Machine Translation System PENSÉE: System Design and Implementation

Sayori Shimohata Toshiki Murata Atsushi Ikeno
Tsuyoshi Fukui Hideki Yamamoto
Oki Electric Industry Co., Ltd.
Research & Development Group, Kansai Laboratories
1-2-27 Shiromi, Chuo-ku, Osaka 540-6025 Japan
{shimohata245,murata656,ikeno546,fukui556,yamamoto436}@oki.co.jp

Abstract

This paper describes a new version of our machine translation system PENSÉE. In the light of its past systems, new PENSÉE is designed to improve portability from developers' point of view and translation quality from users' point of view. The features of new PENSÉE are: 1) Java implementation and 2) pattern-based transfer approach. In addition, new PENSÉE places a great importance on user interface especially in building user dictionaries. We will discuss why and how we resolve the existing MT problems and present dictionary building tools to support user customization.

1 Introduction

We have been researching and developing a series of machine translation (MT) systems "PENSÉE". PENSÉE appeared on the market first as a Japanese to English MT system in 1986 and then as an English to Japanese one in 1988. In the early stage. PENSÉE was a rule-based transfer MT system running on a workstation. With the improvement in the computer performance and the increase of personal computer(PC) users in 1990's. PENSÉE has changed its platform from a workstation to a PC.

The number of MT users is growing year by year. And accordingly, a more comfortable user environment and a higher-quality translation will be required. To meet the demand of those users, we are launching a new version of PENSÉE which features 1) Java implementation and 2) pattern-based approach. Java implementation ensures portability and security on network computing. Pattern-based approach improves the quality of translation and makes customization easier. In addition, new PENSEE places a great importance on user interface especially in building user dictionaries. In this paper, we will describe the features of new PENSEE and present dictionary building tools to support user customization.

2 Java Implementation

Figure 1 illustrates an overview of new PENSÉE (we call it "PENSÉE"' here after). PENSÉE is written entirely on Java platform. Java is an object-oriented programming language developed by Sun Microsystems. Inc [Gosling et al. 96]. Under the slogan "Write once, run anywhere", Java has been designed for use on networks from the beginning. The application written in Java works on any kind of compatible devices that support the Java platform.

This Java platform's ideal fits the trend of the computer world. Under the circumstances that users can choose and change their platforms to their tastes, the applications are expected to work on as many platforms as possible. Java enables us to develop PENSÉE independent of platforms and saves the time and expense of multiple ports. We are developing PENSÉE on a network computing environment and confirm that PENSÉE will run on the different platforms, such as Unix. Windows, MacOS, and Java OS, without any adjustments [1].

Another merit in adopting Java implementation is that Java makes it easier to use components of the application. This brings us not only the simplicity of system design but also the reusability of components. PENSÉE includes the components which are useful to other natural language processing systems. However, it was difficult to share the components because they heavily depended on specific implementation details. We designed an application programming interface

[1] After the development period, we will register PENSÉE for the 100% Pure Java(TM) certification process.

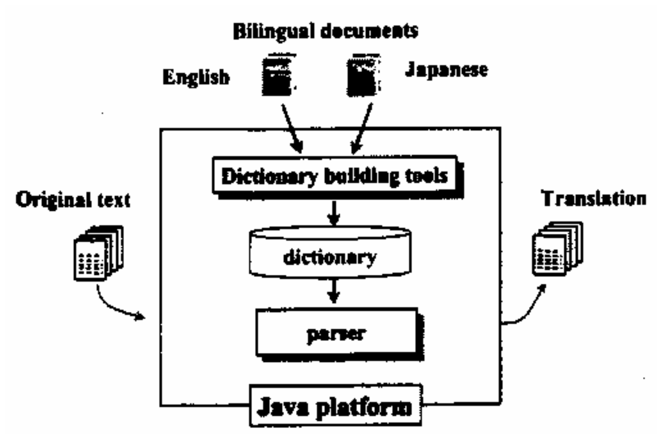


Figure 1: Overview of MT system PENSEE

API) for natural language processing systems and develop PENSÉE according to the API. Therefore the components of PENSÉE can be used smoothly in the other NLP applications.

