

Regular Paper

A Tool for Creating Glossed Japanese Reading Materials

GREGORY HAZELBECK^{1,a)} HIROAKI SAITO¹

Received: March 28, 2011, Accepted: September 12, 2011

Abstract: We present a system that assists Japanese-language teachers with the creation of electronic reading materials which contain glosses. Although glosses are traditionally generated for only content words, we propose a hybrid method for identifying and glossing functional expressions and conjugations, which enables the system to both generate more glosses and display them in an appropriate manner for learners. Coverage analysis and empirical evaluations show that our hybrid method allows our system to cover more functional expressions and conjugations than previous systems while maintaining good performance. Feedback received during interviews with Japanese-language teachers show that they react very positively toward both the new glosses and the system in general. Finally, results from a survey of Japanese as a foreign language learners show that they find the new glosses for functional expressions and conjugations to be very helpful.

Keywords: CALL, reading materials, automatic glossing, functional expressions

1. Introduction

Preparing materials for a class can be a time consuming task for teachers, and this is no exception for Japanese-language teachers. For instance, when preparing reading materials teachers must dedicate time to not only finding a suitable text, but also preparing supplementary materials that help learners understand the vocabulary and grammar used in the text. Tools that assist teachers in creating these materials have the potential to save them time as well as make it easier for them to create richer materials that improve the students' learning experience.

Figure 1 shows one type of electronic reading material that is often seen on the Internet. A text is shown with links attached to each word appearing in it. Clicking on one of these links displays an explanation for that word in the right-hand column. These explanations, called glosses, have been shown to improve reading comprehension in numerous studies [3], [5], [13]. Furthermore, glosses used in a computer-assisted language learning (CALL) environment have been shown to have an even greater effect on reading comprehension than traditional paper-based glosses [14]. This is because CALL-based glosses are more flexible in giving faster access to definitions of a larger number of words.

In this paper, we present a system that assists Japanese-language teachers in the creation of the same type of reading material shown in Fig. 1. Although reading assistant systems, which learners can use to automatically generate this type of reading material, have been developed in previous work, to our knowledge there has been no previous system that specifically aids teachers with the labor intensive task of creating these materials. Therefore, we decided to develop a system that gives teachers maximum flexibility in the creation of these materials while still

providing many of the time-saving benefits of fully automated systems. This is accomplished by first automatically generating glosses for a given text, and then letting the teacher edit those glosses to fit their needs.

Automatic generation of glosses enables the teacher to provide learners with a wealth of helpful information without having to go through the time consuming process of collecting and writing all of it themselves. While glosses have traditionally been generated for only content words, our system uses a hybrid method that we call HGlosser which enables it to generate them for functional expressions and conjugations as well. Japanese functional expressions are compound expressions that contain content and function words, and can have both compositional and non-compositional meanings. An example would be the functional expression について (nitsuite) which can have a compositional meaning of "to follow" or a non-compositional meaning of "about." Conjugations are verb forms created by inflection and in Japanese they serve many purposes. One example would be the potential form which is used to express the ability to do something. The potential form of the verb 走る (hashiru), meaning "to run," is 走れる (hashireru), meaning "to be able to run."

Supporting functional expressions and conjugations is necessary because both are very important parts of Japanese grammar. Functional expressions account for a very large percentage of the grammar lists for the intermediate and advanced levels of the Japanese Language Proficiency Test (JLPT) [6], and a strong grasp of conjugations is fundamental to understanding Japanese. However, supporting conjugations also provides our system with an additional benefit besides facilitating understanding. Like many natural language processing systems in Japanese, our system uses a morphological analyzer as a pre-processing step. However, using the output of this tool directly would produce results that are unnatural to many Japanese as a foreign language (JFL) learners. This is due in part to the parts of speech

¹ Graduate School of Science and Technology, Keio University, Yokohama, Kanagawa 223-8522, Japan

^{a)} greg@nak.ics.keio.ac.jp

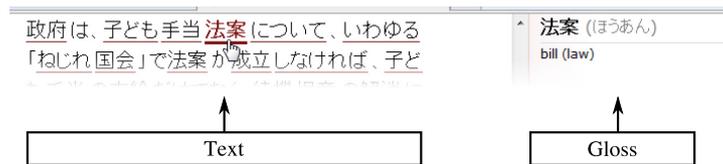


Fig. 1 An example of an electronic reading material with glosses (shown from our system).

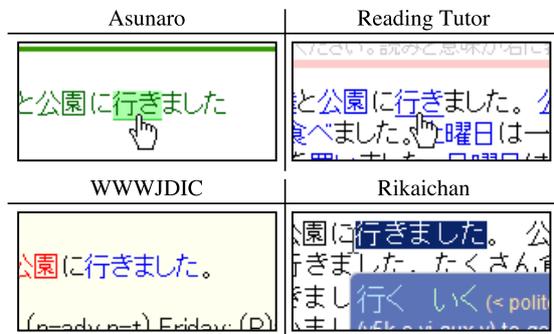


Fig. 2 An example of how word and gloss boundaries differ between reading assistant systems.

