

Service Composition Scenarios for Task-Oriented Translation

Chunqi Shi, Donghui Lin, Toru Ishida

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

Abstract

Due to instant availability and low cost, machine translation is becoming popular. Machine translation mediated communication plays a more and more important role in international collaboration. However, machine translators cannot guarantee high quality translation. In a multilingual communication task, many in-domain resources, for example domain dictionaries, are needed to promote translation quality. This raises the problem of how to help communication task designers provide higher quality translation systems, systems that can take advantage of various in-domain resources. The Language Grid, a service-oriented collective intelligent platform, allows in-domain resources to be wrapped into language services. For task-oriented translation, we propose service composition scenarios for the composition of different language services, where various in-domain resources are utilized effectively. We design the architecture, provide a script language as the interface for the task designer, which is easy for describing the composition scenario, and make a case study of a Japanese-English campus orientation task. Based on the case study, we analyze the increase in translation quality possible and the usage of in-domain resources. The results demonstrate a clear improvement in translation accuracy when the in-domain resources are used.

Keywords: Machine Translation Mediated Communication, Task-Oriented, Scenario, Service Composition

1. Introduction

Machine translation mediated communication is being demanded with greater urgency. Because most people have difficulty in mastering a foreign language, international collaboration makes translation mediated communication essential. Compared with human translation, machine translation (MT) systems have the advantages in of lower cost, higher speed, and greater availability, all of which are attracting increasing numbers of users. Though MT translation becomes much more popular, in most cases, MT systems do not provide assured translation quality. In international collaboration, the accurate transfer of concepts (for example, facts, experiences, or theories), is very important and necessary, especially for task-oriented translation (compared with social/emotion-oriented services). Thus, in task-oriented machine mediated translation, the words and sentences expressing state concepts should be translated with higher accuracy. For example, in the task of providing campus orientation to a foreigner, disaster can result if the graduate policies are wrongly translated, or the locations of important actions are wrongly translated. We have already studied how to automatically locate the occurrence of misconceptions in MT mediated communication (Yamashita and Ishida, 2006).

Here, we focus on designing an MT system for a task-oriented MT mediated communication, with the goal of making good use of in-domain resources to improve the translation accuracy. Integrating more in-domain resources (for example, dictionaries and parallel texts) is an efficient way of raising translation accuracy (Wu et al., 2008). Generally speaking, the designer of an MT mediated communication task can clearly master the overall requirements of the in-domain resources needed, which is helpful in raising translation accuracy. However, since most potential task designers are not computing professionals, a mechanism is

needed to help them by providing the information of in-domain resources and allowing their easy use. Besides, considering the multiplicity of short-term communication tasks, the way of training a new system is too costly and too inflexible for the task designer. Thus, a light-weight mechanism to handle in-domain resource usage is needed.

To design such a light-weight MT system, two main issues should be addressed. (1) How to best utilize the in-domain materials in a light-weight manner. Many studies have examined domain adaptation for promoting translation quality. For example, in-domain dictionary and parallel text are used for training language model in a statistical machine translation system (Och et al., 1999; Wu et al., 2008). (2) How to help the task designer by providing the necessary information. Different communication tasks have different topics and different participants (Bangalore et al., 2006). When topics are changed, the task designer has to identify the information of the new in-domain resources needed to perform the translation.

To utilize in-domain resources, we choose service computing techniques. The Language Grid platform (Ishida, 2011; Murakami et al., 2010) provides mechanism to wrap resources, software, or online services into language services, including machine translators, parallel texts, dictionaries, and morphological analyzers. Moreover, many mechanisms for creating composite language services have been designed. Among them, the composite translator, which is based on multi-hop translation, combines a dictionary service and a translator service to improve translation quality. It takes two steps to utilize in-domain resources. First, we wrap in-domain resources, such as dictionaries and parallel texts, into a dictionary service and a parallel text service. Second, we provide a composite language service for translation based on those dictionary services, parallel text services, and machine translator service.

For the latter, we use a clear scenario-based mechanism to analyze the requirements of the topics present in a communication task, and the mapping of language services. Moreover, we design a simple declarative script language as the interface that allows the task designer to describe this scenario. Based on the described scenario, a composite service is generated for task-oriented translation.