3 Pattern-based Translation

As to translation, we employ a pattern-based transfer approach [Abeille et al. 90] [Takeda 96]. This approach uses a set of bilingual patterns shown in table 1. In the parsing process, trees are rewritten by applying the source patterns. Terminals and non-terminals *are* processed in the same framework but lexicalized patterns have priority over symbolized patterns [2]. A plausible parse tree will be selected among possible parse trees by the number of patterns applied. Then the parse tree is transferred into target language by using target patterns which are correspondent to the source patterns.

English	*Japanese*
S → 1:NP 2:VP	S → 1:NP が 2:VP
NP → a 1:N	NP → 1:N
VP → 1:V 2:NP	VP → 2:NP を 1:V
VP → take 1:NP	VP → 1:NP を する
VP → take a bath	VP → 風呂 に 入る
V → take	V → とる
N → bath	N → 風呂

Table 1: Examples of bilingual patterns

The pattern format is simple but highly descriptive. It can represent complicated linguistic phenomena and even correspondences between quite different structures in the languages. All the knowledge necessary for the translation, whether grammar rules or user dictionaries, will be compiled in the pattern format. It allows users to customize translations easily and effectively. Most commercial MT systems, including former PENSÉE series, limit the entries for user dictionaries in words and frozen expressions. Pattern-based translation, on the other hand, enable users to enter flexible expressions and even grammar rules as far as they follow the pattern format. Therefore, users can obtain high-quality translations by building user dictionaries (=patterns).

In the following section, we will discuss the importance of user customization and present dictionary building tools to support the user customization.

[2] We define a symbolized pattern as a pattern without a terminal and a lexicalized pattern as that with more than one terminal. PENSEE prepares 1000 symbolized patterns and 130.000 lexicalized patterns as a system dictionary.

4 Customization

4.1 Workflow

User dictionaries are developed by users to supplement or override the terms supplied in the system dictionaries. Users tend to use MT systems in very limited domains in which only one possible use of a word is necessary. Therefore, building a sufficient user dictionary is essential to a high-quality translation.

Development of a user dictionary should be automated as much as possible because it is one of the most time-consuming tasks for MT users. PENSÉE provides dictionary building tools, a vocabulary builder and a pattern editor, to support users in building their own dictionaries efficiently. The tools are used for the following purposes: the vocabulary builder extracts translation candidates automatically and the pattern editor helps users to make translation patterns.

Figure 2 is a workflow of dictionary building process. We assume that users are translators who need to make a considerable amount of translation in a specific domain and accumulate translation they have done. Before using the vocabulary builder and the pattern editor, English document and its Japanese correspondent should be aligned sentence by sentence. To align the sentences, we calculate similarity of word distribution and a document format such as blank spaces and indents [Sukehiro et al. 98]. Sentence alignment is started up when a user selects bilingual documents and click the "align" button on the display. Figure 3 and figure 4 show bilingual documents before and after the sentence alignment. Users can edit the result interactively if there is an irrelevant alignment.

4.2 Vocabulary Builder

Vocabulary builder extracts correspondences of word sequences in aligned bilingual documents. It is based on the method of finding correspondences proposed

