Speaking Louder than Words with Pictures Across Languages

Andrew Finch, Wei Song,
Kumiko Tanaka-Ishii, Eiichiro Sumita

■ *In this article, we investigate the possibility of cross-language communication using a synergy of words and pictures on mobile devices. On the one hand, communicating with only pictures is in itself a very powerful strategy, but is limited in expressiveness. On the other hand, words can express everything you could wish to say, but they are cumbersome to work with on mobile devices and need to be translated in order for their meaning to be understood. Automatic translations can contain errors that pervert the communication process, and this may undermine the users' confidence when expressing themselves across language barriers. Our idea is to create a user interface for cross-language communication that uses pictures as the primary mode of input, and words to express the detailed meaning. This interface creates a visual process of communication that occurs on two heterogeneous channels that can support each other. We implemented this user interface as an application on the Apple iPad tablet and performed a set of experiments to determine its usefulness as a translation aid for travellers*

It has been said that an old Chinese proverb placed a value of 10,000 words on a single picture, and a similar Japanese proverb devalues this to only 100 words. In most languages the current consensus seems to be that a picture is worth 1000 words. Whatever the true worth in words a good picture is capable of conveying (sometimes quite complex) meaning clearly and without the need for language. Show a picture of an elephant to speakers of two different languages, and most likely they will both understand exactly what it means. A picture can in effect ground the meaning to an object or concept in the real world and act as a convenient bridge over language barriers.

Picture Books

Our idea originally stemmed from the rise in popularity of picture book translation aids in Japan. These books are a modern interpretation of the traditional phrase book, and they improve on it by adding image annotations and allowing users to compose their own phrases by combining fragments of sentences that are found on the same page together. For example, figure 1 illustrates the process of communication using a picture book. The process is simple: the user of the book simply points at pictures or text on the pages of the book, in a particular order. In this case let's assume the user is a Japanese person wanting to communicate with an English speaker. The user first points to "I

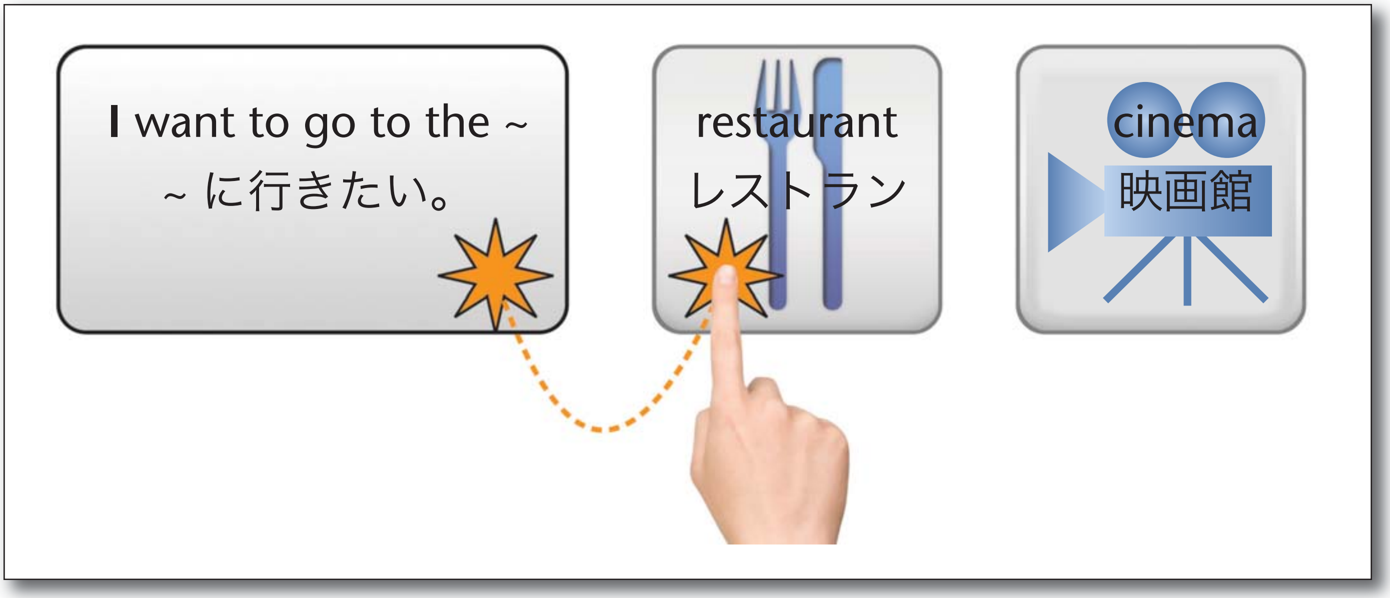


Figure 1. Communication Using a Picture Book.

want to go to the ~" Here the ~ is a placeholder for a number of possible filler items that appear on the same page. In this example we give two possible filler items: restaurant and cinema; the user chooses restaurant.

The picture book is a powerful idea because it is easy for users to understand the communication process and because the use of pictures to support the words in the book will not only aid the process of visual search for phrases but also assist the communication process. However, the picture book has limits by virtue of its being a book, namely: the number of pictures contained in the book is limited; complex expressions cannot be constructed; the search for the appropriate pictures can be laborious; and pictures are only designed to be combined with pictures on the same page. Combining pictures with others not designed to be used with them may not make sense.

The aim of our research was to try to find a way to create a process of visual communication in a similar form to the picture books but within the framework of an intelligent interactive information system capable of mediating to facilitate the communication.

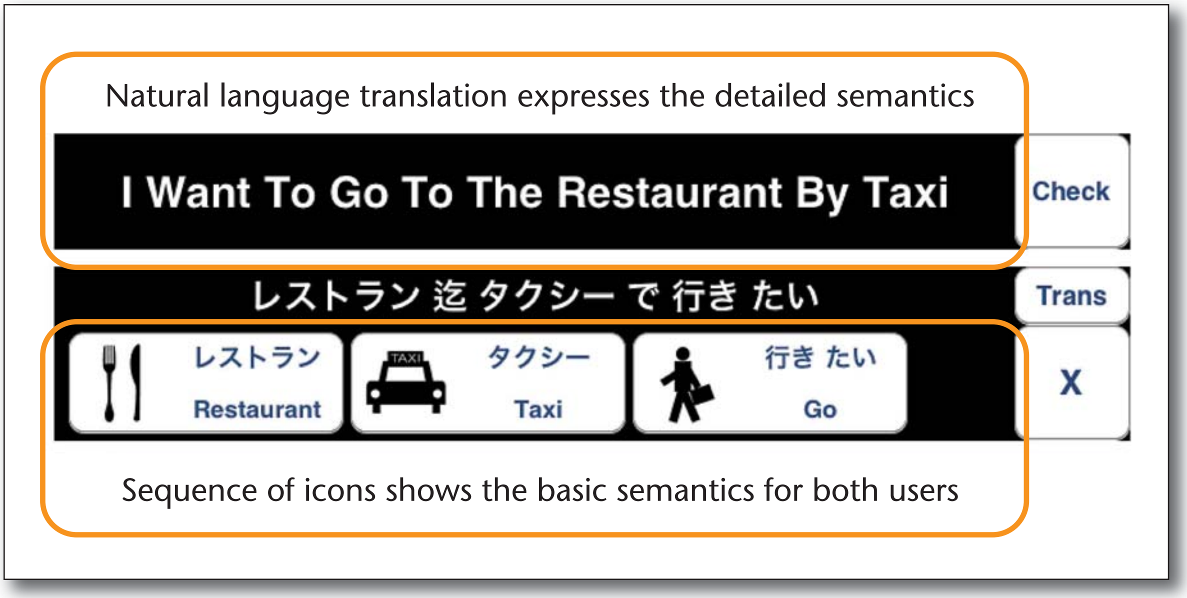
Machine Translation

If a machine is going to lend a hand in the communication process between two people, perhaps the most obvious way it can contribute is by providing an automatic translation of the natural language expressing what is intended to be communicated. Machine-translation (MT) systems already exist on mobile devices, for example the VoiceTra and TextTra mobile applications that take their input from speech or from text, respectively. Machine translation however is also not without problems. First, neither of the two input methods previously described are perfect for use on mobile devices. Textual input is very cumbersome on small mobile devices, and speech-recognition systems frequently make errors that are hard for the users to correct. Second, the MT systems themselves can make errors. Sometimes nonsense is generated, or if the MT system is particularly skilful very fluent output can be produced that carries totally the wrong meaning. The users may have no idea what has been communicated to the other party, and in some cases users may believe they understand perfectly what was expressed, when in fact they are gravely mistaken.

Our Idea

Our idea is a very simple one: use pictures as the user input method. The users should be able to input the gist of what they wish to say in the form of a sequence of picture icons and then let the machine work out what they intend to express and provide a translation of this in the other language.

Our system is called picoTrans (picture icon translator) and in essence the users communicate through the system by pointing at pictures in much the same way they would do with a picture book. As the users tap icons, they appear in sequence in a box on the user interface, and at the same time the system generates natural language for them to express their full meaning. A user can interact with the system to modify or refine the



Natural language translation expresses the detailed semantics

I Want To Go To The Restaurant By Taxi

レストラン 迄 タクシー で 行き たい

Sequence of icons shows the basic semantics for both users

Figure 2. Generating Natural Language from Sequences of Picture Icons on the picoTrans User Interface for Japanese Input.

natural language that's produced. When the natural language expresses what the user intends to say, the user can have the machine translate it for them.