2. Motivation

The emergence of Web techniques has changed both the way of human communication and the usage of machine translation. With regard to the former, machine mediated communication becomes so popular that people have begun to live through the Internet, learn through the Internet, and work through the Internet. Moreover, international collaboration has become much more common. The chance of multilingual people communicating with each to finish a task has been greatly increased. It means that machine translation mediated communication will be very helpful, if the translation quality is adequate. Different from social/emotion-oriented communication, task-oriented communication is more domain specific, which means that the domain involved can be predicted before communication. Thus, better translation quality can be achieved for task-oriented MT communication by preparing the appropriate in-domain resources. For the latter, the translation services have emerged as an easy way to access translations, like Google Translate ¹. People, who have created their in-domain resources are beginning to share them through the Internet. For example, a large number of services are shared through Language Grid. Moreover, research on MT service selection and aggregation is providing new ways to enhance translation quality (Ishida, 2011)

In task-oriented MT mediated communication, the participants of communication, such as the task designer and the communication subjects, can be active in promoting translation quality if proper in-domain resources are available. The task designer can prepare the necessary in-domain resource, like dictionaries, for reuse. Communication subjects can choose different expressions to better transfer their ideas, which is different from document translation. However, such participation in quality promotion has been ignored. To facilitate such participation, a light-weight solution is necessary for in-domain resources utilization.

Take the example of a Japanese-English orientation task on the Internet. The task designer planned two topics: legal procedures and student life. If only Google was used for translation, the translation quality of student life information would be poor because of the in-domain information like the address location. The translation quality can be promoted by using a dictionary holding suitable address information. Of course, it happens that the communication subjects manually post-edit the translation result according to the dictionary results. However, manual usage of these in-domain resources will reduce the communication efficiency, and raise the problem of inconsistent translation of ignored words.

Thus, our scenario design aims to enable the task-oriented communication designer to participate in raising the quality, by creating different composite language services with suitable in-domain resources.

3. Scenario as Requirement

A scenario of MT mediated communication is a description of the topics that are likely to be raised in the designed task (see Figure 1). The content of task-oriented communication can be partitioned into topics (Bangalore et al., 2006). To succeed in task-oriented translation, high quality translations in each topic are necessary. Meanwhile, there are subjects (sender and receiver) of the communication task. And there are requirements of translation properties from the subjects. Thus the task designer should write scenarios for improving task-oriented translation.

For example, in the Japanese-English orientation task, the scenario aims at supporting translations for the Japanese student to introduce school information to the English student. The task has two topics: legal procedures and student life. With regard to legal procedure translations, existing high quality parallel texts can be used. Next we consider how to make scenario descriptions.

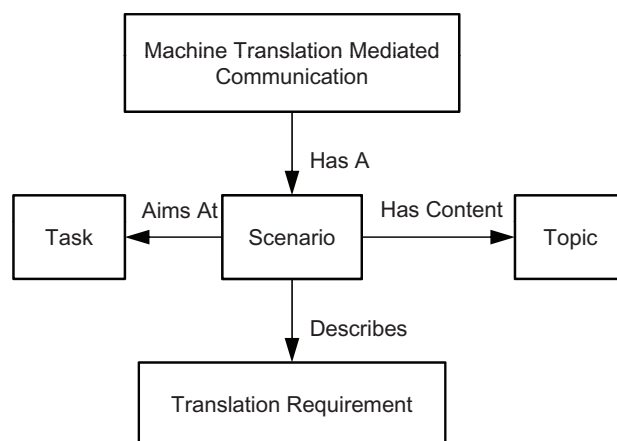


Figure 1: The role of a scenario in a task-oriented machine translation mediated communication

Here, a scenario describes the translation of each particular topic (see Figure 2). First, the sequence of topics obviously affects the scenario description. There are generally two types of sequences: fixed and dynamic. The medical reception for foreigners is a typical fixed sequence. With regard to dynamic sequences, remote fault diagnosis has no fixed topics, because different faults are encountered. It is easy to detect fixed sequence topics by using techniques such as tracking and boundary segmentation. To detect dynamic ones requires classification or searching according to feature data, such as comparable texts or keywords (Shen et al., 2006).