that morphological analyzers use. These parts of speech are based on the type of Japanese grammar that is taught to native speakers [4], [19], which differs significantly from the descriptive grammar used when teaching JFL [18]. Yoshikawa [18] gives an example of how the construction of 食べて (tabete) is explained to native speakers as the conjunctive particle て (te) attached to the conjunctive form of the verb 食べ (tabe), while in JFL it is simply called the te-form of the verb 食べる (taberu). Thus, even word boundaries differ between the two views of Japanese grammar. This difference can also be observed in several reading assistant systems. The top of Fig. 2 shows two systems (Asunaro [1]^{*1} and Reading Tutor [7]^{*2}) developed at universities in Japan that use a morphological analyzer while the bottom of Fig. 2 shows two systems (WWWJDIC's Text Glossing feature^{*3} and Rikaichan^{*4}) developed by JFL learners that do not use a morphological analyzer. One can clearly see that the boundaries of words (and glosses) are different. Based on these observations and on preliminary feedback that we received on our own system, we decided to convert the output of the morphological analyzer to a format that is more natural for JFL teachers and learners. This conversion is easily accomplished in our system because HGlosser supports conjugations. To our knowledge there has been very little previous research that investigates glossing in a JFL setting of not only content words, but also more grammatical elements like functional expressions and conjugations. In this paper we give a detailed description of HGlosser, its implementation, and analyze its coverage and performance.

Although HGlosser can generate glosses for more than just content words, there are still words like compound nouns, proper nouns, particles, etc. that are unsupported and are thus left un-glossed. Therefore, in our system we provide a way to easily edit

the glosses, which allows the teacher to add glosses to the reading material that could not be automatically generated. Although this editing must be performed manually, we believe it enables the creation of more comprehensive and customized reading materials.

The rest of this paper is organized as follows: in Section 2 we introduce related work; in Section 3 we give a detailed explanation of HGlosser; in Section 4 we describe the tools for editing glosses; in Section 5 we present results of our empirical evaluations of HGlosser, one-on-one interviews with Japanese-language teachers, and a survey of JFL learners; finally in Section 6 we conclude and discuss future work.

2. Related Work

There have been numerous reading assistant systems developed that generate the same type of reading material that we focus on in this paper. Reading Tutor [7] uses a morphological analyzer called ChaSen to parse the text and then displays information from the EDR dictionary for each word found. Feedback from learners about the system has been very positive. A survey of 35 advanced learners showed that all of the learners thought that the system was useful. Asunaro [1] was developed for engineering students from Asia and features a multilingual interface that supports viewing word definitions in English, Chinese, Thai, Malaysian, and Indonesian. It also uses a dependency structural analyzer called CaboCha to display a dependency tree of the text. Rikaichan, an extension for the Mozilla Firefox web browser, generates glosses for any document that can be viewed with that software. Although these systems generate the same type of reading material as our system, they fundamentally differ in purpose. These systems are tools that learners use when reading while our system is a tool that teachers use to create reading materials.

Gloss generation for non-content words is performed in several of the above reading assistant systems. Reading Tutor does provide a separate tool that generates glosses in Japanese for only functional expressions in a given text. Rikaichan supports the generation of conjugation glosses as well as some functional expression glosses but is unable to differentiate between non-compositional and compositional usages of such expressions. According to Abekawa et al. [2] support for functional expressions was planned for Asunaro, however at the time of writing this paper support for functional expressions is not yet implemented in the publicly accessible version. In contrast our system supports gloss generation for both functional expressions and conjugations. The method that our system uses can differentiate between non-compositional and compositional usages of functional expressions and has been proven [12] to perform better than the

^{*1} <http://hinoki.ryu.titech.ac.jp/asunaro/index-e.php>

^{*2} http://language.tiu.ac.jp/index_e.html

^{*3} <http://www.csse.monash.edu.au/~jwb/cgi-bin/wwwjdic.cgi?9T>

^{*4} <http://www.polarcloud.com/rikaichan/>

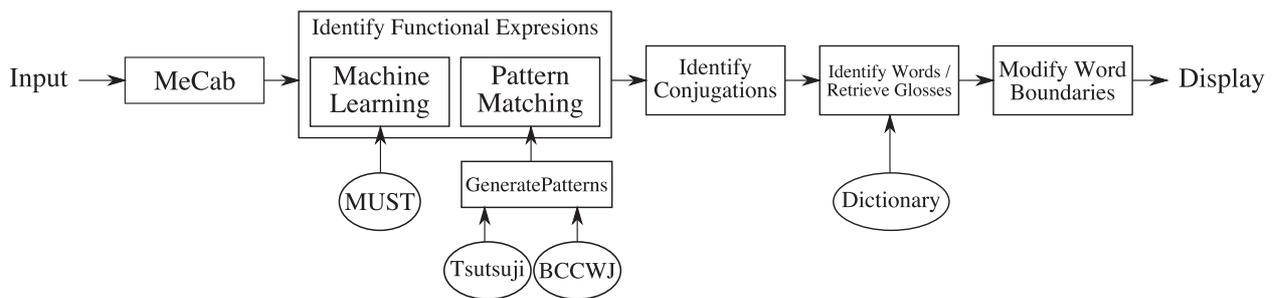


Fig. 3 HGlosser's gloss generation process.

identification method used in Reading Tutor [8]. Finally, the setting in which our system performs gloss generation for functional expressions and conjugations is also different from the reading assistant systems. Those systems are being used directly by the learner so any gloss generated would be immediately displayed to them, whereas in our system the teacher has the opportunity to modify these glosses before the learner ever uses the reading material.

Previous research on Japanese functional expressions has included work on identification methods as well as resources that aid identification. A hierarchical dictionary of functional expressions called Tsutsuji was developed in Matsuyoshi, Sato, and Utsuro [9]. The top level of the dictionary's nine level hierarchy contains the lexical form of 341 expressions. The second level categorizes these expressions by meaning. The remaining seven levels contain various surface forms for each expression where the insertion/deletion of particles and other conjugations have been made. We use the Tsutsuji dictionary in HGlosser for pattern matching (Section 3.2).