When the authors originally came up with this idea, it appeared to be a very exciting new approach to user input, but at the same time seemed to be a risky endeavour that might end in failure. The prototype user interface would require a lot of effort to develop, and there was little evidence to suggest it might work at all. Fortunately, the excitement outweighed the risk; the risk was taken and the system was developed but within the limited role of an intelligent phrase book for travellers where the linguistic challenges are not so great as for open-domain natural language. This article will follow the development of our prototype system and look at how well we succeeded in realizing our idea. We start with a simple example to illustrate how one might use our system.

An Example Scenario

In this scenario a user fluent in his or her own native language (we will call it the *source language* from now on) is attempting to communicate with another user fluent in a target language. The two users do not share a common language in which they are able to communicate. Figure 2 shows the way that picoTrans could be used to communicate the same expression as in the picture book example in figure 1. With picoTrans the user also points to a sequence of icons and the icons appear in sequence on the display; however, in our case the sequence of icons is maintained for the users to see and interact with if necessary. When the input of the icon sequence is complete, the system generates the full sentence in the source language automatically, which is then translated by the machine-translation software and displayed on the screen together with the icon sequence.

Advantages over Existing Approaches

The main benefit of our system over a picture book stems from the fact that the number of picture icons contained in our system is potentially huge, and large or unbounded classes such as numerical expressions or place names can be handled. In addition picture icons in the picoTrans system can be combined arbitrarily, increasing expressiveness, and the application can help the user to find the appropriate picture icon, either by lexical search or predictive entry. Last and certainly not least, the users are able to interact with the device.

The benefits from taking our approach over using only machine translation are twofold. First, the semantics conveyed by the picture icons themselves can often be enough to communicate the entire meaning of what was needed to be expressed, and for cases where they are insufficient, they nevertheless constitute an independent channel of communication between the parties that can be used as a sanity check for the machine-translation system. Second, the process of pointing and clicking at icons is a very natural input method suitable for the current generation of touchscreen mobile phones and tablets.

There are other benefits that we will come to later in the article, but there are also some serious concerns. The most basic concern is our original worry: will this work at all? Is it possible to gener-

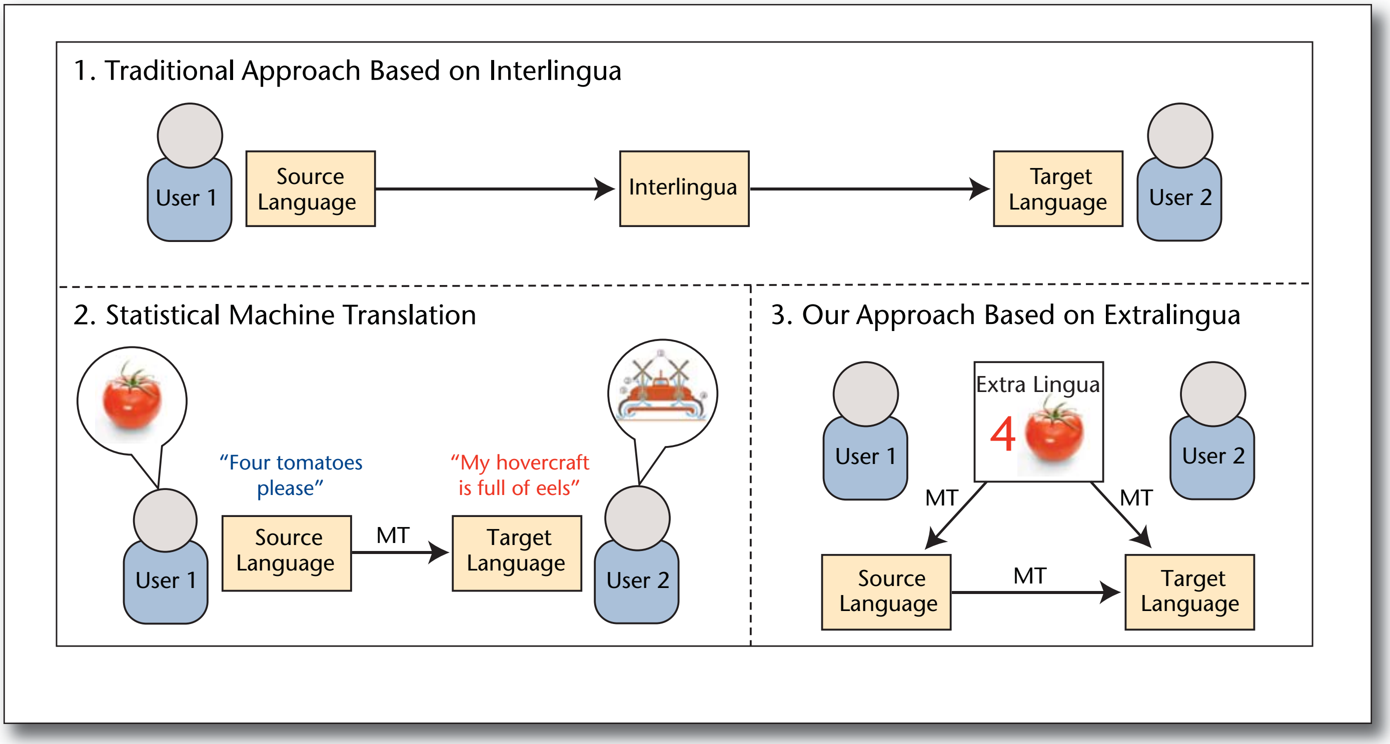


Figure 3. Various Translation Channels Among Communicators Using Machine Translation

ate natural language from a sequence of vague concepts lacking the syntactic glue that binds them together and moderates how they combine together to impart the complete meaning of an expression? In general, the answer to this may well be no, but in this research we have chosen to work within a useful but restricted domain, that of travel conversation. Working within this domain means that typically sentences are shorter (in the English part of our basic travel expression corpus [BTEC] the average sentence length is around seven words), and the domain of discourse is quite narrow, allowing the machine to predict the user's intended meaning more easily than if the domain of discourse was totally open.

Adding a Second Channel of Communcation

A key feature of our approach is that it opens up a second heterogeneous communication channel, which we will call an extralingua because it is a channel for extralinguistic communication and because of its relationship to the term *interlingua*. Figure 3 shows two methods for conducting machine translation that have been studied in the MT domain. First (box 1 in figure 3), translation can be performed through an interlingua, an intermediate language placed in between the source and the target language. When the interlingua is a natural language, the communication channel can be a concatenation of two MT systems: the first from source to interlingua, the second from interlingua to target (Paul et al. 2009). Second (box 2 in figure 3) is a process of direct translation from source to target. This method can reflect most current approaches to machine translation.

In contrast, our approach (box 3 in figure 3) uses an extralingua, which is exposed to both communicators. Both users are able to interact with the extralingua, assisted by three MT systems: the first between the extralingua and the source language, the second between the source language and the target language, and the third between the extralingua and the target language. The approach is analogous to the interlingual approach, with natural language acting as the interlingua, and with the source language (sequences of pictures) being exposed and comprehensible to both users. The reader might wonder why MT is needed at all if such an extralingua exists. This is in fact the point: the communicators lack a common language through which they can communicate, and so far we have only considered ways to bridge this gap by using just a single MT channel. However, in many circumstances, the communicators do have other means for communication, such as images, signs, and actions, and will often use them when other means fail. This other mode of communication can be adopted in parallel independently of the MT channel, but our idea is to tightly couple a second communication channel directly into a machine-translation system. Note that the pictures in the figure appear to be contributing to the machine-translation process itself. In fact they do make a contribution, in their ability to regularize the language being translated, albeit indirectly.

There are multiple advantages to taking this approach. First and above all is to improve the quality of communication between users. Adopting an extralingua allows the users to communicate through two heterogeneous channels. Since we cannot expect MT output to be perfect, having a second independent mode of communication to reinforce or contradict the MT system will lead to a greater mutual understanding. In the figure, of course no modern MT system worth its salt would make such a grand translation error for such a simple sentence, but note that *tomatos* is misspelled and an MT system would not know how to translate it. This brings us to the second advantage: since the user input process is mediated by the user interface, the input can be regularized. Icons cannot be misspelled and the regularization can reduce the number of possible forms the language can take, thereby lowering the complexity of the translation task itself. Icons can even stand for groups of words (carrying their correct translation), and their translations can be passed through the translation process as a single unit without fear of their being translated independently and then scattered in the translation.

Related Work

The basic premise of our user interface, that sequences of images can convey a meaningful amount of information, is directly supported by an interesting study into the effectiveness of using pictures to communicate simple sentences across language barriers (Mihalcea and Leong 2008). Using human adequacy scores as a measure, Mihalcea and Leong found that around 76 percent of the information could be transferred using only a pictorial representation. Furthermore, the Talking Mats project (Murphy and Cameron 2008) has developed a communication framework consisting of sets of pictures attached to mats to enable people with communication difficulties to communicate. Ma and Cook (2009) conducted a study of various ways, including images and animations, in which verbs could be visually represented, finding that the efficacy of the method depended on the nature of the verb being represented. In research into collaborative translation by monolingual users, Hu, Bederson, and Resnik (2010) propose an iterative translation scheme where users search for images or weblinks that can be used to annotate sections of text to make its meaning more explicit to another user who does not share the same language. In other related work, Zhu et al. (2007), demonstrate the usefulness of a text-to-picture transduction process (essentially the converse of our icon-to-text generation process) as a way of automatically expressing the gist of some text in the form of images.

The preceding sections have hopefully imparted the general idea behind our approach; we now move on to describe the user interface in detail.

User Interaction Process

A flow diagram of the operation of the user interface operation is given in figure 4, and a diagram of the user interface of the first-generation picoTrans system for a Japanese user in full is shown in figure 5. The user interface elements in figure 5 are labeled with circled numbers, which are referenced in the text and also in figures 4 and 6.