Second, with detected topics, the goal is to map the proper MT services for each topic. Based on existing research on the selection and aggregation of language services, it is possible to utilize several functionally-equivalent translation services for selection, or describe property require-

¹<http://translate.google.com/>

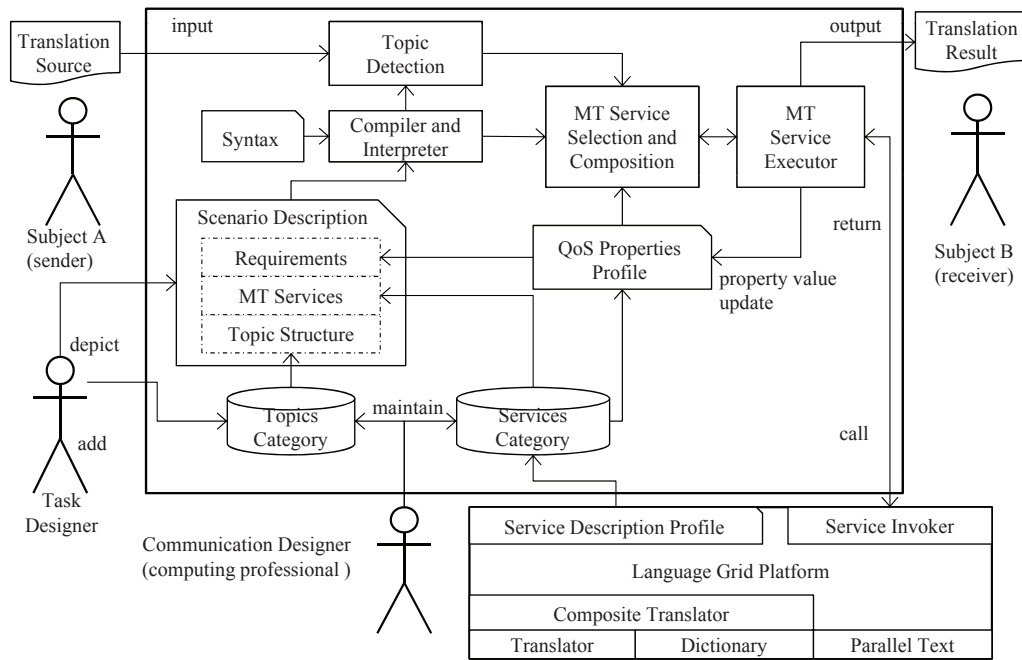


Figure 3: Architecture of scenario based service composition for task-oriented machine translation (MT)

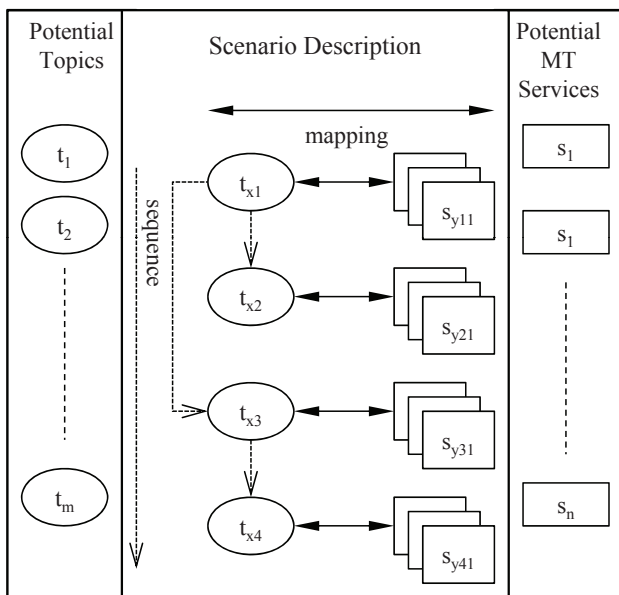


Figure 2: Scenario description aims to appoint proper machine translation (MT) services for each target topic

ments like time, cost for composition, or even consider human translation for the highest translation quality (Lin et al., 2010).

For example, in the campus orientation task, there are two fixed sequence topics: registering procedure (t_{x1}) and graduation policy (t_{x2}) (see Figure 2). For the first topic, Google Translate can be used (as S_{y11}). Higher quality parallel text can also be used (as S_{y12}). For simplicity, we can track certain keyword “graduate” to detect the second topic. For the second topic, Google can still be used for translation (as S_{y21}). A school location dictionary (as S_{y22}) can be added. Then we need to composite and select the transla-

tion services for each sentence in each topic (S_{y12} and S_{y12} for t_{x1} , S_{y21} and S_{y22} for t_{x2}).

From the reuse angle, different communication tasks can share their topics and thus available services. A topic that has already been configured and categorized, can be reused by the task designer. Moreover, the mapping MT services can be reused, if they have been suggested by a previous task designer. Thus, the potential topics and potential MT services can be maintained and reused. For example, given configured topics $\{t_1, t_2, \dots, t_m\}$ and language services $\{s_1, s_2, \dots, s_n\}$, the duty of a task designer will be to choose the proper topics $\{t_{x1}, t_{x2}, t_{x3}, t_{x4}\}$ and proper MT services $\{s_{y11}, s_{y12}, \dots\}$ 2).