To our knowledge Tsuchiya et al. [15] presented one of the first methods for identifying Japanese functional expressions. Their method used manually created rules and patterns based on morphological information to identify each functional expression. Another method for identifying functional expressions was presented by Tsuchiya et al. [17] and uses a supervised machine learning technique called Support Vector Machine (SVM). MUST [16], a corpus of manually annotated examples of 337 different functional expressions collected from the 1995 Mainichi newspaper corpus, was used as training data for this method. Shime et al. [12] presented an empirical evaluation that compared these two methods and the results show that the method using machine learning performs best. We do use Tsuchiya et al. [17]'s method in our system. However, since the coverage of this method is limited to the functional expressions covered in MUST, we decided to integrate it into our own hybrid method in order to cover more functional expressions. We also make several changes (detailed in Section 3.1) to the implementation of this method in order to adapt it to our specific application.

Yu et al. [19] also made the same observation that the output of current morphological analyzers is not suitable in a JFL setting. They modified the output to use part-of-speech (POS) labels that JFL teachers are familiar with, enabling them to more easily annotate sentence patterns in Japanese text. We do not change the POS labels in our system because they are never displayed, but we do perform more aggressive modifications in order to produce

Table 1 Examples of Japanese functional expressions.

にあたり (niatari)	
1. Functional	アパートに入居するにあたり、隣所に挨拶回りをするのは日本の習慣です。 It is a custom in Japan to greet your neighbors when you move into a new apartment.
2. Content	ボールが顔面にあたり、歯が折れた。 The ball hit me in the face and broke my tooth.
なければなりません (nakerebanarimasen)	
3. Functional	明日は学校に行かなければなりません。 I have to go to school tomorrow.

appropriate gloss boundaries.

3. Automatic Glossing

In this section we give a detailed explanation of HGlosser. Figure 3 shows each step performed by HGlosser to automatically generate glosses for a text that has been input into the system. Teachers can input and manipulate any text through the system's web interface. HGlosser will first pre-process the text using MeCab (version 0.98), a morphological analyzer. For MeCab we chose to use the UniDic (version 1.3.12) dictionary instead of its default dictionary because it was shown that it performs better on texts from a wide range of domains [11].

Next, HGlosser uses both machine learning and pattern matching techniques to identify functional expressions. This is a difficult task because functional expressions in Japanese can have both compositional and non-compositional meanings. For example, in Table 1, sentences 1 and 2 contain the *にあたり* (niatari) compound expression. In sentence 1, this expression has a functional, non-compositional meaning of "when." However, in sentence 2, the same expression has a compositional meaning that results simply from using the post-particle *に* (ni) and verb *あたり* (a conjugated form of *あたる* (ataru), meaning "to hit") together. We refer to this as the content usage of a functional expression. However, there are also functional expressions where this type of content usage is very rare (or even nonexistent). Sentence 3 shows an example of the *なければなりません* (nakerebanarimasen) functional expression which has a very common functional meaning of "must" or "have to."

3.1 Machine Learning

We first apply the method proposed in Tsuchiya et al. [17] that uses machine learning to identify functional expressions with balanced functional vs. content usage ratios. In this method the problem of functional expression identification is first formulated as a

Morpheme	Morphological Features	Candidate Features	...	Predicted Label
m_{i-3}	入居 ... ニュウキョ	- - ...		O
m_{i-2}	する ... スル	- - ...		O
m_{i-1}	に ... ニ	2 1 ...		B-FUNCTIONAL
m_i	あたり ... アタリ	2 2 ...		?
m_{i+1}	、 ... -	- - ...		?
m_{i+2}	隣 ... トナリ	- - ...		?
m_{i+3}	近所 ... キンジョ	- - ...		?

Fig. 4 An example of how YamCha performs chunking. Morpheme m_i 's label is predicted using all of the outlined features and labels.

chunking task. Japanese sentences are represented as a sequence of morphemes, so the task becomes to identify the morphemes that constitute a functional expression and then chunk (or group) those morphemes into either a functional or content chunk (corresponding to functional and content usages respectively). The authors perform chunking with an SVM-based tool called YamCha.

Features used to train the SVMs are categorized into two groups: morphological features and candidate functional expression features. Morphological features consist of a morpheme's surface form, POS, conjugation type and form, dictionary form, and pronunciation. Candidate functional expression features are only used when a morpheme is known to appear in a functional expression. These features include information like the length of the candidate functional expression and the position of the current morpheme in that expression. Please see Tsuchiya et al. [17] for a more detailed explanation of the features. **Figure 4** shows how YamCha uses these features to perform chunking. YamCha predicts morpheme m_i 's label using all of the features for morphemes m_{i-2} to m_{i+2} as well as m_{i-1} and m_{i-2} 's predicted labels. Labels are in the IOB2 format which is used to represent chunks. In this format, the first morpheme in a chunk is labeled as B while all other morphemes in the chunk are labeled as I. Any morphemes not in a chunk are labeled as O. Since in this task there are two types of chunks (functional and content), the set of possible labels is: {B-FUNCTIONAL, I-FUNCTIONAL, B-CONTENT, I-CONTENT, O}.