This system used a categorical icon input method, a simple, natural extension of the picture book communication process adapted for use on mobile devices. We will start by describing this method, but there are other possibilities for icon input that will be introduced later.

In brief, we allow the users to input what they wish to express as a sequence of bilingually annotated icons. This is in essence the same idea as the picture book. Users can switch the user interface into their own language by pressing the User Interface Language Toggle Button (#12 in figure 5).

The simplest form of the translation process proceeds as follows:

(1) The users select a category for the concept they wish to express

(2) The users select a subcategory

(3) The users choose the first icon in the sequence
   (a) Go to (1) to select another icon for the sequence
   (b) The icon sequence is complete, and the corresponding source sentence is acceptable. Continue to step (4)

(4) The users click the Trans button

(5) The translation appears in the translation text area

A user interface flow diagram (storyboard) for the full user interface is given in figure 4. In this figure the boxes represent the user interface elements, and the links represent relationships between them. Labels on the links denote specific user actions associated with transitions between two interface elements. Numbers in parentheses in the figure correspond to the user interface elements associated with the steps in the list above.

Briefly, there are three kinds of interaction required for the user: (1) selection of icons; (2) selection and refinement of the source sentence; and (3) viewing the output translation. Each of these steps is explained in more detail in the following sections.

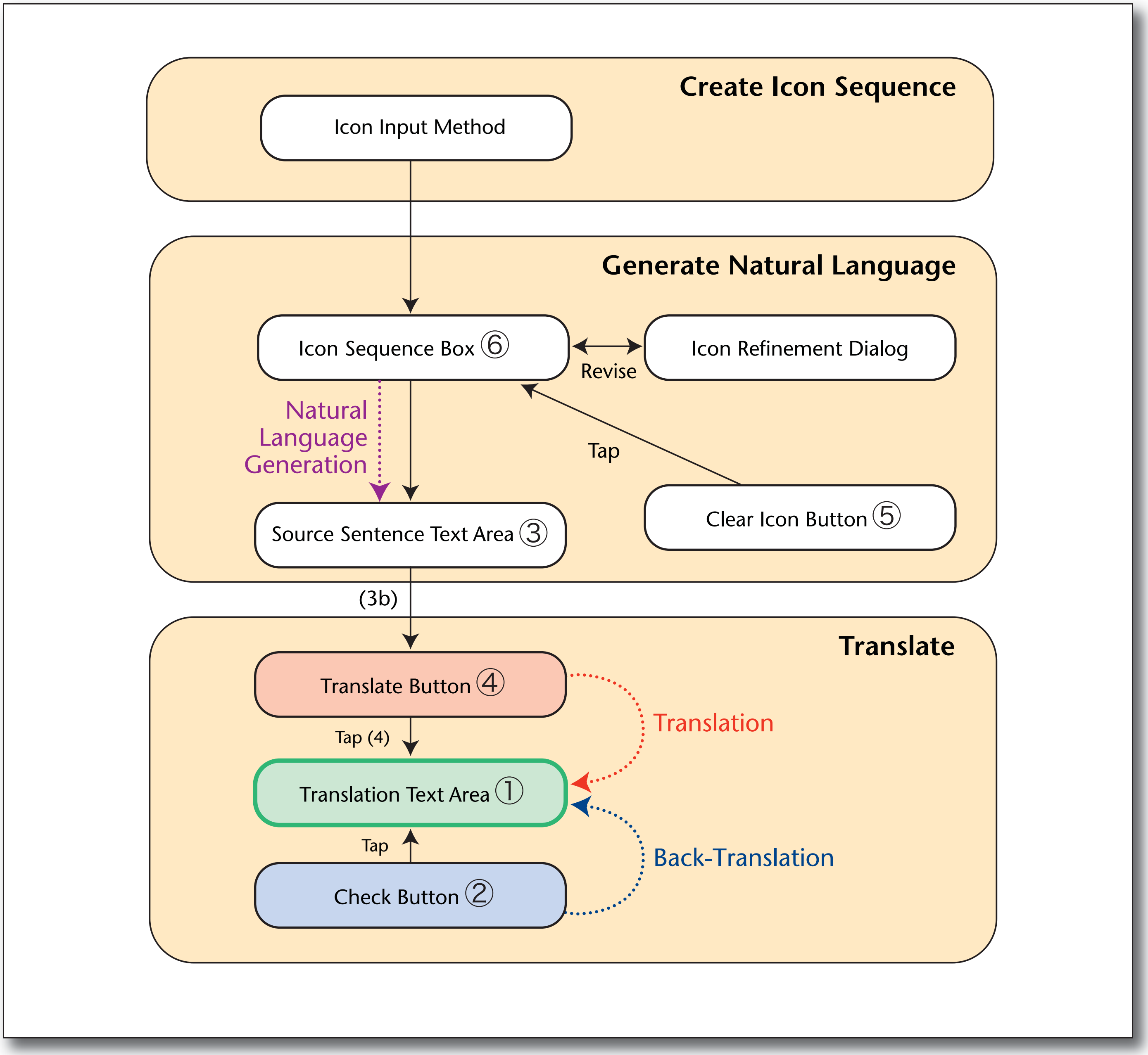


Figure 4. A Flow Diagram for the User Interface.

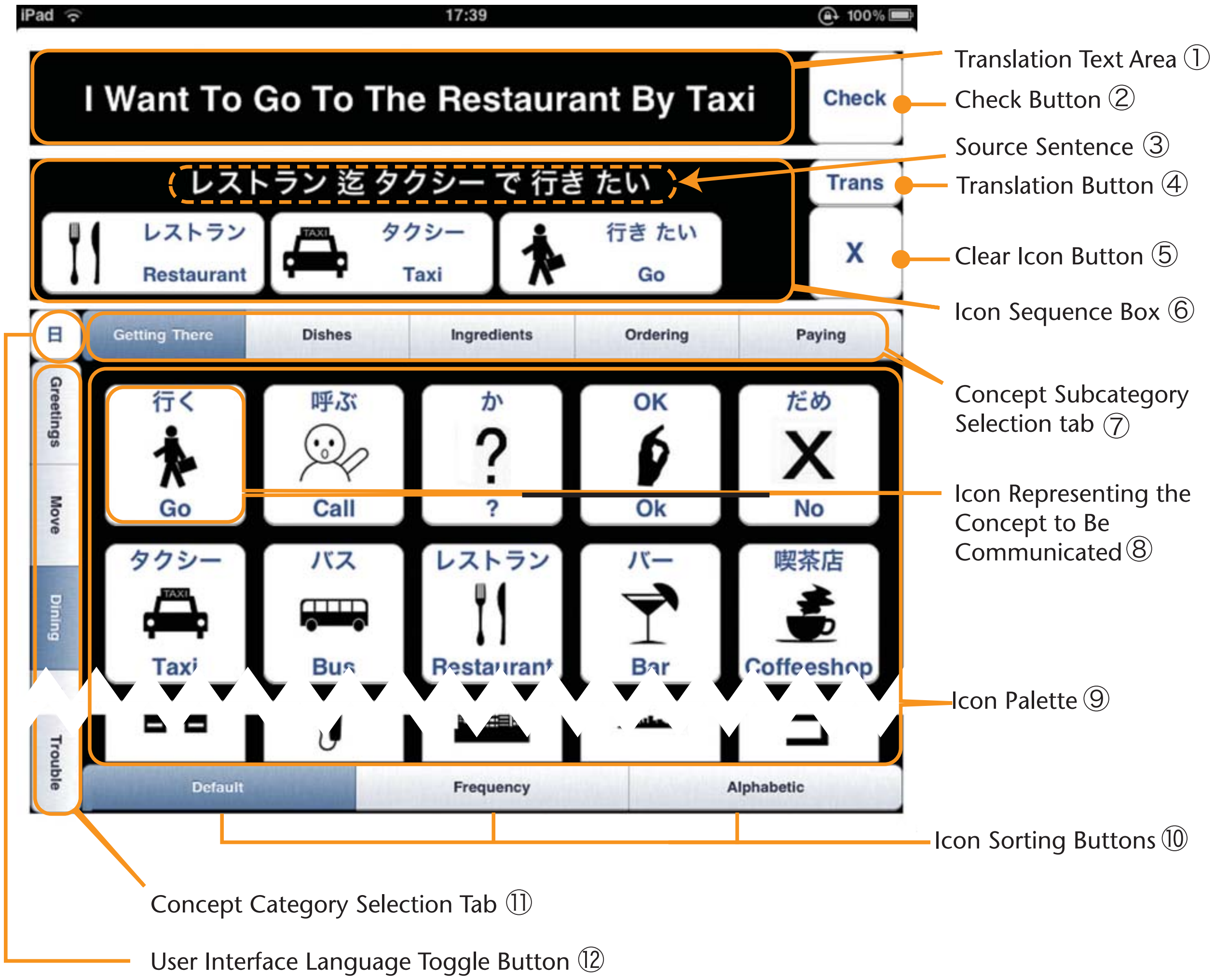
The numbers in parentheses represent the steps in the translation process described in the User Interaction Process section. The circled numbers refer to the numbers used to reference the user interface elements shown in figure 5.

Creating a Sequence of Picture Icons

The communication process commences with icon selection, which takes place in the icon palette area of the interface (labeled as #9 in figure 5). This contains the set of icons available immediately to the user for translation. The icons are arranged in a rectangular array and can be organized in default order, frequency order, or alphabetical order by pressing the corresponding Icon Sorting Button (#10 in figure 5).

