International Conference
ICT for Language Learning

YANSIS: An “Easy Japanese” Writing Support System

Takeshi Nagano[1], Akinori Ito[2]
Tohoku University (Japan)

Abstract

***I**n these days, many foreigners visit Japan from many countries for sightseeing, education or getting jobs. In Japan, they have many natural disasters such as earthquake, tsunami, flood or eruption. When a disaster occurs, an authority is responsible to inform of various information for evacuation or life under a refugee camp to not only Japanese native speakers but also non-native speakers. However, multilingual announcement is not a realistic way of achieving this because of the limitation of human resource under such an emergency situation.*

Under this background, “Easy Japanese” (EJ) has been proposed. EJ is a constrained language, which consists of limited vocabulary and grammar. Thanks to the linguistic limitation, a sentence in EJ is more easily understood than that in ordinary Japanese by non-native speakers of Japanese. Because of its easiness to use, "Easy Japanese" has been widely used in public office web site, leaflet or broadcast.

The problem is that composing sentences in EJ needs training, because a Japanese native speaker cannot understand what words or phrases are difficult to understand for non-native speakers. To make composition in EJ easier, we developed a system that helps a writer to compose sentences in EJ without knowledge of the limitation of EJ. The system is named YANSIS, which stands for "YAsashii Nihongo SIen System” (Easy Japanese writing support system). YANSIS consists of six components. The UI component provides user interface such as Japanese text input and buttons. The Japanese morphological analyzer component is used to split an input sentence into words. We need this component because a Japanese sentence does not have spaces between words. The Japanese level analyzer component determines difficulty level of each word based on the vocabularies of Japanese language proficiency test (JLPT). The recommendation component finds out difficult phrases in the input sentence and recommends how to rewrite them. The Japanese easiness estimator component estimates difficulty of the whole sentence based on machine learning technique. The example search component searches examples of EJ sentences that are related to a word in the input sentence.

The first version of YANSIS was implemented in Java, and thus it runs on any OS if Java runs on it, such as Windows, Linux or Mac OS X. In addition, we ported YANSIS to Android and iOS. Because Android apps are based on Java, we could reuse many components of the original version in the Android version, except the UI component. However, for Java is not available on iOS, we needed to re-implement all components including Japanese morphological analyzer using Objective-C.

1. Introduction

Nowadays, many foreigners visit Japan because of sightseeing, working, getting jobs. Not only visiting, many foreigners stay in Japan for education or work. Therefore, information is often provided in not only Japanese but also other languages such as English, Chinese and Korean.

In Japan, we have many natural disasters such as earthquake, tsunami, eruption, flood. When a disaster happens, many Japaneses as well as foreigners have difficulty to continue their normal life. To reduce the damage of the disaster, the authority informs of helpful information for evacuation or life under a refugee camp to not only Japanese native speakers but also Japanese non-native speakers. The problem is how to inform the non-native speakers of Japanese of precise information quickly. Multilingual announcement is considered as one solution, but it is not realistic because human resource is quite limited under such a situation.

Under this background, ``Easy Japanese'' (EJ) has been proposed as an alternative solution [1]. EJ is a constrained Japanese that uses limited vocabulary and grammar, such as Basic English [2]. This linguistic limitation in EJ makes an EJ sentence more understandable than ordinary Japanese by non-native speakers of Japanese. Because of easiness of understanding, EJ has been widely used in public documents, website of local authority or broadcast [3][4].

The problem of using EJ is that composing sentences in EJ needs training, because an untrained Japanese native speaker cannot understand what words or phrases are difficult to understand for non-native speakers. To make composition in EJ easier, we developed a system that helps a writer to compose sentences in EJ without knowledge of limitation of EJ. The system is named YANSIS, which stands for “YAsashii Nihongo SIen System” (Easy Japanese writing support system).

2. How to support writing in EJ

The informal rules for translating Japanese into EJ was described in [1]. We follow these rules when designing the system. At first, a user composes a sentence in normal Japanese, and input them to YANSIS. Then YANSIS analyzes the sentences and displays suggestions for making the sentence easier, sentence by sentence. Here are what YANSIS do:

(1) To make a sentence easily understandable, length of the sentence should be kept short. Thus, the system warns the user when sentence length is too long.