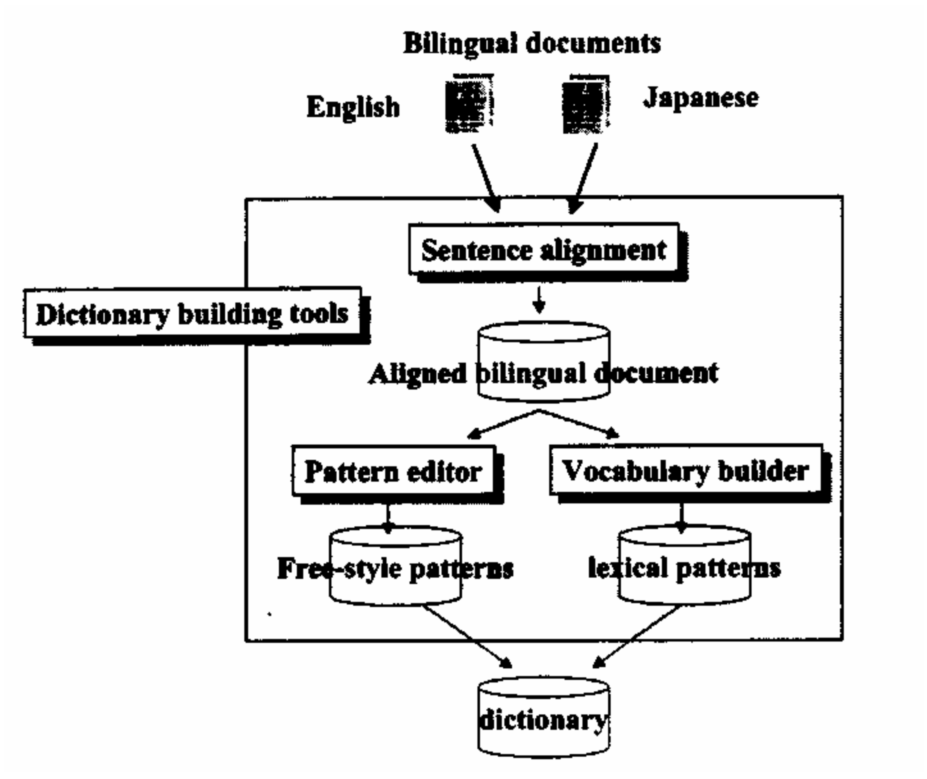
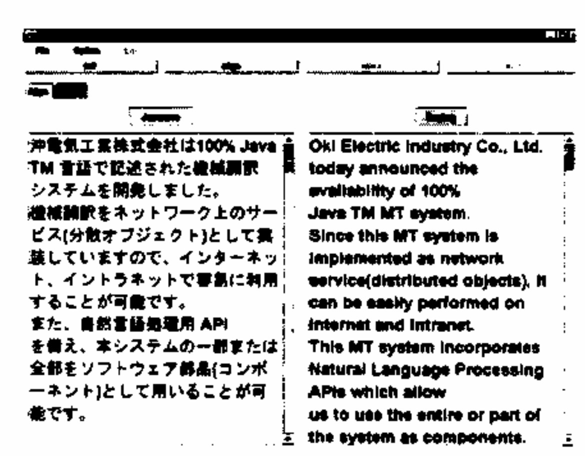


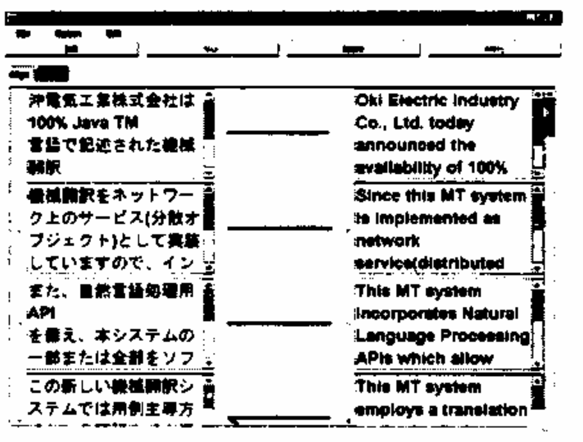
Figure 2: Workflow of dictionary building



沖電気工業株式会社は100% Java
TM 言語で記述された機械翻訳
システムを開発しました。
機械翻訳をネットワーク上のサー
ビス(分散オブジェクト)として実
装していますので、インターネッ
ト、イントラネットで容易に利用
することが可能です。
また、自然言語処理用 API
を備え、本システムの一部または
全部をソフトウェア部品(コンポ
ーネント)として用いることが可
能です。

Oki Electric Industry Co., Ltd.
today announced the
availability of 100%
Java TM MT system.
Since this MT system is
implemented as network
service(distributed objects), it
can be easily performed on
internet and intranet.
This MT system incorporates
Natural Language Processing
APIs which allow
us to use the entire or part of
the system as components.