Categorical Icon Selection

To winnow the choice of icons, the user first selects the icon category in the Icon Category Tab. Let's say the user selects the category *Dining*. The interface will select the default subcategory from the subcategories for that icon. After the category has been selected the user then either accepts the default subcategory selection or selects the appro-



I Want To Go To The Restaurant By Taxi

レストラン 迄 タクシー で 行き たい

Figure 5. The Annotated User Interface for the picoTrans System.

priate subcategory on the subcategory tab. Let us assume the user selects the subcategory *Getting there*. Once the subcategory has been chosen the application displays the icons for that particular subcategory in an order described in the next section. In this example, the user would see icons for Taxi, Go, and so on.

We allow the user to order the icons in two other ways, the first being in alphabetical order, which allows an icon whose name is known to be quickly found, and the second being by expected frequency of use. We used an empirical corpus-based approach to achieve this. Each of the icons is associated with a particular content word that appears in the corpus of text used to train the machine-translation system. We use the frequency of occurrence in the corpus of travel expressions of this content word as an indicator of how frequently this concept is likely to be used in real travel dialogues and consequently offer the user the option of ordering by this frequency.

We extend the icon ordering approach described above to also manage the ordering process of the categories and subcategories. Since each category and subcategory represents a set of icons, for which occurrence counts in a corpus of text are known, we can estimate the a priori probability of the user selecting an icon from a given category/subcategory and use these probabilities as a way of ordering the categories/subcategories in the tabs. The cate-

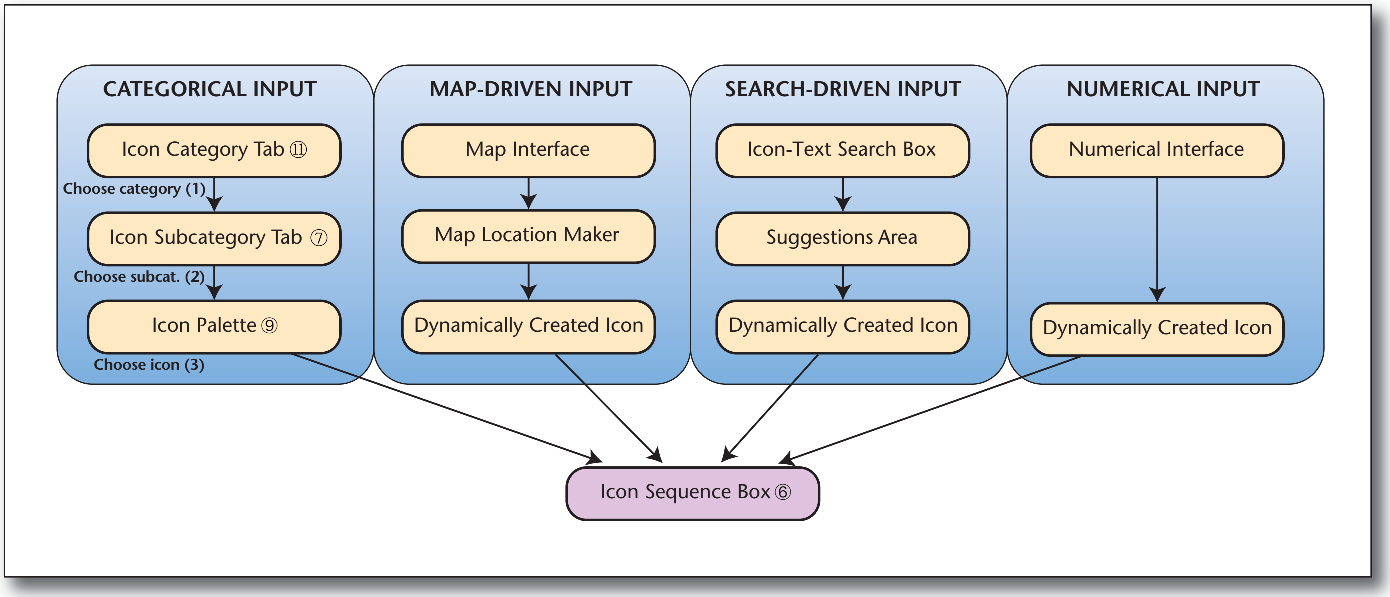


Figure 6. Various Methods for Icon Input in the picoTrans System.

gories and subcategories are ordered by their expected frequency of use.

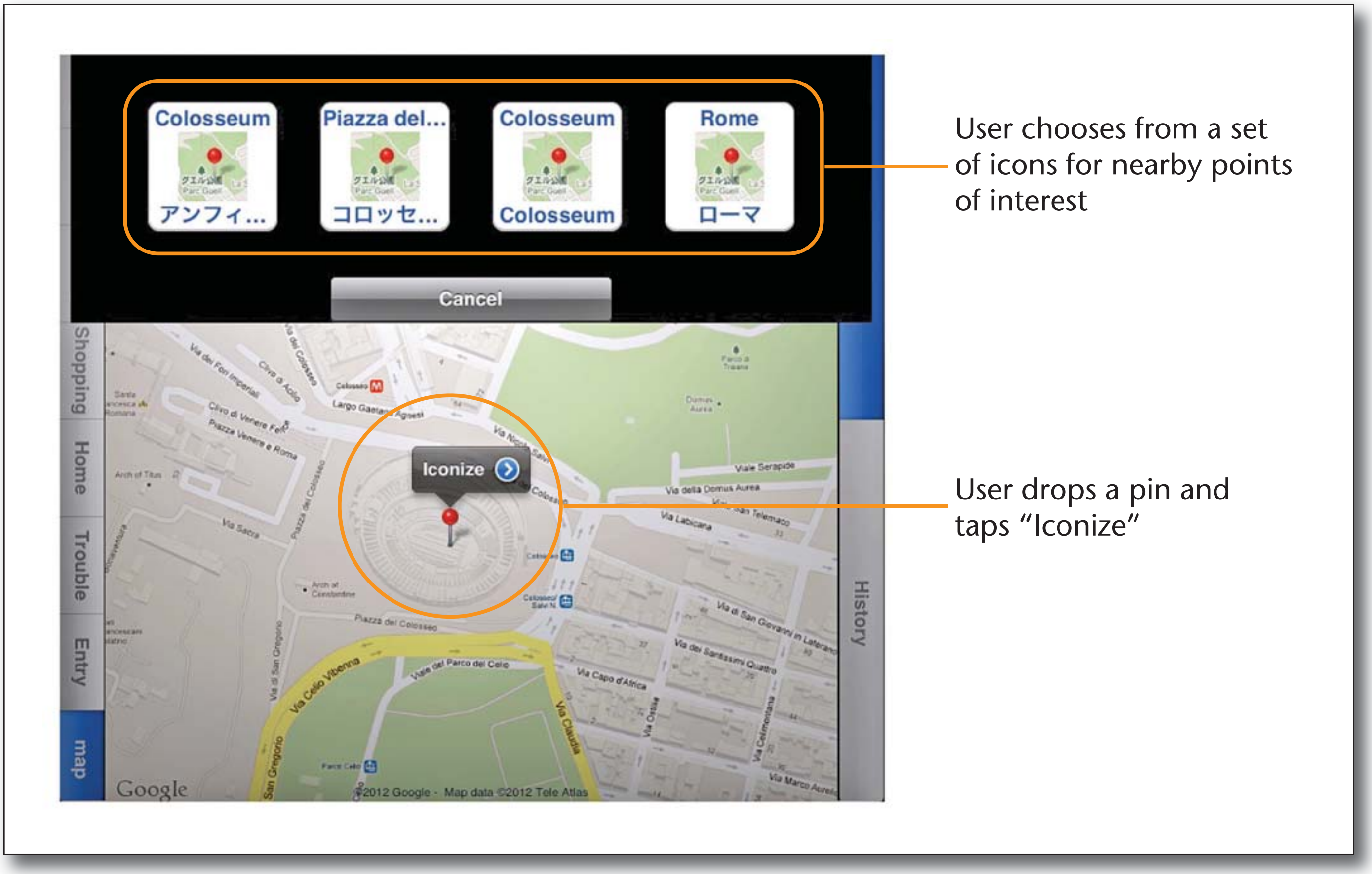
Icon Search

The initial versions of the picoTrans system reported in Song et al. (2011) and Finch et al. (2011) were deliberately based around an icon-only interface in order to investigate the implications of this approach. However in order to increase the number of icons that can be input, it is possible to relax this constraint and allow textual input alongside the original interface. As a result of the feedback received from our studies of user experience we believe it is beneficial to supplement the category/subcategory icon search process, which is useful for icons in commonly used patterns, with a more powerful icon search interface capable of rapidly finding more unusual icons that are not immediately offered by the interface to the user. We base this search on a simple extension of the techniques employed in predictive text input: the users type the first few characters of the word or icon they are interested in inputting, and the system displays all icons in the icon set and all words in its dictionaries that have the characters entered as a prefix. You can see this part of the interface in action later in this article in figure 10.

As more characters are provided by the user, the set of possible icons and words from the bilingual dictionary decreases until the sets are sufficiently small that the user is able to select an icon/word. The idea is the same as predictive text entry and carries with it the same kinds of benefits in terms of input efficiency, but in this input method the selection process is over both words and icons simultaneously. When the user chooses an icon, it can be inserted directly into the icon sequence. If the user chooses an entry from a bilingual dictionary for which there is no icon, either a generic icon can be created for this entry and used, or the icon can be annotated with an image from the web in a process described later in this article.

