# iChi: a bilingual dictionary generating tool

**Varga István**
Yamagata University,
Graduate School of Science and Engineering
dyn36150@dip.yz.yamagata-u.ac.jp

**Yokoyama Shoichi**
Yamagata University,
Graduate School of Science and Engineering
yokoyama@yz.yamagata-u.ac.jp

## Abstract

In this paper we introduce a bilingual dictionary generating tool that does not use any large bilingual corpora. With this tool we implement our novel pivot based bilingual dictionary generation method that uses mainly the WordNet of the pivot language to build a new bilingual dictionary. We propose the usage of WordNet for good accuracy, introducing also a double directional selection method with local thresholds to maximize recall.

## 1 Introduction

Bilingual dictionaries are an essential, perhaps even indispensable tool not only as resources for machine translation, but also in every day activities or language education. While such dictionaries are available to and from numerous widely used languages, less represented language pairs have rarely a reliable dictionary with good coverage. The need for bilingual dictionaries for these less common language pairs is increasing, but qualified human resources are scarce. Considering that in these conditions manual compilation is highly costly, alternative methods are imperative.

Pivot language based bilingual dictionary generation is one plausible such alternative (Tanaka and Umemura, 1994; Sjöbergh, 2005; Shirai and Yamamoto, 2001; Bond and Ogura, 2007). These methods do not use large bilingual corpora, thus being suitable for low-resourced languages.

Our paper presents iChi, the implementation of our own method, an easy-to-use, customizable tool that generates a bilingual dictionary.

The paper is structured as follows: first we briefly describe the methodological background of our tool, after which we describe its basic functions, concluding with discussions. Thorough description and evaluation, including comparative analysis, are available in Varga and Yokoyama (2009).

## 2 Methodological background

### 2.1 Pivot based dictionary generation

Pivot language based bilingual dictionary generation methods rely on the idea that the lookup of a word in an uncommon language through a third, intermediated language can be automated. Bilingual dictionaries to a third, intermediate language are used to link the source and target words. The pivot language translations of the source and target head words are compared, the suitability of the source-target word pair being estimated based on the extent of the common elements.

There are two known problems of conventional pivot methods. First, a global threshold is used to determine correct translation pairs. However, the scores highly depend on the entry itself or the number of translations in the intermediate language, therefore there is a variance in what that score represents. Second, current methods perform a strictly lexical overlap of the source-intermediate and target-intermediate entries. Even if the translations from the source and target languages are semantically transferred to the intermediate language, lexically it is rarely the case. However, due to the different word-usage or paraphrases, even semantically identical or very similar words can have different definitions in different dictionaries. As a result, because of the lexical characteristic of their overlap, current methods cannot identify the differences between totally different definitions resulted by unrelated concepts, and differences in only nuances resulted by lexicographers describing the same concept, but with different words.

### 2.2 Specifics of our method

To overcome the limitations, namely low precision of previous pivot methods, we expand the translations in the intermediate language using



*Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 217–220,
Suntec, Singapore, 4 August 2009. 

information extracted from *WordNet* (Miller et. al., 1990). We use the following information: *sense description, synonymy*, *antonymy* and *semantic categories*, provided by the tree structure of nouns and verbs.

To improve recall, we introduce *bidirectional selection.* As we stated above, the global threshold eliminates a large number of good translation pairs, resulting in a low recall. As a solution, we can group the translations that share the same source or target entry, and set *local thresholds* for each head word. For example, for a source language head word *entry_source* there could be multiple target language candidates: *entry_target*$_1$*, … ,entry_target*$_n$. If the top scoring *entry_target*$_k$ candidates are selected, we ensure that at least one translation will be available for *entry_source*, maintaining a high recall. Since we can group the entries in the source language and target language as well, we perform this selection twice, once in each direction. Local thresholds depend on the top scoring *entry_target*, being set to *maxscore*·*c*. Constant *c* varies between 0 and 1, allowing a small window for not maximum, but high scoring candidates. It is language and selection method dependent (See 3.2 for details).

### 2.3 Brief method description

First, using the source-pivot and pivot-target dictionaries, we connect the source (*s*) and target (*t*) entries that share at least one common translation in the intermediate (*i*) language. We consider each such source-target pair a *translation candidate*. Next we eliminate erroneous candidates. We examine the translation candidates one by one, looking up the source-pivot and target-pivot dictionaries, comparing pivot language translations. There are six types of translations that we label *A*-*F* and explain below as follows.

First, we select translation candidates whose translations into the intermediate language match perfectly (*type A* translations).

