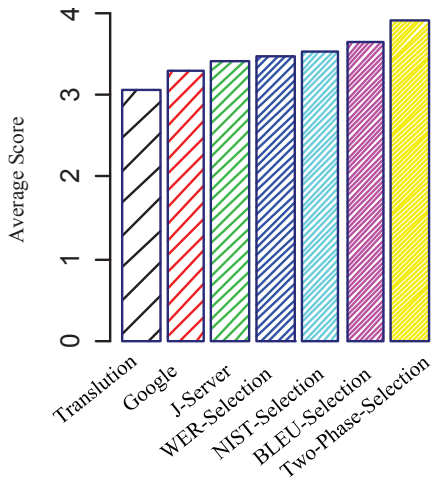

Figure 4: Comparison of Average Score

The results achieved when service selection is performed, are shown in Table 2. We compare the use of only one evaluation method and the proposed two-phase selection, which selects from among the evaluation methods available. Using just only one evaluation method, WER, the average score and hit rate are 3.47 and 72.0%, which is a little better than J-Server. BLEU has heigher average score and hit rate than WER and NIST: 3.56 and 76.2%. While using the proposed two-phase selection mechanism, the average score and hit rate of MT service selection are 3.81 and 81.7%. From the comparison of *Average Score* (see Figure 4) and *Hit Rate* (see Figure 5), we find that the proposed two-phase selection raises the translation quality received by users.

Compared to BLEU selection in Table 2, the proposed two-phase selection has higher hit rate and 7% promotion on average score. Moreover, compared to only one service in Table 1, like just J-Server, the two-phase type selection mechanism offers an 11.1% increase in average score.

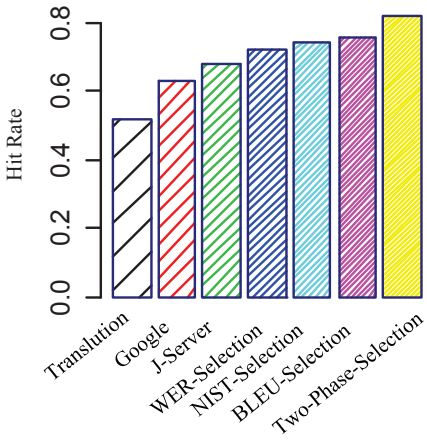

Figure 5: Comparison of Hit Rate

4.3. Discussion

Some limitations of the proposed two-phase evaluation for MT service selection are considered. First, the existing evaluation methods limit the gains possible with the proposed mechanism. The mechanism is not intended to establish a new evaluation method, but to make better use of existing methods. Creating a superior evaluation method is one of the hardest issues in machine translation and natural language processing. Thus, it is meaningful to achieve progress through better utilization of existing evaluation methods. Note that it is easy to import newly created evaluation methods into the proposed mechanism.

Second, data driven classification needs a large training set, which involves time consuming manul effort. Mining users logs to build more training sets will be very helpful. If we tell a MT service user that his feedback will help to promote translation quality, he will be more willing to generate useful MT service usage logs. Moreover, we already trying to integrate human activities into composite service (Lin et al., 2010). Success in this are will make it easier to prepare large training sets.

Third, application of the proposed MT service selection for

Table 2: Comparing average score and hit rate of two selection types

MT Selection Type	Selected Translation Results of MT Services (total 300)	Average Score	Hit Rate
WER selection	122 from Google, 127 from J-Server, 51 from Translution	3.47	72.0%
NIST selection	127 from Google, 125 from J-Server, 48 from Translution	3.53	74.5%
BLEU selection	134 from Google, 122 from J-Server, 44 from Translution	3.56	76.2%
Two-phase selection	115 from Google, 146 from J-Server, 39 from Translution	3.81	81.7%

community translation developers was not presented in detail. Because the proposed broker for MT service selection, is itself is a Web service, after it is published in Web service registry, and the translation quality value of unitary measurement is given (see Section 3.1.). Thus, it can be treated as a single MT service.

5. Related Work

First, automatic evaluation methods have been proposed on many mechanisms, includes string-based comparison, syntactic mechanism, and semantic mechanism. String-based comparison compares the translation result to standard references, and it is currently the most popular mechanism. There are several ways to compare the similarity, including lexical distance, and n-gram precision. Two common lexical distances are length of least common sub-string, such as ROUGE-L (Lin, 2004), and ROUGE-W (Lin, 2004), and edit distance, such as WER (Nieen et al., 2000) and TER (Snover et al., 2006b). N-gram precision has also been extensively studied, such as BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), and ROUGE-N (Lin, 2004). The syntactic mechanism analyzes whether the translation result is in accordance with the syntax of target language, such as linguistic error classification (Farrs et al., 2012). The semantic mechanism checks whether the translation result semantically agrees with the translation source, such as the lexical semantic similarity integration (Wong, 2010). When software of these evaluation methods have been prepared for sharing they can be wrapped into services by Language Grid platform, which makes our proposal more powerful.