(2) In a sentence in EJ, difficult words or phrases with ambiguous meaning should be avoided. Thus the system points out which words or phrases are difficult. The user can search the expressions of EJ related to the pointed words or phrases.

(3) The system shows the total easiness of the sentence by 0-100 scale.

A user checks the comments given by YANSIS, and rewrites the sentences so that the warning messages are suppressed. At the same time, the user can confirm that how the revision of a sentence makes the sentence easier to understand by comparing the easiness scores of the sentence before and after the revision. If the input sentence becomes easy enough, the user can add the sentence to the “edited text” field. After writing all the sentences, the user can save the composed sentences as a text file.

3. System implementation

YANSIS was designed to run on various platforms. To achieve this, YANSIS consists of several individual components, which include user interface (UI) component and other five components. Overview of the system is shown in Fig. 1. YANSIS was implemented in Java because Java runs on variety OS, such as Windows, Linux and OS X. Next, we describe details of the components.

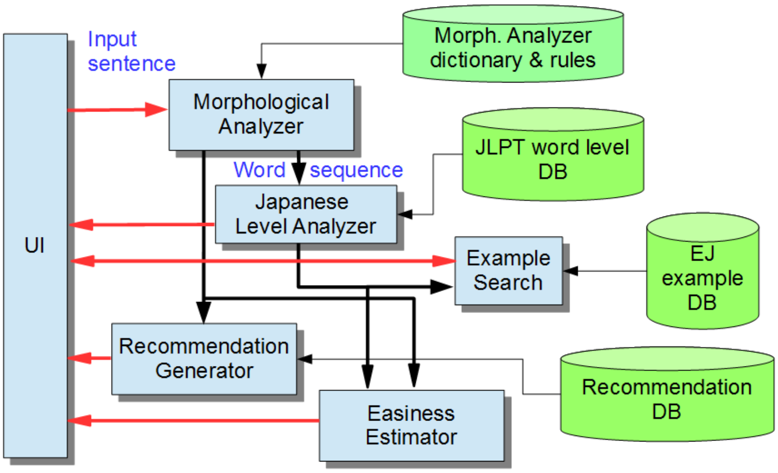


Fig. 1 Overview of the system

3.1 UI component

Fig. 2 shows the UI of YANSIS. The UI has a text input area to input sentence. A user can input one or more sentences in this area. By pushing the analysis button, the system begins analysis of input sentences. The analysis results of the sentences are displayed in the analysis result field, sentence by sentence. Here, difficult words are marked as magenta or red. If the user clicks a word in the displayed sentence, a window pops up, which displays information of the clicked word, such as part of speech, pronunciation, and JLPT level. Fig. 2 shows an example of the pop-up window. The “example sentence button” is in the window; if we click this button, example EJ sentences related to the word are shown in the example sentence field. Evaluation field shows analysis results and recommendations. Add button adds the input sentence to the edited text field.

3.2 Japanese Morphological analyzer component

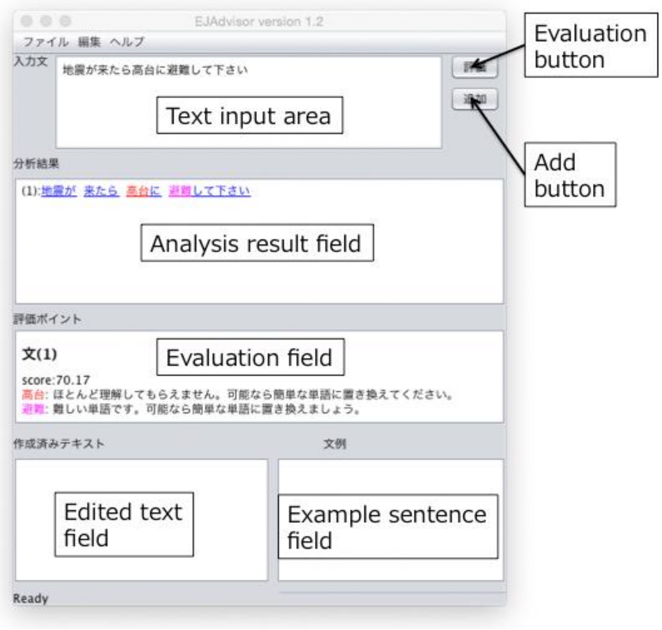
A Japanese sentence has no spaces between words, unlike an English sentence. Thus, we need to split the input sentence into words for investigating difficulty of the words. In general, Japanese morphological analyzers are used for this task [5][6]. We used a morphological analyzer Sen [7] because it is implemented in Java and easy to incorporate into a Java-based application.