For our implementation of this method we first developed our own SVM-based chunking tool in Python using LIBSVM's (version 2.89) Python interface. This tool is very similar to YamCha except that it uses an SVM with a linear kernel while YamCha uses a polynomial kernel. In the original method both functional and content usages could be explicitly identified because each would be in a different chunk. However, in our application, we only need to identify functional usages because content usages can be identified similarly to any other word via dictionary lookup. Therefore, we removed the content chunk labels in the MUST training data so that only functional usages would be identified. We also decided to train a separate model for each functional expression because it enables us to add new training data and functional expressions without having to retrain everything.

When identifying functional expressions in a given text, we perform chunking using each functional expression's model. However, since we already know all of the possible surface forms that a functional expression's morpheme sequence can have, we

only need to perform chunking on portions of the text where such surface forms appear. We refer to these specific surface forms as candidate functional expressions. Our first step in identifying functional expressions is to find all of the candidate functional expressions in the text^{*5}. If we take sentence 1 in Table 1 as an example, we would find the にあたり (niatari) candidate functional expression in this step. Next, we extract the morpheme sequence for each candidate functional expression along with up to three morphemes that precede and follow them in the text. In our example, we would extract the following morpheme sequence: {に (ni), 入居 (nyuukyo), する (suru), に (ni), あたり (atari), 、 , 隣 (tonari), 近所 (kinjyo)}. Chunking is then performed on this new smaller sequence of morphemes using only the candidate functional expression's corresponding model, which would be にあたり (niatari)'s model in our example. Finally, the labels for this smaller morpheme sequence are transferred back to the text's original morpheme sequence. When compared to the naïve method of performing chunking on the entire text with every model, this method drastically reduces HGlosser's execution time because relatively few models are used and only a small portion of the text is actually chunked.

3.2 Pattern Matching

While the machine learning-based method can handle functional expressions with balanced functional vs. content usage ratios, recent work suggests that this only covers a third of all functional expressions [10]. Therefore, in the next part of HGlosser we use patterns to identify functional expressions with a high ratio of functional usage. First, surveys are conducted of functional expressions in Tsutsuji using the Balanced Corpus of Contemporary Written Japanese (BCCWJ)^{*6}. Currently, we are focusing on adding support for the auxiliary verb type functional expressions (てしまう (teshimau), ている (teiru), etc.). As of writing this paper, we have selected 24 of these expressions from Tsutsuji's top level as candidates for pattern generation. We also included various surface forms of these expressions from other levels of Tsutsuji resulting in a total of 675 candidate functional expressions. Pseudo code for the algorithm used to generate patterns is shown in **Fig. 5** along with pseudo code for the **ExtractCand-**

^{*5} Overlapping candidate functional expressions are resolved by the same rules that Tsuchiya et al. [17] used. Specifically, select the candidate that starts at the leftmost morpheme. If more than one candidate starts at the same morpheme then select the longest candidate.

^{*6} Balanced Corpus of Contemporary Written Japanese Monitor Release Data (2009 Version).

```

GeneratePatterns( $C$ : list of candidates from Tsutsuji)
01  $\mathcal{P} \leftarrow$  an empty list
02 for each candidate  $c$  in  $C$  do
03    $S \leftarrow$  sentences that contain  $c$  in the BCCWJ
04   for each sentence  $s$  in  $S$  do
05      $M_s \leftarrow$  morpheme sequence of  $s$ 
06      $M_c \leftarrow$  ExtractCandMorph( $c, M_s$ )
07     if  $M_c \neq \text{null}$  and VerbChk( $c, M_c, M_s, \mathcal{P}$ ) = true then
08       append  $M_c$  to  $\mathcal{P}$ 
09       break out of loop on line 4
10     end if
11   end for each
12 end for each
13 return  $\mathcal{P}$ 

```

Fig. 5 The **GeneratePatterns** algorithm.

```

ExtractCandMorph( $c$ : functional expression candidate,
 $M_s$ : morpheme sequence for a sentence  $s$ )
01 for each morpheme  $m$  at position  $i$  in  $M_s$  do
02   if  $c$  starts with  $m$ 's surface form then
03      $M_c \leftarrow$  a list with a single element  $m$ 
04      $c_{\text{notmatched}} \leftarrow$  substring of  $c$  not matched by  $m$ 's surface
05      $m_{\text{offset}} \leftarrow 1$ 
06     while  $c_{\text{notmatched}} \neq$  the empty string do
07       if  $i + m_{\text{offset}} =$  length of  $M_s$  then
08         break out of loop on line 6
09       else
10          $m_{\text{next}} \leftarrow$  the morpheme in  $M_s$  at position  $i + m_{\text{offset}}$ 
11         if  $c_{\text{notmatched}}$  starts with  $m_{\text{next}}$ 's surface form then
12           append  $m_{\text{next}}$  to the list  $M_c$ 
13            $c_{\text{notmatched}} \leftarrow$  substring of  $c_{\text{notmatched}}$  not matched by  $m_{\text{next}}$ 's surface
14           increment  $m_{\text{offset}}$  by 1
15         else
16           break out of loop on line 6
17         end if
18       end if
19     end while
20     if  $c_{\text{notmatched}} =$  the empty string then
21       return  $M_c$ 
22     end if
23   end if
24 end for each
25 return null

```

Fig. 6 Pseudo code for the **ExtractCandMorph** method.

```

VerbChk( $c$ : functional expression candidate,
 $M_c$ : candidate's morpheme sequence,
 $M_s$ : morpheme sequence for a sentence  $s$ ,
 $\mathcal{P}$ : list of previously extracted candidate morpheme sequences)
01  $m_{\text{prev}} \leftarrow$  the morpheme preceding  $M_c$  in  $M_s$ 
02 if  $m_{\text{prev}}$  is a verb then
03   if  $c$  is in the top level of Tsutsuji then
04     return true
05   else if  $c$  is in the lower level of Tsutsuji
06      $M_{\text{parent}} \leftarrow$  morpheme sequence of  $c$ 's top-level parent in Tsutsuji (found in  $\mathcal{P}$ )
07     if last morpheme in  $M_c$  exists in  $M_{\text{parent}}$  then
08       return true
09     end if
10   end if
11 end if
12 return false

```

Fig. 7 Pseudo code for the **VerbChk** method.