The role of a scenario was explained above. In the following, we show how to realize scenario-based service composition, and for communication task designers, how to write scenarios in detail.

4. Task-Oriented Translation

This section describes how to realize proper MT service composition based on the scenario description.

4.1. Architecture

Our architecture for creating a composing MT service given a topic source is shown in Figure 3. We start with the four actors in this MT system: sender, receiver, task designer, and communication designer.

- Sender and receiver: As the subject in the communication task, the sender has several planned topics for the task, but he/she has freedom to elaborate the topic content, which will be translated and sent to the receiver. Such communication is different from question and answer sessions in that the receiver will interact

based on his own understanding and provide feedback to the sender. Given the receiver's feedback, which indicates the status of understanding, the sender can explain further as needed. But they face the problem of poor translation, which breaks the communication circle. If the translation quality can be promoted, they will communicate more fluently.

- Task designer: He/she has a clear image of topics associated with this task, the requirement of the subjects, and information of in-domain resources. The designer has the duty of configuring in-domain resources for raising translation quality, which is essential for efficient communication. This designer depicts the scenario of this task, including the topic structure, mapped MT services, and requirements (more details are explained in Section 4.2.). Note that the task designer is not likely to be a computing professional, so a simple interface is preferred.
- Communication designer: He/she represents a computing professional person, who can implement and configure the system, especially the management of categories of potential topics and MT services. Mostly, this person is not often involved with the details of concrete communication tasks.

The inner architecture includes four main components: topic detection, compiler and interpreter, MT service selection and composition, and MT service executor (see Figure 3). The functions of four components are depicted as follows.

- 1) Topic Detection: It locates topics in the source messages from the sender. The categories of topics are the output of the Compiler and Interpreter component. The content of detected topic will be translated by mapped MT services.
- 2) Compiler and Interpreter: According to syntax, they compile the scenario description script provided by the task designer, and interpret the topics, MT services, and requirements.
- 3) MT Service Selection and Composition: They deduce the most appropriate composite solution for translating the detected topic content. For each detected topic, based on the requirements and the candidate MT services, one translation service will be selected.
- 4) MT Service Executor: It accesses the language grid platform and returns the translation results. The results are sent to the receiver.

We briefly introduce the implementation of the main components below.

- 1) Topic Detection: We implement this function by simply tracking appointed keywords for fixed topics, or classifying keywords for dynamic topics. For complex situations, existing research can be used (Allan, 2002).
- 2) Compiler and Interpreter: We use SWI-Prolog² for compiling and interpreting the declarative interface language, which is easy.

- 3) MT Service Selection and Composition: After interpretation, the requirements on MT service are interpreted to yield quality of service (QoS) constraints. Then QoS-based MT service selection and composition can be applied (Yu et al., 2007).
- 4) MT Service Executor: A Language Grid client is used, and the selected service can be executed upon the input of service name and parameters. The translation result is passed to the receiver. Thus, various services can be invoked from the Language Grid platform.

Two category databases (based on PostgreSQL³) are illustrated functionally.

- Topics Category: It holds topic name (as keyword) and information for topic detection, such as description, keywords, and task relationship. It is maintained by the communication designer, and new topics can be submitted by the task designer.
- MT Services Category: It holds MT service name, URL, operation names, and parameters. It is maintained by the communication designer.

4.2. Scenario Description Language

For scenario description, we provide a declarative script language as interface, Scenario Description (SDL) Language. A Backus-Naur Form (BNF) definition is described (see Table 1). It contains three parts: topic structure, MT services, and requirements.

- Topic Structure: The topics of different grain levels and their sequence are depicted by a topic forest $\langle \text{topic-forest} \rangle$. Fixed topic sequences can be well depicted by topic tree $\langle \text{topic-tree} \rangle$. For the dynamic topic sequence, it is depicted as a set of fixed topics, which is a topic forest.
- MT Services: Several candidate services are mapped to a topic $\langle \text{topic-name} \rangle$. A service variable $\langle \text{service-variable} \rangle$ represents these candidates $\langle \text{candidates} \rangle$. Two relationships, the alternative selection and composition of dictionary and translator, are depicted by two marks, ‘|’ and ‘+’, respectively, for example, *foreign-life-para|google+city-dict* (see Figure 4).
- Requirements: A requirement variable $\langle \text{requirement-variable} \rangle$ is mapped to a topic $\langle \text{topic-name} \rangle$. A requirement is depicted as a list of constraints $\langle \text{constraint-list} \rangle$ on properties.