Figure 3: Display before sentence alignment



沖電気工業株式会社は
100% Java TM
言語で記述された機械
翻訳

Oki Electric Industry
Co., Ltd. today
announced the
availability of 100%

機械翻訳をネットワー
ク上のサービス(分散オ
ブジェクト)として実装
していますので、イン

Since this MT system
is implemented as
network
service(distributed

また、自然言語処理用
API
を備え、本システムの
一部または全部をソフ

This MT system
incorporates Natural
Language Processing
APIs which allow

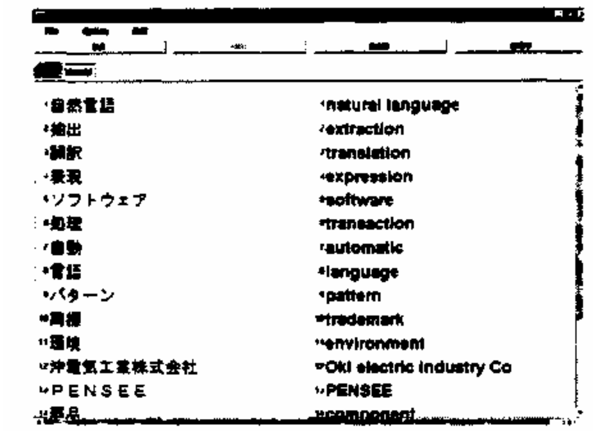
この新しい機械翻訳シ
ステムでは用例主導方

This MT system
employs a translation

Figure 4: Display after sentence alignment

by [Kitamura and Matsumoto 96]. Translation candidates of word sequences are evaluated by a similarity measure between the sequences defined by the co-occurrence frequency and independent frequency of the word sequence.

All a user has to do is to click the "learn" button after the sentence alignment. Then the vocabulary builder starts the process and displays the translation candidates. They are not word-to-word translations but the translations that reflect the word preference of the domain in which they are acquired. Figure 5 shows the result of vocabulary builder acquired from the aligned sentences in figure 4. Users are able to modify the result through GUI if it contains irrelevant word sequences. Then they enter the word list into the user dictionary by clicking the "entry" button.



1自然言語	1natural language
2抽出	2extraction
3翻訳	3translation
4表現	4expression
5ソフトウェア	5software
6処理	6transaction
7自動	7automatic
8言語	8language
9パターン	9pattern
10商標	10trademark
11環境	11environment
12沖電気工業株式会社	12Oki electric industry Co
13PENSEE	13PENSEE
14部品	14component

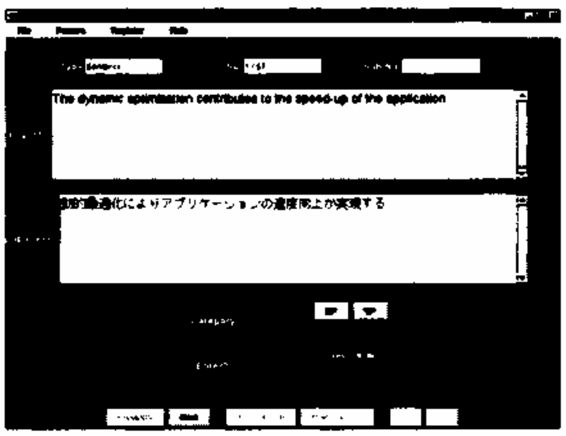
Figure 5: Result of vocabulary builder

According to [Kitamura and Matsumoto 96], the vocabulary builder extracts the correspondences with SO % accuracy, which depends on the volume of the documents. We think it will be so practical as to reduce users' work. The vocabulary builder is now used to extract correspondences of word sequences. But it will be improved to extract monolingual word sequences from a monolingual document and also correspondences of uncontinuous word sequences from bilingual documents [Shimohata et al. 99].