Morph and **VerbChk** methods in Fig. 6 and Fig. 7 respectively.

The **ExtractCandMorph** method simply returns the candidate c 's morpheme sequence. If the candidate's string does not match the boundaries of morphemes in M_s then *null* is returned. The **VerbChk** method returns true if a candidate is from Tsutsuji's top level and the morpheme immediately preceding it in M_s is a verb.

It returns true for lower level candidates if the last morpheme in its morpheme sequence is also in the morpheme sequence of its top-level parent candidate from Tsutsuji. We force candidates from lower levels to satisfy an extra condition because their lower frequency in the BCCWJ increases the probability that a sentence with the wrong expression/usage will be selected. This algorithm

Table 2 An example of HGlosser’s output at various steps.

Input								
引っ越すにあたり、近所の方々が手伝ってくださった。 (The neighbors helped us out when we moved.)								
MeCab								
Morpheme	引っ越す	に	あたり	...	手伝っ	て	下さっ	た。
Identify Functional Expressions (Machine Learning)								
Morpheme	引っ越す	にあたり	...	手伝っ	て	くださっ	た。	
Tag	-	niatari	...	-	-	-	-	-
Identify Functional Expressions (Pattern Matching)								
Morpheme	引っ越す	にあたり	...	手伝っ	てくださっ	た。		
Tag	-	niatari	...	-	tekudasaru	-	-	-
Identify Conjugations								
Morpheme	引っ越す	にあたり	...	手伝っ	てくださっ	た。		
Tag	-	niatari	...	-	tekudasaru	past	-	-
Modify Word Boundaries								
Morpheme	引っ越す	にあたり	...	手伝ってくださった				

近所の方々が手伝ってくださった。

手伝う (てつたう)
to help, to assist, to take part in

くださる
An honorific that expresses that someone else did something for you.
(Click here for more information)

た
Verb Past Tense
(Click here for conjugation information)

Fig. 10 An example of how a generated gloss is displayed in our system.

Table 3 HGlosser benchmarks on two data sets.

Data Set	Text Length (in characters)		Execution Time (in seconds)		
	Maximum	Minimum	Average	Maximum	Minimum
Newspaper	4,533	38	0.037	0.524	0.004
Books	1,032	108	0.059	0.133	0.011

て下さっ
⟨Morpheme(て, Particle, て), Morpheme(下さっ, Verb, 下さる)⟩

Fig. 8 A pattern generated by GeneratePatterns.

Verb Past Tense
Position=-1, POS=Verb
Position=0, DictForm=た

Fig. 9 Conjugation pattern.

produces one pattern per functional expression. As can be seen in the example in **Fig. 8**, a pattern is composed of the expression’s morpheme sequence, where each element in the sequence contains the morpheme’s surface form, POS, and dictionary form. All patterns are used to also check if the previous morpheme is a verb. Using this algorithm, we were able to generate 395 of these patterns with our 675 candidate functional expressions.

3.3 Conjugations and Glossing

After the functional expressions have been identified, the next step identifies any conjugations in the text. Since there are only a limited number of conjugations, we decided to manually create patterns for each one. **Figure 9** shows an example of one of these patterns. Next, content words in the text are identified using morphological information provided by MeCab and dictionary look-

up. Glosses are then retrieved from several resources for each content word, functional expression, and conjugation. We use the Japanese-English dictionary EDICT^{*7} as the default dictionary to generate glosses for content words and currently includes all of the word’s senses in its gloss. Finally, modifications are made to the word boundaries by attaching auxiliary verb type functional expressions and conjugations to the word stem preceding them. **Table 2** shows how an example sentence is glossed by HGlosser and **Fig. 10** shows how one of the glosses is finally displayed in the system.

After developing HGlosser we ran several benchmarks to investigate its execution time. We used two data sets, one that contained 100 articles selected at random from the 1995 Mainichi newspaper corpus and another that contained 50 texts from Japanese language textbooks and test preparation books. Every benchmark was run on an Intel Core 2 Duo 3.0 GHz machine with 4 gigabytes of RAM. The results of these benchmarks are shown in **Table 3**. As can be seen in this table, HGlosser was able to process each text in both data sets in under a second, which we believe is sufficient for our application.

^{*7} http://www.csse.monash.edu.au/~jwb/edict_doc.html



Fig. 11 The gloss editor in the system's web interface.



Fig. 12 How to remove (left) and add (right) glosses.

4. Gloss Editing

In this section we will describe how our system facilitates gloss editing. Once glosses have been automatically generated and displayed, the system provides several tools to assist the teacher in editing them. The first tool is the gloss editor itself (shown in Fig. 11). The body of the gloss can be edited using a rich text editor, which enables the teacher to easily add things like links, pictures and tables even if they do not know HTML. The next tool displays alternate definitions for the word taken from other resources. This enables the teacher to quickly change the definition if they prefer the way that an alternate definition is written. Currently, alternate English and Japanese definitions are taken from the EDR dictionary and Wikipedia. However, in the future, dictionaries for other languages could be added.