For example, the task designer wants to plan the task of campus orientation communication. It includes two fixed sequenced topics, *legal_procedure* and *student_life* (see Figure 4). They are noted as two top-grained topics, each of which has low-grained sub-topics. Here, *office* and *warning* are the low-grained sub-topics of *legal_procedure*. Each sub-topic is mapped variables of MT services, e.g. *office* is mapped with *Serv1*. For the task, on the root topic *campus_orientation*, a requirement variable is mapped *QosCo*. Finally, the variables, *Serv[1-5]* and *QosCo*, are concreted

²<http://www.swi-prolog.org/>

³<http://www.postgresql.org/>

Table 1: Scenario description language for general translation service developer

Name	Syntax	Description
Program	$\langle program \rangle ::= \langle topic_forest \rangle \langle services \rangle \langle requirements \rangle.$	Program contains topic structure, MT services and user requirements.
Topic Structure	$\langle topic_forest \rangle ::= \langle topic_tree \rangle \langle topic_forest \rangle.$ $\langle topic_tree \rangle ::= \langle topic \rangle ' := ' \langle topic_list \rangle.$ $\langle topic \rangle ::= \langle topic_name \rangle (\langle service_variable \rangle, \langle requirement_variable \rangle).$ $\langle topic_list \rangle ::= \langle topic \rangle \langle topic_list \rangle, \langle topic \rangle.$	Parent-child link depicted by ' := '; Precedence depicted by sequence list; Each topic combined with a service variable and a requirement variable.
MT Services	$\langle services \rangle ::= \langle service \rangle \langle services \rangle \langle service \rangle.$ $\langle service \rangle ::= \langle service_variable \rangle ' := ' \langle candidates \rangle.$ $\langle candidates \rangle ::= \langle service_name \rangle \langle candidates \rangle ' ' \langle service_name \rangle.$ $\langle service_name \rangle ::= \langle atomic_name \rangle \langle service_name \rangle ' + ' \langle atomic_name \rangle.$	MT Service variable assigned with a list of service candidates; Alternatives connected with ' '; Composite service depicted by '+'. Requirement variable assigned with a list of constraint;
Requirements	$\langle requirements \rangle ::= \langle requirement \rangle \langle requirements \rangle \langle requirement \rangle.$ $\langle requirement \rangle ::= \langle requirement_variable \rangle ' := ' \langle constraint_list \rangle.$ $\langle constraint_list \rangle ::= \langle constraint \rangle \langle constraint_list \rangle, \langle constraint \rangle.$ $\langle constraint \rangle ::= \langle property_name \rangle \langle operator \rangle \langle value \rangle.$	Each constraint is a value limitation on certain property.

by the candidate services and constraints. (see Figure 4).

```

campus_orientation(_, QoSCo):=
    legal_procedure(_, _), student_life(_, _).
legal_procedure(_, _):=
    office(Serv1, _), warning(Serv2, _).
student_life(_, _):=
    tuition(Serv3, _), class_schedule(Serv4, _),
    health_check(Serv5, _).

Serv1:= foreign-life-para | google + city-dict |
    j-server + city-dict.
Serv2:= crime-disaster-para | google + city-dict |
    j-server + city-dict.
Serv3:= school-life-para | google + edu-dict |
    j-server + edu-dict.
Serv4:= google + edu-dict | j-server + edu-dict.
Serv5:= medic-para |
    google + medic-dict + city-dict |
    j-server + medic-dict + city-dict.

QoSCo:= cost = 0.
    
```

Figure 4: A script of scenario description for campus orientation task

("-para": parallel text; "-dict": dictionary; "+": composition of dictionary and translator; "|": alternative services; "_": empty combination)

4.3. Service Composition Algorithm

The composition algorithm will choose the most appropriate service composition, when the topic structure, MT services, and requirements are provided. Here, considering the cost of time, we provide a simple translation service composition algorithm that focuses on providing adequate quality rather achieving the optimum quality. It selects candidate services for each topic, by checking whether the combination of candidate services fulfill the requirement constraints. It includes two parts. The former prepares the necessary data (see Algorithm 1). The latter uses the back-

tracking strategy to check the potential candidates (see Algorithm 2).

In preparing the necessary data (see Algorithm 1), we briefly describe the input of MT service composition. As mentioned previously, topics can be represented as topic forests. Because of the independence between topic trees in the topic forest, we only focus on the topic tree.