Numerical Expresions

Numerical expressions pose a major problem for machine translation due to the sheer variety of expressions that could be used to input them. The expressions “203,” “2 0 3,” “2-0-3,” “two oh three,” “two hundred and three” are just a few of the many possible ways a user could input the number 203 into a text-based machine-translation system. Furthermore, even if the number is known, its type may not be; it could be a room number, a phone number, a time, and so on, and this type can affect the way in which the number needs to be translated (1973 as a year is expressed as “nineteen seventy three,” as a price it would be expressed as “one thousand nine hundred and seventy three,” and as a telephone extension “one nine seven three”). In a recent extension to the picoTrans system, we allow the input of numerical expressions by dynamically creating typed icons. Dates and times for example can be entered using standard user interface components such as the UIDatePicker element in the current iOS API. In this manner, is it clear to both parties exactly what date and time is intended, and furthermore the underlying machine-translation system receives



User chooses from a set of icons for nearby points of interest

User drops a pin and taps "Iconize"

Figure 7. Selection of a Point of Interest on a Map, from Which to Create an Icon to Be Used in the Current Expression.

not only an unambiguous numerical value for the numerical expression but also its type. It is also possible that the users may interact with the icons on the device, for example to negotiate a price or agree on the time for a meeting.

A common strategy for translating numerical expressions in a statistical machine-translation system is to attempt to automatically identify them in the input and then use a rule-based approach to translate that part of the sentence, integrating it with the statistically translated part of the sentence later in the process. In our approach, the user is identifying the numerical expressions as part of the input process, and the rule-based translation of the expression will be more accurate if the expression's type is known.

Map-Driven Icon Selection

For travelers, perhaps the most important class of named entity is the class of place names. This class includes hotels, restaurants, sightseeing spots, railway stations and so on, and including them all explicitly as individual icons would result in an unmanageably large icon set. To address this issue, in the picoTrans system we handle this class of named entity using a dynamic icon creation scheme driven by a bespoke visual map-based interface designed specifically for this task. The basic idea behind this method of input is simple: users point directly to a location on a map, and the system will construct a new icon based on the closest point of interest on the map. The Google maps we use in the picoTrans system have bilingually annotated points of interest, so not only is it possible for us to annotate the icon we construct bilingually, but it is also possible for us to pass the correct translation for this named entity directly to the machine-translation system, guaranteeing that it is translated correctly.

In figure 7, the user has just dropped a pin on the map and the system has dynamically created a set of icons for the points of interest around the indicated location. This icon can be inserted into the icon sequence for the current expression, and behaves just like any other icon. This example is one of the most powerful illustrations of the additional clarity images can contribute to ground terms in the communication process. In a text-only machine-translation system, the system at best can give the other party the correct textual expression indicating the location being discussed and at worst will result in a version of the place name corrupted by the translation process. The words in multiword expressions can become separated and their translations may become distributed over the target sentence. In the picoTrans system, the other party will see the intended location clearly on a map (that he or she can interact with if necessary to get his or her bearings), and in addition to this will receive both a textual translation analogous to the text-only method described earlier (but with the named entity correctly translated).

Furthermore, creating icons for places in this manner has the advantage over simply typing them into the system in that unlike a sequence of words representing a named entity in text, in this case the position and extent in the input sequence of the entity is known, and so it its type. That is, we know this entity should be translated as a single unit, we know its translation, and also we know it is a place. Knowing the type of a named entity can be of considerable help when training a machine-translation system, since the system may be trained with tokens representing the types as proxies for the word sequences representing the named entities themselves: for example "How do I get to <PLACE>?" would replace "How do I get to Mornington Crescent?" This will have a beneficial effect on statistical models that suffer from data sparseness when trained with lexical data.

Generating Natural Language from Icons

The task of transforming our icon sequence into the full source sentence is quite similar to the task of transliteration generation. The task here is to monotonically transduce from a sequence of icons into a sequence of words, whereas transliteration is a process of monotonic transduction from one sequence of graphemes into another. Our approach therefore is able to borrow from existing techniques in the transliteration field for sequence-to-sequence transduction.

Transliteration generation can be performed using a phrase-based statistical machine-translation framework using a monotonic constraint on the word-reordering process (Finch and Sumita 2008). We adopt a similar approach but use a Bayesian bilingual alignment technique to derive the translation model (Finch and Sumita 2010; Finch, Dixon, and Sumita 2012). The data for training the system was created by word deletion from a monolingual corpus of travel expressions, and deletion proceeds based on the parts of speech of the words. In essence, function words representing the syntactic sugar that binds the sentence together are deleted leaving sequences of content words that are proxies for the icons in our system.

Icon Refinement

The icons on the icon palette can be tapped to add them to the end of the sequence of icons to be translated. The icons when tapped have two types of behavior that depends upon the ambiguity of the underlying content word used to represent them on the interface. By ambiguity here we mean that the icon can give rise to several related forms in the natural language we will generate; for example, a verb like go can take several forms. For an example see figure 8.

If the underlying content word is unambiguous, the user simply chooses an icon by tapping it. The icon changes appearance briefly as feedback to the user that it has been tapped, and a smaller version of the icon appears in the Icon Sequence Box #6, on the end of the sequence of icons already present (if any).

If the underlying content word is ambiguous, the user also chooses the icon by tapping it. The icon changes appearance briefly as feedback to the user that it has been tapped, and a disambiguation dialogue box appears on the screen offering the user several choices for the precise meaning of the icon; this is shown in figure 8. Often these ambiguous icons are verbs, and the choices are various possibilities for the verb's form. For example the icon representing the concept *go*, might pop up a dialogue asking the user to choose among *will go*, *want to go*, *will not go*, *went*, and so on. Once the users have chosen their preferred semantics for the icon, a smaller version of the icon appears on the end of the sequence of icons already present (if any) in the Icon Sequence Box (#6 in figure 5).

The Icon Sequence Box contains the sequence of icons used to express the source sentence to be translated. As the user chooses icons from the icon palette, they appear in sequence from left to right in the Icon Sequence Box. The Icon Sequence Box contains two buttons, the Translation button used to initiate translation and the Clear Icon button. When the user taps the Clear Icon button (#5 in figure 5), all icons are removed from the Icon Sequence Box. The user can remove individual icons from the sequence by swiping them.

Source Sentence Selection and Refinement

The process of refinement of the semantics of the source sentence is performed using the refinement dialogue box. Once the icon sequence has been chosen, the user is shown the system's suggested source sentence for the sequence (#3 in figure 5).

The system has been designed so that this source sentence is most likely to be what the user would say. Ideally this would be driven by the context of the expression within the dialogue that the user is having, but in our current system it is based on a corpus-based model driven by the frequency of

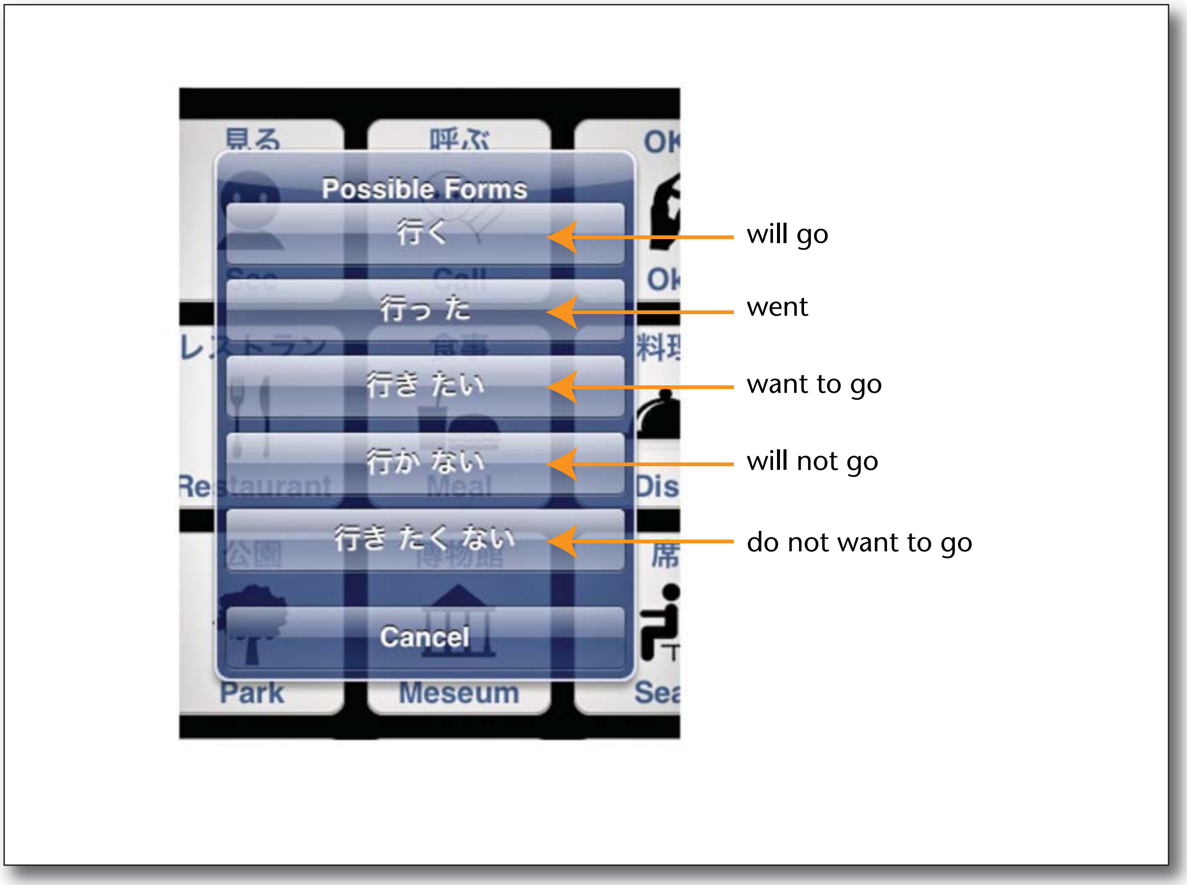


Figure 8. The Icon Semantics Disambiguation Dialogue.

occurrence of the components used to build the source sentence.