4.3 Pattern Editor

Pattern editor enables users to make translation patterns with simple operation. Users specify the words or phrases in certain English sentence and Japanese translation which can be symbolized by certain grammatical units. Then, translation patterns are generated from the sentences users have marked up.

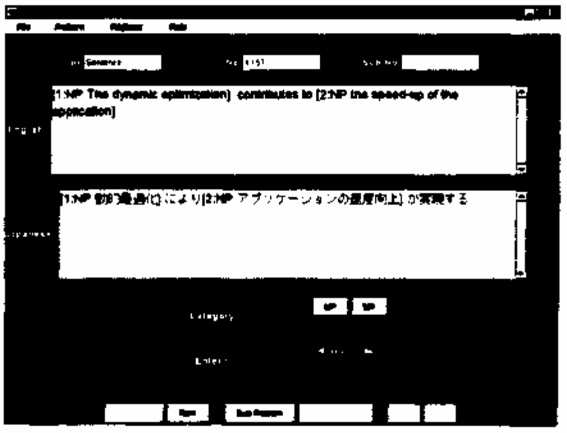
Take a sentence in figure 6, for example. First, "the dynamic optimization" is marked up by mouse dragging and identified as a noun phrase by clicking "NP" button. Next, "動的最適化", the translation of "the dynamic optimization", is marked up and identified as a noun phrase. Then, both phrases are given the same number to show that they are correspondent to each other. This process is repeated until no substitutive correspondence is remained. After the mark-up is finished as shown in figure 7 , the translation patterns are generated by clicking "yes" button. In this example, three translation patterns shown in table 2 are generated from the marked up sentences.



The dynamic optimization contributes to the speed-up of the application

動的最適化によりアプリケーションの速度向上が実現する

Figure 6: Display before pattern editing



[1:NP The dynamic optimization] contributes to [2:NP the speed-up of the application]

[1:NP 動的最適化]により[2:NP アプリケーションの速度向上] が実現する

Figure 7: Display after pattern editing

The pattern editor reduces both the time to make translation patterns and the careless mistakes such as lack of "]". But we think it is still time-consuming to make translation patterns manually. We are aiming to shift pattern making process to automatic extraction by extending the vocabulary builder. In that case, the pattern editor will be used as a supplementary tool.

English	*Japanese*
S → 1:NP contribute to 2:NP	S → 1:NP により 2:NP が 実現する
NP → the dynamic optimization	NP → 動的最適化
NP → the speed-up of the application	NP → アプリケーションの速度向上

Table 2: Generated patterns

5 Conclusion

We have described the current status of our new MT system PENSÉE. This system is designed to fit into the real translation environment. The most important improvement is that users can obtain translations as they expected with far less effort than they did. We believe that PENSÉE will make a great impact upon improving usability in both system developers and MT users.

References

[Gosling et al. 96] Gosling G., Joy B., and Steele G. (1996). "The Java(TM) Language Specification," published by Addison-Wesley.

[Abeille et al. 90] Abeille A., Schabes Y., and Joshi A. K. (1990). "Using Lexicalized Tags for Machine Translation". In proceedings of the International Conference on Computational Linguistics(COLING), pp. 1-6.

[Takeda 96] Takeda K. (1996). "Pattern-Based Context-Free Grammars for Machine Translation". In proceedings of 34th ACL, pp.144-151.

[Sukehiro et al. 98] Sukehiro T., Yamamoto H., and Sugio T. (1998). "Machine Translation System for Document Revision" (in Japanese). In Oki Research & Development, Vol.65, No.3, pp.233-236.

[Kitamura and Matsumoto 96] Kitamura M. and MatsumotoY. (1996). "Automatic Extraction of Word Sequence Correspondences in Parallel Corpora". In proceedings of Fourth Annual Workshop on Very Large Corpora(WVLC4), pp. 79-87.

[Shimohata et al. 99] Shimohata S., Sugio T., and Nagata J. (1999). "Retrieving Domain-Specific Collocations by Co-occurrences and Word Order Constraints". In Computational Intelligence, Vol 15, No.2, pp92-100.