Another tool displays links to the word's entry on online dictionaries. Several online dictionaries are provided by default and teachers can easily add new online dictionaries to the system. This tool not only provides teachers with a quick way of looking up words online, but by clicking a checkbox beside the link it will also display the same link directly on the gloss. This gives learners an easy way to access the same online resource.

The system also records every modification that has been made to a gloss by the teacher. Then, along with the previous two tools, it displays a history of changes made to the word's gloss. The teacher can easily change the gloss to one of the versions in the history, preventing them from having to write the same gloss multiple times.

Finally, while gloss links are generated automatically, the system also provides a gloss link editing mode that enables teachers

to add and remove these links. As Fig. 12 shows, a link can be removed by clicking on it and then clicking on the delete button that is subsequently displayed, whereas new links can be added by clicking on an unlinked character and then using the arrow buttons to select the rest of the link's characters. This mode enables the teacher to create new glosses for words that the system is unable to identify automatically. There could also be situations where the teacher wants to attach only a few glosses to the text. In this case, the teacher can turn off the automatic glosses and use this mode to create only the glosses that they need. Once the teacher is finished modifying the glosses, the final reading material can be exported as a single HTML file.

5. Evaluation

In this section we present the results of several evaluation experiments. We first show the results of a set of experiments that were conducted to evaluate HGlosser's performance and coverage. Then, we present the results of interviews with Japanese-language teachers about the system. Finally we present results of a survey of JFL learners showing their reactions to the new glosses generated by our system.

5.1 HGlosser's Performance

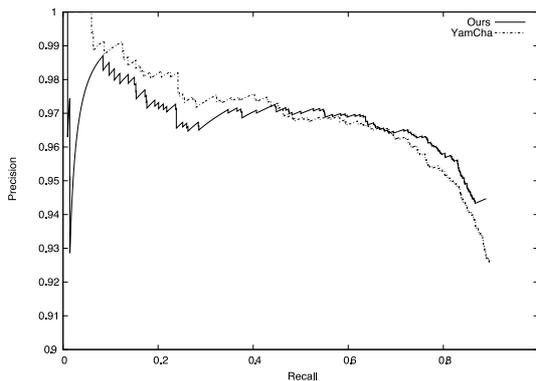
We conducted two experiments to evaluate HGlosser's ability to correctly identify functional expressions. The first experiment was conducted to verify that our implementation of Tsuchiya et al. [17]'s method performs as well as the original. Evaluations of the original method conducted in Shime et al. [12] used a set of 59 functional expressions. Our testing data only contained 54 of these expressions, which were all used for our experiments. We

Table 4 Training/testing corpora.

Corpus	Examples		Words
	func	content	
Training	1,767	1,463	114,699
Testing	5,347	1,418	244,324

Table 5 Chunker performance.

Metric	Chunker	
	YamCha	Ours
Precision	0.926	0.944
Recall	0.902	0.890
$F_{\beta=1}$	0.914	0.916

**Fig. 13** Precision-Recall curve for the first experiment.

trained HGlosser using the MUST corpus and tested it on annotated functional expression data from the 1995 Mainichi newspaper corpus. Details of both data sets are shown in **Table 4** and results of the first experiment are shown in **Table 5** and **Fig. 13**. The results show that our implementation performs as well as the original. Since YamCha was used in the original paper, only the results obtained from using SVM with the polynomial kernel were reported. However, in our implementation we found that using the linear kernel gives slightly better precision.

An error analysis of the false positives shows that this method has difficulties with functional expressions where both functional and content usages appear in similar contexts. For example, if we look at the training data for the functional expression と言いながら (toinagara), we see that many of its functional usages are preceded by some kind of quote and followed by a comma. However, when we look at its content usages we see that they also appear in the exact same kind of context. Although these occasional false positives will result in the generation of an incorrect gloss, this is not a serious issue in our system because teachers can easily remove such glosses.

In the second experiment we evaluated the patterns that we extracted from the BCCWJ. We first used HGlosser to extract functional expressions from the entire BCCWJ. Then we randomly selected 2000 instances of the functional expressions that were identified with the extracted patterns. After verifying all 2000 instances we only found 2 incorrectly extracted expressions. Both errors occurred when the copula である (dearu) was incorrectly identified as the auxiliary verb である (tearu). In the next version we plan to fix this simply by modifying the **VerbChk** function in our algorithm to also verify that the verb has been correctly

Table 6 Results of the coverage experiment.

System	Conjugations	Auxiliary Verbs	Other Functional Expressions
HGlosser	52	39	85
MUST	-	15	85
Rikaichan	52	-	-
JLPT	52	39	286

conjugated.

5.2 HGlosser's Coverage

In the next experiment we evaluated HGlosser's coverage of functional expressions and conjugations. Since this system was developed for use in a JFL setting, we decided to calculate coverage as the number of functional expressions and conjugations in the JLPT grammar lists [6] that are supported. We use grammar lists based on the old four-level format of the JLPT because lists for the new five-level format have not been released. For conjugations, we count the number of conjugations in levels 3/4. For functional expressions, we first divide them into two categories: auxiliary verb type in levels 3/4 and others in levels 1/2, and then count the supported expressions in each category. For comparison, we also calculated the coverage of Rikaichan and MUST. As can be seen in the final results in **Table 6**, the hybrid approach used by our system enables it to cover functional expressions and conjugations that previous systems do not support. This allows us to not only provide learners with more glosses, but also display them in a manner they are more accustomed to.