One aim of the application is to predict the source sentence correctly for the user; however, this will not always be possible and the user interface allows the user to interact to refine the semantics of the icon sequence. To do this, the user simply taps on the icon in the Icon Sequence Box #6 in figure 5) that marks the last (rightmost) icon in the sequence of icons that requires modification. The application highlights this icon and pops up a dialogue box similar to the icon disambiguation dialogue box in figure 8 that allows the user to choose the precise semantics for the icon sequence up to and including the selected icon. The choices presented have been extracted from the source language corpus, and are ordered in terms of probability assigned by the natural language generation engine.

Translation

The users tap the Translation button (#4 in figure 5) when they are satisfied that the source sentence generated from the (possibly refined) sequence of icons they have selected represents the expression they wish to communicate to the other party. The application then sends the source sentence to the machine-translation server for translation. The translation in the target language appears in the Translation Text Area (#1 in figure 5), when the user taps the Translate button above. The translated result is shown to the person whose mother tongue is the target language. The user and his or her couterpart communicate through viewing the translated result (together with the icon sequence).

System Architecture

The translation and back-translation for the picoTrans system are performed by calling an API that communicates with machine-translation servers on the Internet. The language generation is performed by the OCTAVIAN machine-translation decoder running locally on the device. We have found the mobile device more than powerful enough to perform this task. In future versions of the system we believe it should be possible to use compact models that allow us to embed the whole system on the device, enabling it to operate without the need for a network connection.

Problems and Solutions

During the development of the picoTrans prototype, we encountered a number of obstacles in both the design and operation of the interface, some of which were unexpected. This section details some of these problems and the solutions we devised to deal with them.

Occlusion

One of the major problems affecting user input on mobile devices is occlusion: the user's hands can obscure parts of the user interface, and studies have shown that this can lead to a considerable decrease in performance (Vogel and Baudisch 2007; Baur, Boring, and Butz 2010). With this in mind we have designed the user interface such that the interface elements avoid issues with occlusion. The general layout of the application uses the pocket calculator as a metaphor. The interface groups its interface elements into three distinct levels, each level being physically higher on the device itself as well as performing a higher-level function. The functions at each level correspond directly to the three phrases of user interaction: the selection of icons; the selection and refinement of the source sentence; and the display of the translation results.

Icon selection functions are performed at the bottom of the device. The area involved with icon sequence display, editing, and refinement is immediately above the icon selection area. While interacting with the icon sequence, the user's hands will obscure parts of the icon selection area, but typically the user will not need to use this area again unless an error has been made. The uppermost interface element is the Translation Text Area. This is never obscured and the users can immediately see their translation, which typically completes in a fraction of a second, without needing to move their hand.

Communication Errors

When using a machine-translation system for communication, there is a risk of misunderstanding because the machine-translation result may not be correct, and since the source language user does not understand the target language, he or she does not understand the machine-translated result. As explained earlier, although it is reasonable to expect that machine-translation errors will be reduced by using a picture-based user interface, machine-translation systems are still not free from errors. In order to tackle misunderstandings arising from machine-translation errors, the user interface has two further user interaction possibilities.

First, the correctness of the translation can be confirmed by back-translation: the user presses the Check button (#2 in figure 5), and the translated text is fed to a machine-translation system that translates from the target language back into the source language.

Second, the sequence of icons is explicitly displayed on the device for both communicators to see. Picture books have problems associated with viewing and memorizing the sequences of icons, especially when they are located on different pages. The proposed system has the full functionality of the picture-based translation aids, and in addition the selected icons are more easily shared by the two communicators since they are in plain view on the display.

Cultural Issues

Some icons have meaning only within one particular culture. Figure 9 shows two examples of icons that were drawn by Japanese artists for our system that are not appropriate for use in the system because their meaning is specific to Japanese culture. In the first, the character in the icon is bowing, and in the second the Japanese circle symbol called maru (which corresponds to the tick symbol used to indicate correctness) is used to denote OK. This maru symbol is meaningless in most cultures. The only solution to this problem is to find icons that will work across cultures. In the case of OK, the tick symbol cannot be used as it would be confusing to Japanese users potentially meaning the opposite of what was intended. The A-OK symbol in the right-hand icon in the figure, would convey the right meaning in most cultures; in Japanese it can mean zero, or money if inverted, but it is more commonly used to mean OK. It is the icon currently used for OK in the system, but it may not be suitable for use in Peru where it is an offensive gesture.

Linguistic Issues

The first prototype of the picoTrans system was developed only for a single language, Japanese, since resources were limited and the system was being developed in Japan. In some ways Japanese is a difficult language for input with picoTrans, since, for example, there are no word boundaries as such in Japanese; however, in other important respects we believe Japanese is a language that is well suited to this approach. Japanese is constructed using content words together with particles that typically indicate the grammatical function of the content words they postfix in the sentence. For example looking at the expression communicated in figure 2 (we romanize it here with the English meaning in parentheses): *resutoran*(restaurant) *made*(to) *takushi*(taxi) *de*(by) *ikitai*(want to go). These content-word/particle units are relatively self-contained and can usually be moved freely into other positions in the sentence without damaging the meaning or even the grammaticality of the expression. In our example, *takushi*(taxi) *de*(by)

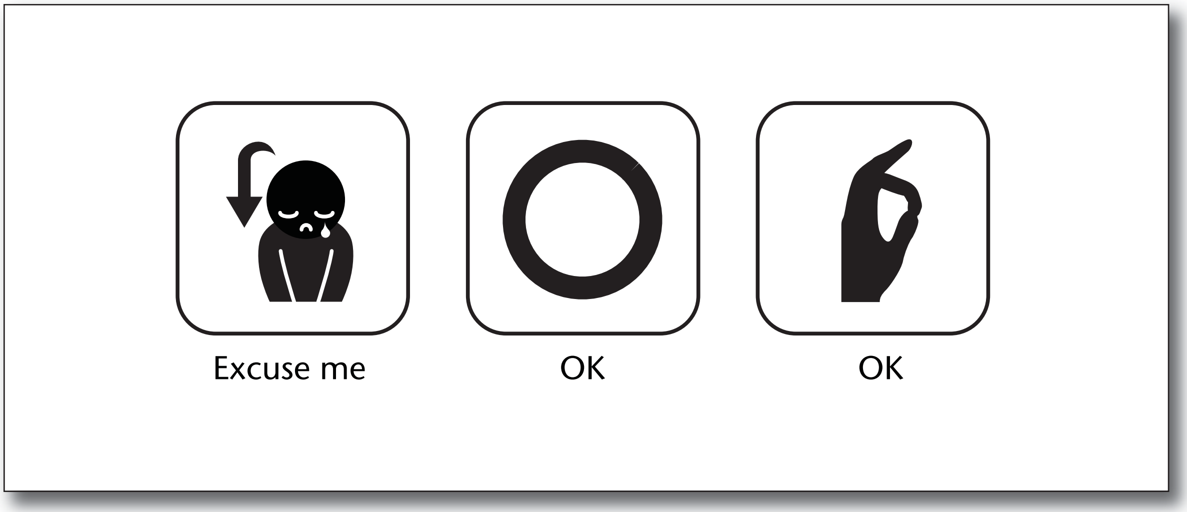


Figure 9. Culturally Specific Icons.

resutoran(restaurant) *made*(to) *ikitai*(want to go) is also absolutely fine.

In short, these content-word/particle units are ideal candidates to represent the picture icons in picoTrans: simply strip away the particles, make picture icons for the content words, and restore the particles during the language generation process. Of course this is an oversimplification of the real process, but we believe nonetheless it is representative enough to cast doubt on the applicability of picoTrans to other languages.

To address this concern, we developed a second prototype system that used the same icon set as the Japanese version but allowed input in a far less well-behaved language: English. We first developed the English system in the same fashion as the original Japanese system, and found that although the natural language generation was about the same quality as the Japanese system, the process of interaction with the system to refine the generated language did not work as effectively as in Japanese. Our solution was to allow more detailed interaction at the word level. The user interface for English input is shown in the top part of figure 10. The primary difference between the English and Japanese interfaces is the existence of + elements in between the generated words. These allow the user to insert additional words in any position as needed. The user is also able to modify words by tapping on them or to remove them by swiping. The word choices offered during the interaction at the lexical level are guided by the statistical language models of the source language generation engine.

Out-of-Vocabulary Words and Expressions

Out-of-vocabulary words (OOVs) create an enormous problem for machine translation. Translation systems that are trained on bilingual corpora that do not contain a particular word are unable to translate it. Since bilingual corpora are expensive to produce and therefore limited in size, OOVs are a common occurrence in user input. Typically one of the following strategies is adopted to handle OOVs, although none are particularly satisfactory:

Do not translate the OOV. Nothing corresponding to the OOV will appear in the target word sequence.

Pass the OOV directly into the target word sequence as-is. Occasionally this will work, but often a foreign word in the target word sequence isn't desirable.

Transliterate the OOV. If the word is a loan word this can be effective, but often words that should be translated will be transliterated in error.

Our approach to overcome the inability of machine translation systems to translate OOVs is to use images automatically acquired from the web and/or image libraries to annotate icons that have no associated images, and possibly no translation known to the system. The users themselves provide the annotation during the communication process. This approach is applicable to both OOVs and entries from bilingual dictionaries that have no icon associated with them. In many cases, even for very rare words and expressions, images will exist on the web, and the image alone will serve as a proxy for the lexical translation in both languages.