3.3 Japanese level analyzer component

The Japanese level analyzer determines difficulty level of a Japanese word based on the vocabulary for Japanese Language Proficiency Test (JLPT) [8]. The EJ we suppose is based on the older version of the JLPT level that has four grades: grade 4 to grade 1, grade 1 to be the most difficult. JLPT was revised in 2010 [9], and the current JLPT has five grades from N5 to N1, but the grade YANSIS suggests does not coincide with the current JLPT grade. YANSIS has Japanese word DB of each (older) JLPT level. When a word is given, YANSIS searches the database to find out JLPT level of the word. Table 1 shows vocabulary size of each grade.

Table.1. Vocabulary of each DB

JLPT GRADE	VOCAB. SIZE
1	3024
2	3770
3	717
4	760

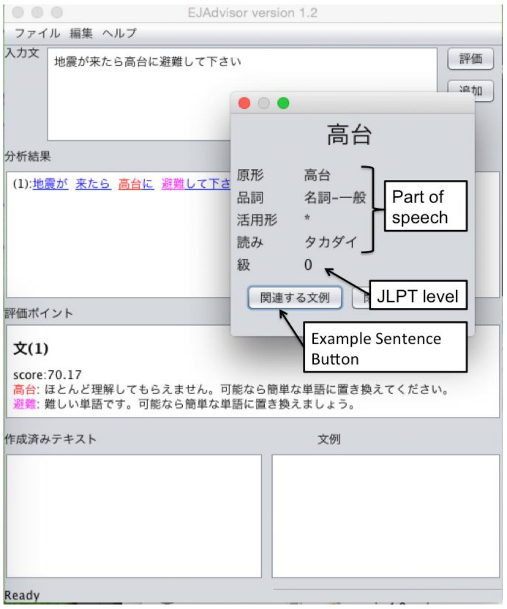


地震が来たら高台に避難して下さい

(1):地震が 来たら 高台に 避難して下さい

文(1)

score:70.17
高台: ほとんど理解してもらえません。可能なら簡単な単語に置き換えてください。
避難: 難しい単語です。可能なら簡単な単語に置き換えましょう。



地震が来たら高台に避難して下さい

(1):地震が 来たら 高台に 避難して下さ

高台

原形 高台
品詞 名詞-一般
活用形 *
読み タカダイ
級 0

文(1)

score:70.17
高台: ほとんど理解してもらえません。可能なら簡単な単語に置き換えてください。
避難: 難しい単語です。可能なら簡単な単語に置き換えましょう。

Fig.2 The UI of YANSIS

Fig.3 Pop-up window

3.4 Recommendation component

Recommendation component shows warning messages and recommendataions for reducing difficulty of the sentence. There are three kinds of recommendations: avoid long sentence, difficult words and phrases.

As mentioned above, a sentence in EJ should be short. YANSIS warns when length of the sentence is longer than 30 syllables. As for the JLPT level, YANSIS checks JPLT level of all words, and gives warnings for words of level 2, level 1 or not in the vocabulary of JLPT. Next, this component carries out pattern matching between the input sentence and the phrases in the recommendation database, in which phrases that should be avoided are stored. The phrases in the DB includes 32 expressions of conjecture (... might be ...), double nagation, or other complicated expressions.

3.5 Japanese easiness estimator component

Japanese easiness estimator component calculates Japanese easiness of the input sentence using multiple regression analysis [10]. Parameters of multiple regression analysis were optimized based on subjective evaluation in advance. Feature parameters using multiple regression analysis were extracted from the input sentence. We used features such as proportion of morphemes belong to a certain JLPT level, proportion of characters in a specific character class (such as Hiragana and Katakana and Kanji).