5.3 Teacher Interviews

In order to evaluate the system's usability and gather general feedback, we conducted one-on-one interviews with four Japanese-language teachers. Each teaches Japanese to foreign students of a different level (beginning, intermediate, intermediate-advanced, or advanced) at a university in Japan. During the interviews, we first gave a short 10–15 minute demonstration of the system where we explained each feature and how it was used. We then gave the teacher a chance to use the system and answered any questions that they had about it. Finally, we asked them a set of questions about the different features of the system as well as their opinions on how the system could be used. We will first present feedback received from the teachers about our system's functional expression and conjugation glosses. Then we will present feedback about other features of the system.

5.3.1 Functional Expression and Conjugation Glosses

In order to evaluate the effect of supporting functional expression and conjugation gloss generation in our system, we had each teacher use two different versions of the system, one version with support for these new types of glosses and one version without (i.e., glosses were generated for content words only). We then asked each teacher which version they preferred and all four teachers answered that the version with support for the new glosses was better. The teacher of the beginning class said that it was very important to have such glosses because they often observed learners who would ignore the grammar of a sentence and just look up the definitions of each content word. The teacher

of the intermediate class said that such glosses would be especially useful for intermediate level learners because that is when functional expressions are often introduced. The teacher of the intermediate-advanced class said that they would definitely want those kinds of glosses in the reading material and the teacher of the advanced class said that there may be times when they do not want a certain expression's gloss showing in the reading material because they are teaching it in class, but that they definitely want the system to support these kinds of glosses because then they could simply remove the gloss for that one expression. All of the teachers believed that support for functional expression and conjugation gloss generation was both useful and necessary for this type of system.

5.3.2 Other System Features

The teachers' reactions to the overall system was very positive. All four of the teachers said that the system was easy to use and none had any major difficulties when using the system even after only a short demonstration. Two of the teachers even made specific comments about how responsive the system was. All of the teachers also said that they found the alternate definitions and web dictionary tools useful. Many reacted very positively to being able to quickly access a word's entry in a web dictionary. One teacher said that they would probably not use the alternate definitions for regular words, but that the definitions from Wikipedia would be very useful when working with higher level texts. All teachers also said that they had no problem with the method for adding/removing gloss links. One teacher commented that being able to remove the glosses is especially helpful since some would be unnecessary for higher level learners. However, another teacher said that while the current method is fine, it might be easier if new glosses could be created by click-and-dragging rather than using arrow buttons.

None of the teachers expressed any major problems with the way our system is currently displaying glosses. We did ask their opinion about the absence of ruby in the system. Ruby are small pronunciation guides written above kanji characters. Teachers often write ruby on reading materials to help learners read kanji that they might not be familiar with. In this system (and all of the reading assistant systems mentioned in this paper) the pronunciation of the kanji is shown inside the gloss, not directly above the characters in the text. Therefore, learners must click on the gloss' link to see the pronunciation. Teachers of the intermediate-advanced and advanced classes said that not displaying ruby was actually better because it would force learners to try to remember the pronunciation of the kanji rather than rely on the ruby. However, teachers of the beginning and intermediate classes said that it would be better to have an option to display ruby. Although the teacher of the beginning class did acknowledge the same issue with ruby as the intermediate-advanced and advanced class teachers mentioned. Given that some teachers would use an option to display ruby though, we believe it would be helpful to include it in future versions of the system.

We also asked the teachers about whether they would use the Japanese definitions that the system provides. Three of the teachers said that they would use the Japanese definitions for very advanced learners, while the other teacher of the intermediate-

Table 7 Results of the JFL learner survey.

Reading Material	Very Bad	Bad	Fair	Good	Very Good
Beginner	0	0	2	2	2
Intermediate	0	0	0	3	2
Advanced	0	0	0	1	3
Total	0	0	2	6	7

advanced class said that they would not. This teacher said that it was not because of the system itself, but rather that they did not believe there existed a Japanese dictionary that had been sufficiently edited for learners. None of the teachers said that they had any problems with the manner in which our system displayed gloss links, or with the format of the conjugation and auxiliary verb glosses. They also did not express any problems with the amount of functional expressions supported. Finally, all of the teachers commented that this system would make it very easy for them to create reading materials that learners could use to prepare before coming to class. Overall, the teachers were very pleased with the system and their comments gave us valuable feedback that will help us during future work.

5.4 Learner Survey

In the last evaluation experiment, we surveyed 15 JFL learners to collect their reactions to the new functional expression and conjugation glosses. The learners were asked to use one of three reading materials (one for beginning, intermediate, and advanced proficiency levels respectively) that were constructed using our system. The texts used for the beginning and intermediate level materials were selected from a Japanese textbook while the text used for the advanced material was a newspaper article. We asked the learners to pay special attention to the new types of glosses for functional expressions and conjugations in the materials. After using the reading material the learners were simply asked to rate it based on how helpful and easy to understand the new glosses were. Results of the survey are shown in **Table 7**. Overall the results of the survey show that learners react very positively to the new types of glosses with no learners giving them a Bad or Very Bad rating.