Next, human evaluation is important to confirm any automatic evaluation method using norms such fluency and adequacy. Semi-automatic evaluation has also received a lot of attention, such as the evaluation method HTER (Snover et al., 2006a), which requires human editing. The proposed mechanism also requires human assessment for preparing training set, which builds up the data-driven classification.

Last, making evaluation methods easier to access is becoming a strong demand. There are some research on how to prepare references for these evaluation methods. Currently, there is no powerful way to utilize unsupervised references. Though many studies have pointed out that round-trip translation is not adequate, others treat round-trip translation as the easy approach with the lowest costs (Hu, 2009). Research is progressing on ways to provide standardized interface (Cer et al., 2010) or even evaluation services (Eck et al., 2006), so that these functions can be utilized by more people. The proposed mechanism has benefited a lot from such existing research.

6. Conclusion

We proposed a two-phase evaluation for MT service selection that suits for both end-users and community translation developers. Because of increased the number of MT services, we face the problem of selecting, for the given translation source, the best MT service. Considering ease of implementation and extendibility, we designed two-phase architecture for selecting MT services. In the first phase, we import multiple evaluation methods, analyze attributes of the translation source, and select the most appropriate evaluation method using the decision tree approach. This data-driven classification enables one among multiple evaluation methods to be selected dynamically, and voids the deficiencies raised by employing just one evaluation method. In the second phase, the MT services are executed based on the Language Grid platform. The results of MT services are evaluated by the selected evaluation method. The translation quality values are generated and ranked yielding the best translation result. We designed an algorithm for this MT service selection.

Finally, we implemented and tested a prototype based on the proposed mechanism. The results showed that the proposed mechanism offers better translation quality than employing just one MT service. Our proposal raises the translation quality by 7% compared to the approach which employs just one evaluation method, and at least 11.1% promotion than employing just one MT service.

Acknowledgement

This research was partially supported by Service Science, Solutions and Foundation Integrated Research Program from JST RISTEX.

7. References

S. Banerjee and A. Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the 43rd Annual Meeting of the Association of Computational Linguistics (ACL-2005)*, pages 65–72.

Arif Bramantoro, Ulrich Schfer, and Toru Ishida. 2010. Towards an integrated architecture for composite language servicesand multiple linguistic processing components. In *LREC 10*, pages 3506–3511.

Daniel Cer, Michel Galley, Daniel Jurafsky, and Christopher D. Manning. 2010. Phrasal: a toolkit for statistical machine translation with facilities for extraction and incorporation of arbitrary model features. In *Proceedings of the NAACL HLT*, pages 9–12, Stroudsburg, PA, USA. ACL.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Human Language Technology conference (HLT-2002)*, page 128132.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2006. A flexible online server for machine translation evaluation. In *Proceedings of EAMT 2006*, Oslo, Norway.

Mireia Farrs, Marta R. Costa-Juss, and Maja Popovic Morse. 2012. Study and correlation analysis of linguistic, perceptual, and automatic machine translation evaluations. In *JASIST*, pages 174–184.

Chang Hu. 2009. Collaborative translation by monolingual users. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI EA '09, pages 3105–3108, New York, NY, USA. ACM.

Toru Ishida. 2011. *The Language Grid: Service-Oriented Collective Intelligence for Language Resource Interoperability*. Springer.

Donghui Lin, Yoshiaki Murakami, Toru Ishida, Yohei Murakami, and Masahiro Tanaka. 2010. Composing human and machine translation services: Language grid for improving localization processes. In *LREC 10*, pages 500–506.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26.

Yohei Murakami, Donghui Lin, Masahiro Tanaka, Tako Nakaguchi, and Toru Ishida. 2010. Language service management with the language grid. In *LREC*, pages 3526–3531.

Sonja Nieen, Franz Josef Och, Gregor Leusch, and Hermann Ney. 2000. An evaluation tool for machine translation: Fast evaluation for mt research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, pages 39–45.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st ACL*, pages 160–167, Stroudsburg, PA, USA.

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

J. Ross Quinlan. 1993. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

M. Adel Serhani, Rachida Dssouli, Abdelhakim Hafid, and Houari Sahraoui. 2005. A qos broker based architecture for efficient web services selection. In *Proc. 2005 IEEE International Conference on Web Services (ICWS05*, pages 113–120. IEEE Computer Society.

M. Snover, Dorr B., R. Schwartz, L. Micciulla, and J. Makhoul. 2006a. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation*, page 223231.

Mathew Snover, Bonnie Dorr, Richard Schwartz, John Makhoul, and Linnea Micciula. 2006b. A study of translation error rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas Conference 2006*, pages 223–231.

M. Tian, A. Gramm, H. Ritter, and J. Schiller. 2004. Efficient selection and monitoring of qos-aware web services with the ws-qos framework. In *in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 152–158.

Billy Tak-Ming Wong. 2010. Semantic evaluation of machine translation. In *Proceedings of LREC'10*. European Language Resources Association (ELRA), may.