3.6 Example search component

When we compose sentence, it is helpful to check EJ sentences that are similar to the sentence we want to compose. The example search component searches sentences written in EJ that include a search word from the example sentence DB. The example sentence DB includes sentences of announcement from an authority under disaster, written in both EJ and normal Japanese. When a word is given, the component matches the word to the normal Japanese part of the DB. Then it returns the EJ part of the found sentences.

4. Porting YANSIS to mobile platforms

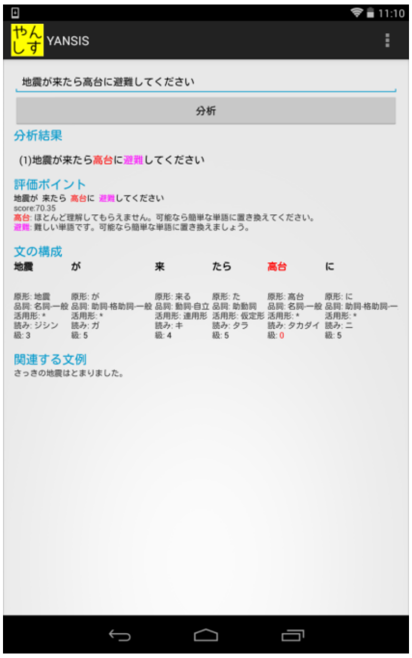
Under a disaster, it is not assured that the authority can use PC, because blackout may happen. If YANSIS can be used on mobile platforms such as smartphone or tablet, it could be used even under the blackout by using various energy sources such as solar cell or hand crank generator [11][12]. Thus, we ported YANSIS to mobile platforms such as Android and iOS.

4.1 Android

Android application is implemented in Java, so we only re-implemented UI component of the original YANSIS, which depends on the system. The Java virtual machine in Android platform has different graphical user interface model from that of PC-based platform, and an application should use fixed-sized single window. Therefore, we changed the UI not to use pop up window. Fig.3. shows a screenshot of Android version of YANSIS.

4.2 iOS

iOS application needs to be implemented in Swift or Objective-C. Since we cannot convert Java source into Swift and Objective-C automatically, we needed to re-implement all components. We chose Objective-C because it is used as developing language for a long time in iOS. In iOS version, we implemented the UI to be almost same as that of the Android version, because we want the iOS version having the same usability of the Android version. Fig.4 shows iOS version of YANSIS.



地震が来たら高台に避難してください

分析結果

(1)地震が来たら高台に避難してください

評価ポイント

地震が 来たら 高台に 避難してください
score:70.35
高台: ほとんど理解してもらえません。可能なら簡単な単語に置き換えてください。
避難: 難しい単語です。可能なら簡単な単語に置き換えましょう。

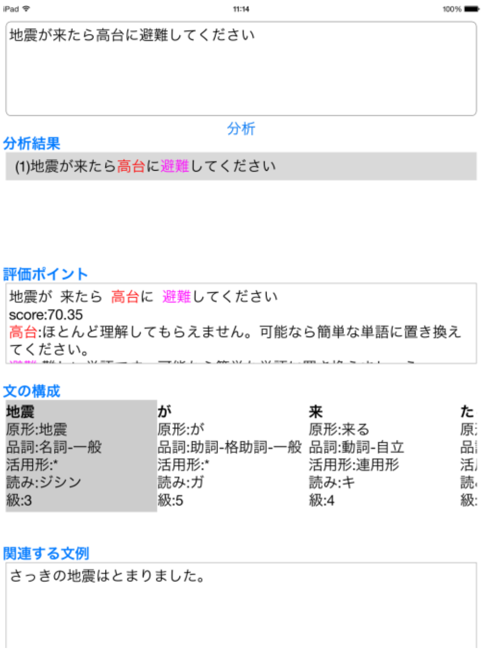
文の構成

地震	が	来	たら	高台	に
原形: 地震	原形: が	原形: 来る	原形: た	原形: 高台	原形: に
品詞: 名詞-一般	品詞: 助詞-格助詞-一般	品詞: 動詞-自立	品詞: 助動詞	品詞: 名詞-一般	品詞: 助詞-格助詞-一
活用形: *	活用形: *	活用形: 連用形	活用形: 仮定形	活用形: *	活用形: *
読み: ジシン	読み: ガ	読み: キ	読み: タラ	読み: タカダイ	読み: ニ
級: 3	級: 5	級: 4	級: 5	級: 0	級: 5