When creating new icons for a word, the users are presented with a set of possible icons each annotated with a different picture and are able to choose the icon that best suits their purpose. This is shown in figure 10, in which the user has typed the word "zarf" into the text input area. The word "zarf," meaning "cup-holder," was chosen deliber-

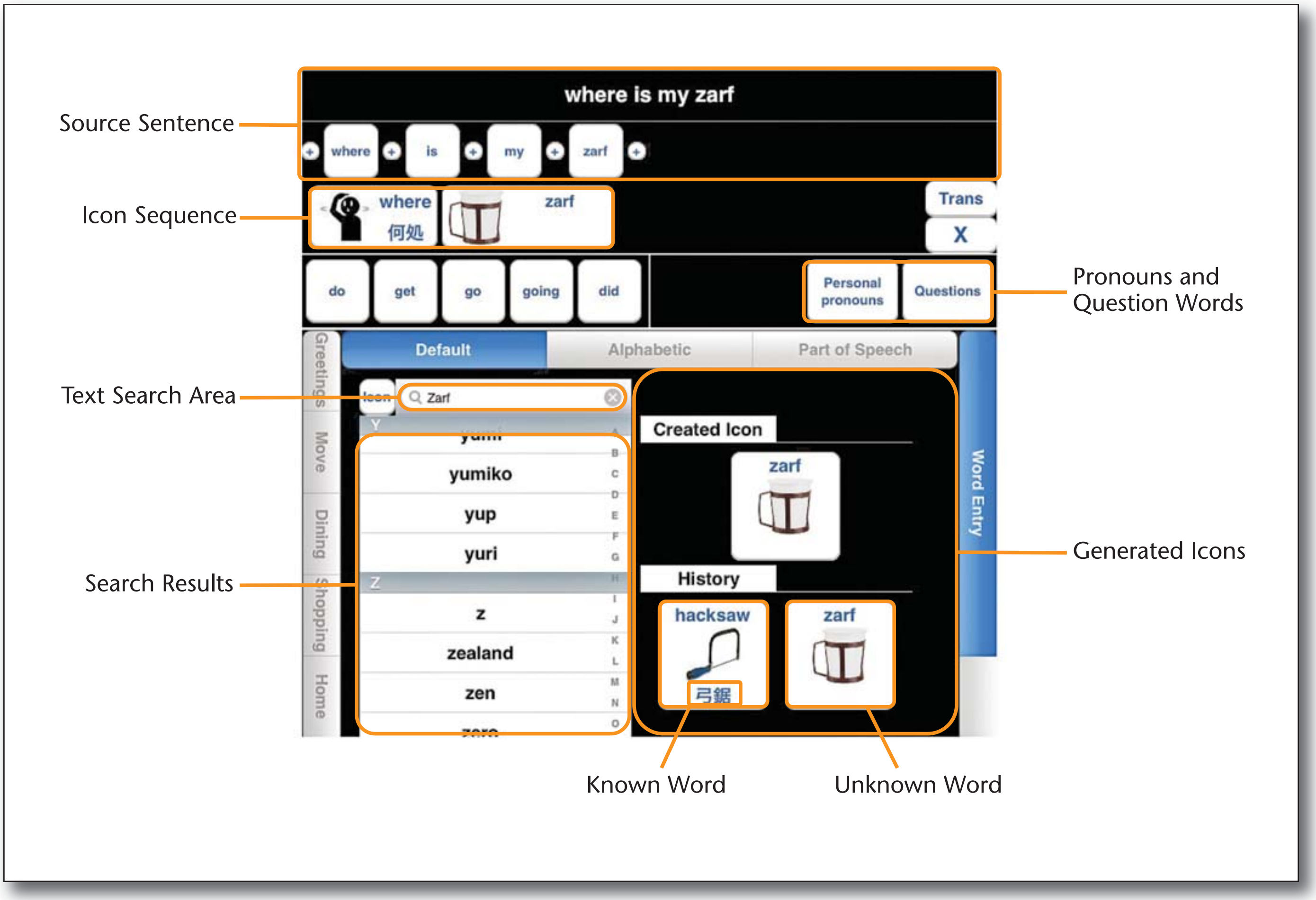


Figure 10. Dynamically Created Icons for a Word Not Represented in the Icon Set.

ately as it is so obscure it is not in the icon set, nor is it in the training corpus for the machine-translation system, nor is it in a large bilingual dictionary resource the system is able to consult to annotate icons. Nevertheless, the system has retrieved an image for this icon from the web and can provide the user with additional images to choose from if necessary. The user has chosen an image for the icon, and the monolingually annotated icon is shown in the Created Icons area. The system keeps a history of previously created icons, and in the figure you can see an icon for hacksaw. This icon wasn't in the icon set but was in the bilingual dictionary and has therefore been annotated with its correct translation. It should be possible to use the icon selections made by the users as a means of gathering images suitable for annotating the icons in the system without the need to have icons hand drawn. In addition this process allows the users the freedom to choose the most appropriate image for their purposes.

User Input Methed Preference

To see how far we could push our idea of icon sequence input, we constrained the input process in all our experiments to use only icons. However, users have their own preferences and may prefer to input text using familiar interfaces, or even prefer spoken input. Pictures may have their own inherent amigiuties (Ma and Cook 2009) that give rise to problems of their own, and sometimes it may be advantageous to avoid using them. We believe that it would be interesting to explore methods for lexical, icon sequence, and speech input within the context of a single interface that allowed users to input in whatever mode they felt most appropriate.

Evaluation

Probably the biggest concern we had about the picoTrans system from the outset of the project was how well it would be able to generate natural

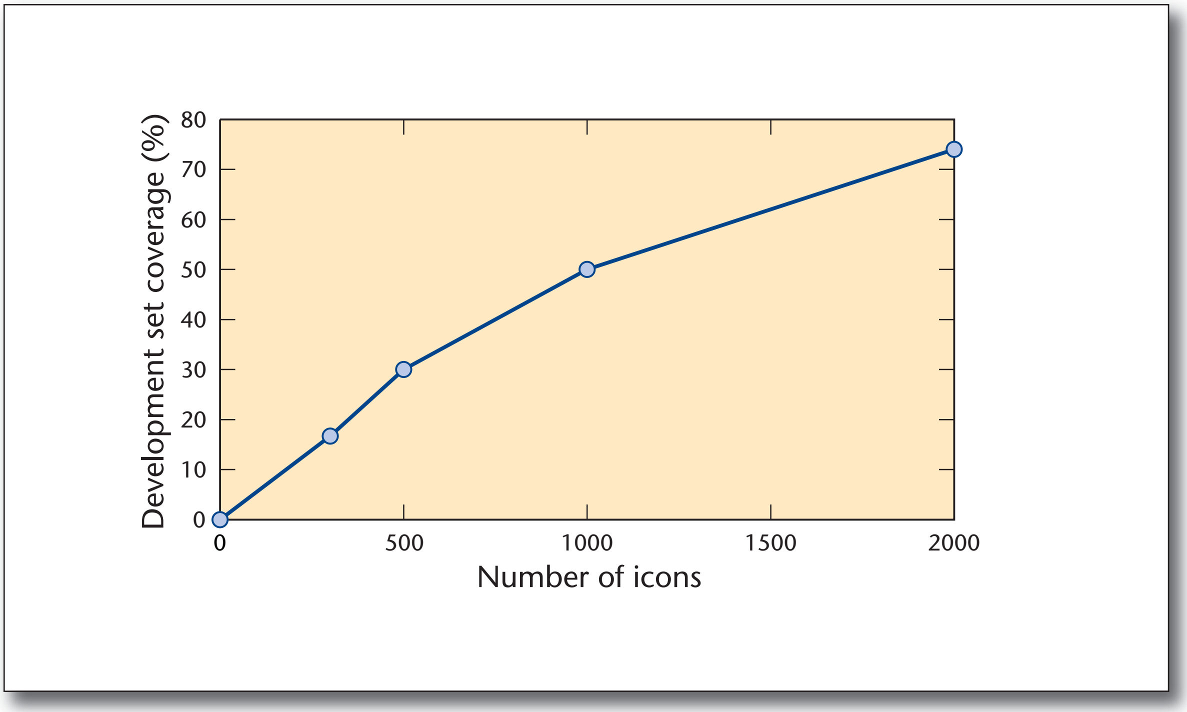


Figure 11. The Coverage of Unseen Data with Icon Set Size.

language from a sequence of icons: how well a limited set of icons can cover the vocabulary necessary to communicate in travel dialogues; how well these icons can be combined to form compound expressions; and how much interaction would be needed from the user to fix the system's mistakes. We conducted an evaluation of the user interface to determine the proportion of travel domain sentences from our corpus it was capable of expressing. To do this we took a sample of sentences from a set of held-out data drawn from the same sample as the training corpus and determined whether it was possible to generate a semantically equivalent form of each sentence using the icon-driven interface and its source sentence generation process.

The version of the prototype system used in this experiment had not been developed sufficiently to include sets of icons to deal with numerical expressions (prices, phone numbers, dates and times, and so on), so we excluded sentences containing numerical expressions from our evaluation set (the evaluation set size was 100 sentences after the exclusion of sentences containing numerical expressions). The set of icons used in the evaluation corresponded to subsets of the 2010 most frequent content words in the English side of the training corpus, that is, content words that occurred more than 28 times in the corpus. This value was chosen such that the number of icons in the user interface was around 2000, a rough estimate of the number of icons necessary to build a useful real-world application.