Topic Structure (*TopicStructure*):

- Leaf Topics (*LeafTopic*): It represents a unit of communication content, the lowest-grain topic. The task designer will appoint service candidates to it. Meanwhile, it is constrained by the requirements of its ancestor. For example, service candidate *Serv1* is mapped to leaf topic *office*, which is constrained by *QoS* (see Figure 4).
- Non-Leaf Topics (*NonLeafTopic*): It represents the structure of topics, which is the ancestor of a leaf topic. A designer might appoint requirements to it. For example, requirement *QoS* is mapped to non-leaf *campus_orientation* (see Figure 4).

MT Services (*MTServices*):

- Candidate Service (*Candidates*): They represent service choices mapped to one leaf topic. For example, the topic *office* is mapped to *foreign-life-para* parallel text service, *google + city-dict*, service, and *j-server + city-dict* service (see Figure 4).

Properties and Requirements (*Properties* and *Requirements*):

- Property Values (*PropertyValues*): They represent the QoS properties of one candidate service. The values of QoS properties are taken from the QoS properties profile, and are updated during execution (see Figure 3).
- Constrains (*Constraints*): They represent the overall constraints on the MT services mapped to topic. For example, *cost=0* means the *cost* property of services should be zero in translating the whole topic of *campus_orientation* (see Figure 4).

To check service candidates(see Algorithm 2), we check the combination of service candidates of leaf topics by apply-

Algorithm 1: Service-Composition (*TopicStructure, MT-Services, Properties, Requirements*)

Input: *TopicStructure*: the topic structure, including leaf topics and non-leaf topics;
MTServices: a set of candidate MT services mapped to the leaf topics;
Properties: the QoS properties of all the candidates services;
Requirements: the constraints on all the non-leaf topics;

```

1 /* prepare leaf topic related data*/
2 LeafTopic ← {leaf-topic | the leaf topic belongs to
  TopicStructure} ;
3 foreach LeafTopic[i] do
4   Candidates[i] ← {candidate | the candidate mapped
  to LeafTopic[i] in MTServices} ;
5   foreach Candidates[i][j] ∈ Candidates[i] do
6     PropertyValues[i][j] ← {property-value | the
  property value of Candidates[i][j] service in
  Properties} ;
7 /* prepare non-leaf topic related data*/
8 NonLeafTopic ← {non-leaf-topic | the non-leaf topic
  belongs to TopicStructure} ;
9 foreach NonLeafTopic[i] do
10  Constraints[i] ← {constraint | the constraint in
  Requirements mapped to NonLeafTopic[i]} ;
11 /* check potential solution by backtracking strategy */
12 Solution ← ∅; HasChecked ← ∅; i ← 0;
13 /* call backtracking check, and return solution */
14 Solution ← Candidate-Check (i, HasChecked, Solution,
  LeafTopic, Candidates, PropertyValues,
  NonLeafTopic, Constraints);
15 return Solution;

```

ing the backtracking strategy. A record of visit topic (i) and candidates ($HasChecked$) are used in backtracking. Whenever a candidate is selected as one solution ($Solution$), all the constraints will be checked ($Constraints$). If the constraints are satisfied, then the next candidate will be checked, otherwise, the visit record of current topic will be cleared, and the algorithm backtracks to last visited leaf topic.

5. Case Study

We conduct a case study, Japanese-English campus orientation communication, analyze the increase in accuracy achieved and the in-domain resource usage. Accuracy is manually checked using the five-level score (5:All, 4:Most, 3:Much, 2:Little, 1:None) method from DARPA TIDES Project at University of Pennsylvania⁴. For the in-domain resource usage analysis, we analyze two in-domain resources, the dictionary and the parallel texts. We count the number of translated sentences in three situations: parallel text used, dictionary plus machine translator, and only machine translator used.

We analyze the translation accuracy of the sentences that a Japanese student sent to an English student in this ori-