関連する文例

さっきの地震はとまりました。

Fig.4. Android version of YANSIS



地震が来たら高台に避難してください

分析結果

(1)地震が来たら高台に避難してください

評価ポイント

地震が 来たら 高台に 避難してください
score:70.35
高台:ほとんど理解してもらえません。可能なら簡単な単語に置き換えてください。

文の構成

地震	が	来	た
原形:地震	原形:が	原形:来る	原
品詞:名詞-一般	品詞:助詞-格助詞-一般	品詞:動詞-自立	品
活用形:*	活用形:*	活用形:連用形	活
読み:ジシン	読み:ガ	読み:キ	読
級:3	級:5	級:4	級:

関連する文例

さっきの地震はとまりました。

Fig.5. iOS version of YANSIS

5. Conclusion

We have developed an EJ writing support system YANSIS. YANSIS was originally implemented in Java for PC-based platforms, and it can review the input sentence and give comments and recommendations for making the sentence easier. We ported YANSIS to Android and iOS.

As a future plan, we will incorporate the text-to-speech functionality into YANSIS. To this end, we are investigating the relation between intelligibility of Japanese utterance and its prosodic feature such as speaking rate and pause [13]. As a next step, we will implement an HMM-based speech synthesizer [14] and incorporate it into YANSIS.

Acknowledgement

We appreciate Prof. Kazuyuki Sato of Hirosaki University and other members of Yasasii Nihongo Workshop. Part of this work is supported by JSPS Kakenhi Grant-in-Aid for Scientific Research (B) Grant Number 26284069.

References

[1] “Easy Japanese”: http://human.cc.hirosaki-u.ac.jp/kokugo/EJ1a.htm

[2] C. K. Ogden. “Basic English: A General Introduction with Rules and Grammar.” Kegan Paul, Trench, Trubner and Co., 1930

[3] M. Moku and K. Yamamoto. “Investigation of Paraphrase of Easy Japanese in Official Documents.” Proc. The 17th Yearly Meeting of Association for Natural Language Processing. 2011.

[4] NHK, “NHK News Web Easy”, http://www3.nhk.or.jp/news/easy

[5] S. Kurohashi, T. Nakamura, Y. Matsumoto, and M. Nagao. Improvements of Japanese morphological analyzer JUMAN. In Proceedings of the International Workshop on Sharable Natural Language, pp. 22-28, 1994.

[6] T. Fuchi and S. Takagi. Japanese morphological analyzer using word co-occurrence: JTAG. In Proc. Int. Conf. on Computational linguistics (COLING '98), 1998.

[7] Sen - Morphological analyzer written in Java: https://java.net/projects/sen

[8] Japan Foundation, “JLPT Japanese Language Proficiency Test,” http://www.jlpt.jp/

[9] Japan Foundation, “Objectives and History”, http://www.jlpt.jp/e/about/purpose.html

[10] M. Zhang, A. Ito and K. Sato, “Automatic Assessment of Easiness of Japanese for Writing Aid of “Easy Japanese” ”, Proc. Int. Conf. on Audio, Language and Image Processing, Shanghai, pp. 303-307, 2012.

[11] K. Tuncer, “Energy Harvesting for Future Mobile Devices,” Proc. of Seminar “Research Trends in Media Informatics”, pp. 33-39, 2010.

[12] C. Schuss and T. Rahkonen, “Solar energy harvesting strategies for portable devices such as mobile phones,” Proc. 14th Conf. of Open Innovation Association (FRUCT), pp. 132-129, 2013.

[13] H. Prafianto, T. Nose, Y. Chiba A. Ito and K. Sato, “A study on the effect of speech rate on perception of spoken easy Japanese using speech synthesis,” Proc. Int. Conf. on Audio, Language and Image Processing, pp. 476-479, 2014.

[14] H. Zen, K. Tokuda and A. W. Black, “Statistical parametric speech synthesis,” Speech Communication, 51(11) 1039-1064, 2009.