We considered the full set of icons together with 250-, 500-, and 1000-icon subsets to measure how the system's coverage diminishes with more and more restricted icon sets. We found that we were able to generate semantically equivalent sentences for 74 percent of the sentences in our evaluation data. This is shown in figure 11 together with statistics (based on a 30-sentence random sample from the 100 evaluation sentences) for cases where fewer icons were used. We feel this is a high level of coverage given the simplifications that have been made to the user interface. A more surprising result came when we looked at how well the system was able to guess the intended meaning from the icon sequence: for 49 of the 74 sentences that we were able to cover with our system (66 percent of them), the system proposed the correct source sentence to the user the first time, with no icon refinement necessary. Decreasing the icon set size resulted in a lowering of the coverage of the system. In our view, a 2000-icon set size would be acceptable, but ideally more icons would be useful. This motivated our work toward the extensions presented in the previous sections, which should give the system considerably more expressive power.

Efficiency

We investigated the entry efficiency of our user

Question	picoTrans	Book
Do you think the meaning was communicated successfully?	4.40	3.37
How do you like the system/book?	4.38	2.89
How usable is the system/book?	4.75	2.75
How efficient was the system/book?	4.38	2.75
Are you confident you communicated what you intended?	4.00	2.89
How frustrating to use was the system/book?	3.75	2.00

Table 1. Users' Impressions of Using the picoTrans System and a Picture Book.

interface by measuring the number of key-press actions needed to input sentences using icons relative to the number that would have been needed to input them using the device's text-entry interface. We assumed that each icon would require three key presses to select, but often the icons from the same icon subcategory can be used, and these icons would only require one key press, so our estimate represents an upper bound for the number of key press actions necessary. The time required for one key press isn't equal for icon input and text input, and we did not measure this in our experiments. We also made no attempt to measure the effect of user input errors on the input process. Measuring these factors remains future work. Our measurements include the additional key presses needed to select the semantics of ambiguous icons and also the key presses necessary to modify the source sentence to have the intended meaning. Our measurements do not include the key presses necessary for kana-kanji conversion.

In our experiments we found that the icon entry system required only 57 percent of the number of key press actions of the text entry method: 941 key presses for the icon-driven input method as opposed to 1650 for text entry.

User Experience

An important factor to consider when developing a system of this kind is how much its users like to use it. This may be only loosely related to its technical competence. In order to measure the users' overall experience we conducted an experiment to compare it to using the picture book, which our system hopefully improves on. We gave the system to eight experimental participants who were fluent in both English and Japanese and asked these users to communicate basic travel expressions using both picoTrans and the picture book. The expressions were chosen so that it was possible to express them using either method, and the users received about three minutes of instruction on how to use each method before the experiment started. Each subject used both the book and the picoTrans system, and these two trials were carried out in sequence. We used a different scenario for each trial and balanced the experiment so that the number of participants that used each combination of scenario and communication method was equal (four participants each). The two scenarios consisted of short sentences from the travel domain, for example: "Could you show me the drinks menu?"

After the experiment the participants were asked a series of questions about their experience and overall impressions from using both picoTrans and the picture book. The answers to these questions were on a scale from 1 to 5, with higher scores always being more favorable. The average scores for each communication method together with the questions themselves are shown in table 1. The results indicate that users generally prefer to use the picoTrans system over the picture book, and in particular the biggest differences between the systems were in the experiments concerning usability, efficiency, and frustration. Participants were allowed a maximum of five minutes to communicate each test sentence. All participants were able to communicate all sentences within this time limit using the picoTrans system; however, two of the participants failed to communicate one sentence when using the picture book.

The current version of the system includes many improvements based on the results and feedback from this user evaluation. To answer the initial questions we voiced about the practicability of our idea, we are pleased to report that a development effort is underway and we plan to release an industrial application based on this technology in the future.

Conclusion

In this article we have presented a novel user interface for cross-language communication on mobile devices. The novelty of our system comes from the fact that it uses a sequence of picture icons as its input, and from this generates natural language in the native languages of the users of the system. Our approach is a fusion of the popular picture book translation aid and machine translation, and aims to exploit the strengths of each while mitigating their main weaknesses.

The process of pointing and tapping on icons is a natural way of inputting on mobile devices, and our experiments have shown that input can be performed efficiently on picoTrans, which is able to predict the user's intended expression from the icon sequence most of the time. Furthermore, our approach opens up a second visual channel of communication between the users in a language of symbols. This channel isn't as expressive as natural language but often it can be sufficient in itself. picoTrans is being developed into an industrial application, and this article has presented some of the issues that this development process has uncovered.

The original prototype was limited to a few thousand icons and could only handle Japanese input; the current system is capable of generating icons dynamically from a bilingual dictionary, from places on a map, or from bespoke dialogues designed for numerical input and can handle either English or Japanese user input. Experiments with the Japanese prototype indicate that users prefer to use this system for communication over both a picture book and a machine-translation system. Our idea opens up an enormous number of possibilities for future research. We believe our technology could find useful applications in other restricted domains, especially those where it's critically important that the correct meaning gets across; the medical domain, for example. We would like to investigate other possible applications of the icon sequence input method used in the picoTrans system. Possible applications might include aiding communication for people with language difficulties and assistive technology for language learners.

Perhaps the most interesting avenue to explore in the future is the most general form of our idea, where the system mediates in a process of visual translation through collaboration. The process of arriving at an understanding from a set of images is the spirit of the game Pictionary and might actually be posed as an entertaining and engaging process for the users, assuming the machine doesn't spoil the fun.

References

Baur, D.; Boring, S.; and Butz, A. 2010. Rush: Repeated Recommendations on Mobile Devices. In *Proceeding of the 14th International Conference on Intelligent User Interfaces,* 91–100. New York: Association for Computing Machinery.

Finch, A., and Sumita, E. 2008. Phrase-Based Machine Transliteration. Paper presented at Workshop on Technologies and Corpora for Asia-Pacific Speech Translation Hyderabad, India, 11 January.

Finch, A., and Sumita, E. 2010. A Bayesian Model of Bilingual Segmentation for Transliteration. Paper presented at the Seventh International Workshop on Spoken Language Translation, Paris, 2–3 December.

Finch, A. M.; Song, W.; Tanaka-Ishii, K.; and Sumita, E. 2011. picoTrans: Using Pictures as Input for Machine Translation on Mobile Devices. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence,* Vol. 3, 2614–2619. Menlo Park, CA: AAAI Press.

Finch, A.; Dixon, P.; and Sumita, E. 2012. Rescoring a Phrase-Based Machine Transliteration System with Recurrent Neural Network Language Models. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012,* 47–51. Stroudsburg, PA: Association for Computational Linguistics.

Hu, C.; Bederson, B. B.; and Resnik, P. 2010. Translation by Iterative Collaboration between Monolingual Users. In *Proceedings of Graphics Interface 2010,* 39–46. Toronto: Canadian Information Processing Society.

Ma, X., and Cook, P. R. 2009. How Well Do Visual Verbs Work in Daily Communication for Young and Old Adults? In *Proceedings of the 27th International Conference on Human Factors in Computing Systems,* 361–364. New York: Association for Computing Machinery..

Mihalcea, R., and Leong, C. W. 2008. Toward Communicating Simple Sentences Using Pictorial Representations. *Machine Translation* 22(3): 153–173.

Murphy, J., and Cameron, L. 2008. The Effectiveness of Talking Mats with People with Intellectual Disability. *British Journal of Learning Disabilities* 36(4): 232–241.

Paul, M.; Yamamoto, H.; Sumita, E.; and Nakamura, S. 2009. On the Importance of Pivot Language Selection for Statistical Machine Translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics,* Companion Volume: Short Papers, 221–224. Stroudsburg, PA: Association for Computational Linguistics.

Song, W.; Finch, A. M.; Tanaka-Ishii, K.; and Sumita, E. 2011. picoTrans: An Icon-driven User Interface for Machine Translation on Mobile Devices. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, 23–32. New York: Association for Computing Machinery.

Vogel, D., and Baudisch, P. 2007. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 657–666. New York: Association for Computing Machinery.

Zhu, X.; Goldberg, A. B.; Eldawy, M.; Dyer, C. R.; and Strock, B. 2007. A Text-to-Picture Synthesis System for Augmenting Communication. In *Proceedings of the 22nd National Conference on Artificial intelligence,* 2: 1590–1595. Menlo Park, CA: AAAI Press.

Andrew Finch received received his Ph.D. in computer science from the University of York, UK. Since graduating he has worked as a researcher in natural language processing at ATR laboratories, and is currently a researcher at NICT Research Laboratories, Kyoto, Japan. His research interests include most aspects of natural language processing, in particular machine translation and transliteration.

Wei Song received his bachelor's degree from Beijing University of Aeronautics and Astronautics in 2008, and master's degree under the supervision of Kumiko Tanaka-Ishii at the University of Tokyo in 2011. He is currently working as an engineer at Canon Inc.

Kumiko Tanaka-Ishii received her Ph.D. from the University of Tokyo, was an associate professor of the University of Tokyo from 2003 to 2012, and is currently a professor at Kyushu University. The goal of her research is to gain understanding of the mathematical structure of language, which she has pursued in the domains of computational linguistics and natural language processing.

Eiichiro Sumita received B.E. and M.E. degrees in computer science both from University of Electro-Communications, Japan, in 1980 and 1982, respectively. He received a Ph.D in engineering from Kyoto University, Japan, in 1999. He is head of the Multilingual Translation Laboratory at NICT Research Laboratories, Kyoto, Japan. His research interests include natural language processing, machine translation, information retrieval, automatic evaluation, e-learning, and parallel processing. He is a member of IEICE, IPSJ, and ACL.