Algorithm 2: Candidate-Check ($i, HasChecked, Solution, LeafTopic, Candidates, PropertyValues, NonLeafTopic, Constraints$)

```

1 /* if check all candidate service, return solution. */
2 if i > length(LeafTopic) then
3   return Solution;
4 /* start from the unchecked candidate, until all candidate
  services of this topic. */
5 j ← HasChecked[i] + 1;
6 for ; j < length(Candidates[i]); j++ do
7   HasChecked[i] ← j;
8   /* check all the requirements, appointed to the
  ancestor non-leaf topics */
9   foreach Constraints[k] do
10    /* if solutions fulfill the related constraints */
11    if LeafTopic[i] is offspring of NonLeafTopic[k]
  && PropertyValues[i][j] satisfy Constraints[k]
  then
12    /* set current checked service as one service in
  solution. */
13    Solution[i] ← Candidates[i][j];
14    /* check next topic. */
15    return Candidate-Check (i+1, HasChecked,
  Solution, LeafTopic, Candidates,
  PropertyValues, NonLeafTopic, Constraints);
16 /* otherwise, resetting checked record of current topic, and
  backtracking to last topic*/
17 HasChecked[i] ← 0;
18 return Candidate-Check (i-1, HasChecked, Solution,
  LeafTopic, Candidates, PropertyValues,
  NonLeafTopic, Constraints);

```

Table 2: Adequacy comparison between Google, J-Server and scenario based composite services

Topic (number of sentences)	Adequacy		
	Google	J-Server	Scenario Based
campus orientation (51)	2.8	2.9	3.8
1 legal procedure (22)	2.3	2.5	4.1
1) office (14)	2.4	2.7	4.2
2) warning (8)	2.1	2.4	4.1
2 student life (29)	3.0	3.1	3.6
1) tuition (9)	2.7	2.9	3.7
2) class schedule (11)	2.9	2.8	3.6
3) health check (9)	2.6	2.7	3.9

entation task. It represents the translation quality that the receiver perceived during the machine translation mediated communication. In the first place, the adequacy of translations provided by Google, J-Server, and scenario base composite service were manually assessed. The average score of sentences associated with one topic was used as the translation accuracy score of this topic (see Table 2). Obviously, scenario based composite service has better adequacy score for each topic. For example, the topic *office*, was expressed by 14 sentences, the average adequacy scores for the three translators are 2.4, 2.7 and 4.2 re-

⁴<http://projects ldc.upenn.edu/tides/translation/transassess04.pdf>

spectively. A comparison of the adequacy of the original topic indicates that the scenario-based composite service has $(3.8-2.9)/2.9=31.0\%$ better adequacy than J-Server.

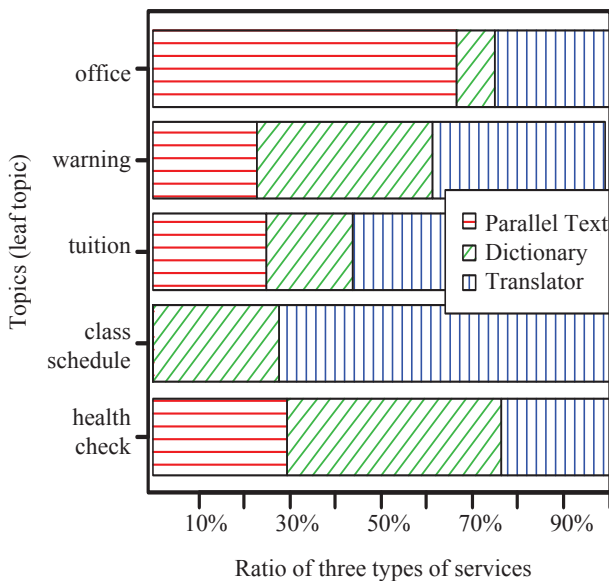


Figure 5: Ratio of the number of sentences translated by *Parallel Text*, *Dictionary* and *Translator* services in each leaf topic

To determine the usage of in-domain resources, we counted the number of sentences translated by *Parallel Text*, *Dictionary* and *Translator* in each leaf topic. The ratios of the number of sentences were determined (see Figure 5). The *Parallel Text* and *Dictionary* services, wrapped versions of in-domain resources, improved the translation accuracy. For example, in the topic *office*, the contribution of *Parallel Text* is obvious (see Figure 5), the scenario-based composite service has much higher adequacy than Google or J-Server (see Table 2).

•SOURCE
Sentence (from warning topic): 好奇心や軽い気持ちで大
麻等薬物には絶対に手を出さないようにしてください。

•TRANSLATION

1. Google: To drugs such as marijuana, please do not lay a hand on an absolute curiosity and with a light heart. (adequacy: 3)
2. J-Server: Please make sure that the cannabis won't be involved in medicine by any means by curiosity and light feeling. (adequacy: 3)
3. Scenario Based Translation
 - 1) Candidate (crime-disaster-para): DO NOT USE drugs such as marijuana under any circumstance. (adequacy: 4)
 - 2) Candidate Selection: (crime-disaster-para), if parallel text service has result, it will be selected firstly.