6. Conclusions and Future Work

In this paper we presented a system that assists Japanese-language teachers with creating glossed reading materials. We gave details of HGlosser, a hybrid method for identifying and glossing functional expressions and conjugations, which enables our system to not only generate more glosses but also display them in a manner more appropriate for JFL learners. We also described the tools that teachers can use in the system to edit glosses. We presented results of evaluation experiments that showed HGlosser can accurately identify more functional expressions and conjugations than previous systems. Finally, we presented results of one-on-one interviews with Japanese-language teachers and a survey of JFL learners. These results showed that teachers find the system to be very useful and learners react positively to the new types of glosses that our system generates.

As future work we plan to make several modifications to the system based on the feedback we gathered from the teachers. We also plan to continue to increase the types of functional expres-

sions that the system can cover as well as investigate new ways of utilizing teacher interaction to facilitate the creation of these types of reading materials.

Acknowledgments We thank Takehito Utsuro for allowing us to use his annotated functional expression data. We also thank all of the Japanese-language teachers and learners for their valuable feedback. Finally, we thank the two anonymous reviewers for all of their helpful comments and suggestions.

Reference

- [1] Abekawa, T. et al.: Development of Japanese Reading System of Multi-Lingual Environment "Asunaro", *Proc. 3rd International Conference on Computer Assisted Systems for Teaching and Learning Japanese*, pp.71–74 (2002).
- [2] Abekawa, T. et al.: Development of Japanese learning system "ASUNARO" and new movement – Learning syntax and the evaluation, *The 65th National Convention of IPSJ*, pp.327–330 (2003).
- [3] Aweiss, S.: Situating learning in technology: The case of computer-mediated reading supports, *Journal of Educational Technology Systems*, Vol.23, pp.63–74 (1994).
- [4] Den, Y. et al.: The development of a multi-purpose electronic dictionary for morphological analyzers, *Report from the Electronic Dictionary Group at the "Japanese Corpus" General Meeting*, pp.21–26 (2006).
- [5] Hayden, S.: An investigation into the effect and patterns of usage of a computer-mediated text in reading comprehension in French. Unpublished Doctoral Dissertation, University of Pennsylvania (1997).
- [6] Japan Foundation and Association of International Education Japan: *Japanese Language Proficiency Test: Test Content Specifications*, Bonjinsha, Revised Edition (2002).
- [7] Kawamura, Y., Kitamura, T. and Hobara, R.: Development of a reading tutorial system for JSL and JFL learners using the EDR Japanese-English dictionary, *Japan Journal of Educational Technology*, Vol.24 (Suppl.), pp.7–12 (2000) (in Japanese).
- [8] Kawamura, Y. et al.: Proposal for Three New Features in the Japanese Reading Tutorial System, *Proc. 4th International Conference on Computer Assisted Systems for Teaching and Learning Japanese*, pp.37–40 (2007) (in Japanese).
- [9] Matsuyoshi, S., Sato, S. and Utsuro, T.: Compilation of a Dictionary of Japanese Functional Expressions with Hierarchical Organization, *Proc. ICCPOL*, pp.395–402 (2006).
- [10] Nagasaka, T. et al.: Analysis and Detection of Japanese Functional Expressions based on a Hierarchical Lexicon, *Proc. 16th Annual Meeting of the Association for Natural Language Processing*, pp.970–973 (2010) (in Japanese).
- [11] Ogiso, T. et al.: Morphological Analyzer Dictionary Benchmark – A Comparison of IPAdic, NAIST-jdic, and UniDic by Genre, *Proc. 16th Annual Meeting of the Association for Natural Language Processing*, pp.326–329 (2010) (in Japanese).
- [12] Shime, T. et al.: Automatic Detection of Japanese Compound Functional Expressions and its Application to Statistical Dependency Analysis, *Journal of Natural Language Processing*, Vol.14, pp.167–197 (2007) (in Japanese).
- [13] Stoehr, L.E.: The effects of built-in comprehension aids in a CALL program on student-readers' understanding of a foreign language, (Doctoral dissertation, University of Texas at Austin, 1999) *Dissertation Abstracts International*, Vol.60, 4353-A (1999).
- [14] Taylor, A.M.: CALL-Based Versus Paper-Based Glosses: Is There a Difference in Reading Comprehension?, *CALICO Journal*, Vol.27, pp.147–160 (2009).
- [15] Tsuchiya, M. et al.: Detecting Japanese Functional Expressions based on Morphological Clues, *Proc. 11th Annual Meeting of the Association for Natural Language Processing*, pp.584–587 (2005) (in Japanese).
- [16] Tsuchiya, M. et al.: A Corpus for Classifying Usages of Japanese Compound Functional Expressions, *Proc. PACLING*, pp.345–350 (2005).
- [17] Tsuchiya, M. et al.: Chunking Japanese Compound Functional Expressions by Machine Learning, *Proc. 2nd International Workshop on Web as Corpus (EACL-2006)*, pp.11–18 (2006).
- [18] Yoshikawa, T.: *Introduction to Japanese Grammar*, ALC Press Inc., (1989).
- [19] Yu, Z. and Sano, H.: Development of Sentence Retrieval System for Japanese Teaching Material Production: Mechanism and Evaluation of Improvement of Search Expression, *IPSJ SIG Notes*, Vol.2006,

pp.25–32 (2006) (in Japanese).



Gregory Hazelbeck received his B.S. in computer science from the University of Kansas in 2006, and M.S. degree from Keio University in 2008. Since 2008 he has been a Ph.D. student in the Graduate School of Science and Technology at Keio University. His research interests include natural language processing and computer-assisted language learning.



Hiroaki Saito received his Ph.D. degree in 1991. He is currently an associate professor at Keio University. His research interests include natural language processing and music comprehension.