Figure 6: An example of adequacy promotion by integrating *parallel text*, in the *warning* topic

•SOURCE
Sentence (from health check topic): 大学病院は、医学部構内の東大路通側に存在し、病院構内と呼ばれている。

•TRANSLATION

1. Google: University Hospital is present at the side of the campus through Higashioji School of Medicine, has been called the hospital campus. (adequacy: 2)
2. J-Server: A university hospital exists in the University of Tokyo Rotsu side of the medical school premises, and is called hospital premises. (adequacy: 2)
3. Scenario Based Translation
 - 1) Candidate (medic-para): *NO RESULT*.
 - 2) Candidate (google + medic-dict + city-dict): University Hospital, exist at the site of Higashioji street on the premises of Faculty of Medicine, has been called the hospital campus. (adequacy: 4)
 - 3) Candidate (j-server + medic-dict + city-dict): A university hospital exists in the Higashioji street side of the medical school premises, and is called hospital premises. (adequacy: 4)
 - 4) Candidate Selection: (google + medic-dict + city-dict), it is the first candidate satisfied.

Figure 7: An example of adequacy promotion by integrating *dictionary*, in the *health check* topic

Two concrete examples are described here, see Figure 6, Figure 7 for *warning* topic and *health_check* topic respectively (see Figure 4). The former requires the use of *parallel text*. Here *crime-disaster-para* parallel text service provides the Japanese-English sentence pairs. The Japanese sender in orientation task can use it for communication, and the English sentence will be sent to the English receiver. Obviously, parallel texts have higher adequacy of than *Google* or *J-Server* outputs (see Figure 6). The latter shows how dictionary can raise the adequacy of translation. This is because the street name *Higashioji street* name is not one word for either *Google* or *J-Server* translation. If the *city-dict* is used, it will not be improperly cut, and the other sentence parts are not mistranslated. The results of multi-hop composition service demonstrate its superior adequacy compared to either *Google* composition or *J-Server* composition (see Figure 7).

Note that these topics are stored in the topic database. When planning another related communication task, we can easily find and reuse useful topics; for example, topic *office* in the “campus orientation” task can be used in other similar tasks like “school history introduction”.

6. Conclusion

We have provided a simple configurable system in a task-oriented machine translation mediated communication, which realizes a light-weight approach to integrating in-domain resources for raising translation accuracy. Based on existing techniques (topic detection and QoS-based machine translation service selection and composition), we

conducted scenario-based analysis, designed the architecture for our system, and provided a declarative language for the communication task designer. We also described a simple case to explain how it works.

By using our architecture, it is easy to take advantage of existing MT services on the Language Grid platform, refer to and reuse already configured topics and MT services, and automatically select proper MT services.

By using the simple SDL language, a task designer can easily combine the topics of communication tasks with in-domain resources. Language Grid provides tools to conveniently wrap in-domain resources into dictionary service and parallel text service. With the services of in-domain resources available, and the SDL program script of composition scenario from the task designer, QoS based MT service composition will be executed yielding more accurate translation results.

Finally, our case study of a campus orientation task showed translation accuracy can be raised by up to 31% and confirmed the convenient reuse of components in new machine translation mediated communication tasks.

Acknowledgement

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

7. References

- James Allan, editor. 2002. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, Norwell, MA, USA.
- Srinivas Bangalore, Giuseppe Di Fabbrizio, and Amanda Stent. 2006. Learning the structure of task-driven human-human dialogs. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, ACL-44*, pages 201–208, Stroudsburg, PA, USA. Association for Computational Linguistics.
- 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.
- Yohei Murakami, Donghui Lin, Masahiro Tanaka, Tako Nakaguchi, and Toru Ishida. 2010. Language service management with the language grid. In *LREC*, pages 3526–3531.
- Franz Josef Och, Christoph Tillmann, Hermann Ney, and Lehrstuhl für Informatik. 1999. Improved alignment models for statistical machine translation. In *University of Maryland, College Park, MD*, pages 20–28.
- Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '06*, pages 35–42, New York, NY, USA. ACM.
- H. Wu, H. Wang, and C. Zong. 2008. Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In *COLING*, pages 993–1000.
- Naomi Yamashita and Toru Ishida. 2006. Automatic prediction of misconceptions in multilingual computer-mediated communication. In *IUI06*.
- T. Yu, Y. Zhang, and K. Lin. 2007. Efficient algorithms for web services selection with end-to-end qos constraints. *ACM TWEB*, 1.