For most words WordNet offers *sense description* in form of synonyms for most of its senses. For a given translation candidate (*s,t*) we look up the source-pivot and target-pivot translations ($s \rightarrow I = \{s \rightarrow i_1, \ldots, s \rightarrow i_n\}$, $t \rightarrow I = \{t \rightarrow i_1, \ldots, t \rightarrow i_m\}$). We select the elements that are common in the two definitions ($I' = (s \rightarrow I) \cap (t \rightarrow I)$) and we attempt to identify their respective senses from WordNet ($sns(I')$), comparing each synonym in the WordNet's synonym description with each word from the pivot translations. As a result, we arrive at a certain set of senses from the source-pivot definitions ($sns((s \rightarrow I'))$) and target-pivot definitions ($sns((t \rightarrow I'))$). We mark $score_B(s,t)$ the Jaccard coefficient of these two sets. Scores that pass a global threshold (0.1) are selected as translation pairs. Since synonymy information is available for nouns (N), verbs (V), adjectives (A) and adverbs (R), four separate scores are calculated for each POS *(type B).*

$$score_B(s,t) = \max_{i' \in s \rightarrow I \cap t \rightarrow I} \frac{|sns(s \rightarrow i') \cap sns(t \rightarrow i')|}{|sns(s \rightarrow i') \cup sns(t \rightarrow i')|} \quad (1)$$

We expand the source-to-pivot and target-to-pivot definitions with information from WordNet (synonymy, antonymy and semantic category). The similarity of the two expanded pivot language descriptions gives a better indication on the suitability of the translation candidate. Since the same word or concept's translations into the pivot language also share the same semantic value, the extension with *synonyms* ($ext(l \rightarrow i) = (l \rightarrow i) \cup syn(l \rightarrow i)$, where $l = \{s,t\}$) the extended translation should share more common elements *(type C).*

In case of antonymy, we expand the initial definitions with the *antonyms of the antonyms* ($ext(l \rightarrow i) = (l \rightarrow i) \cup ant(ant(l \rightarrow i))$, where $l = \{s,t\}$). This extension is different from the synonymy extension, in most cases the resulting set of words being considerably larger *(type D).*

Synonymy and antonymy information are available for nouns, verbs, adjectives and adverbs, thus four separate scores are calculated for each POS.

*Semantic categories* are provided by the tree structure (hypernymy/hyponymy) of nouns and verbs of WordNet. We transpose each entry from the pivot translations to its semantic category ($ext(l \rightarrow i) = (l \rightarrow i) \cup semcat(l \rightarrow i)$, where $l = \{s,t\}$). We assume that the correct translation pairs share a high percentage of semantic categories.

Local thresholds are set based on the best scoring candidate for a given entry. The thresholds were *maxscore*·0.9 for synonymy and antonymy; and *maxscore*·0.8 for the semantic categories (see §3.2 for details).

$$score_{C,D,E}(s,t) = \frac{|ext(s \rightarrow i) \cap ext(t \rightarrow i)|}{|ext(s \rightarrow i) \cup ext(t \rightarrow i)|} \quad (2)$$

For a given entry, the three separate candidate lists of type C, D and E selection methods resulted in slightly different results. The good translations were among the top scoring ones, but not always scoring best. To correct this fault, a *combined selection* method is performed combining these lists. For every translation candidate we select the maximum score ($score_{rel}(s,t)$) from

the several POS (noun, verb, adjective and adverb for synonymy and antonymy relations; noun and verb for semantic category) based scores, multiplied by a multiplication factor (*mfactor*). This factor varies between 0 and 1, awarding the candidates that were selected both times during the double directional selection; and punishing when selection was made only in a single direction. $c_1$, $c_2$ and $c_3$ are adjustable language dependent constants, the defaults being 1, 0.5 and 0.8, respectively *(type F)*.

$$score_F(s,t) = \prod_{rel} \left( \begin{matrix} (c_1 + \max(score_{rel}(s,t))) \cdot \\ (c_2 + c_3 \cdot mfactor_{rel}(s,t)) \end{matrix} \right) \qquad (3)$$

## 2.4 Evaluation

We generated a Japanese-Hungarian dictionary using selection methods A, B and F; with C, D and E contributing indirectly through F.

### *(a) Recall evaluation*

We used a Japanese frequency dictionary that we generated from the Japanese EDR corpus (Isahara, 2007) to weight each Japanese entry. Setting the standard to the frequency dictionary (its recall value being 100), we automatically search each entry from the frequency dictionary, verifying whether or not it is included in the bilingual dictionary. If it is recalled, we weight it with its frequency from the frequency dictionary.

Our method maintains the recall value of the initial translation candidates, owing to the bidirectional selection method with local thresholds. However, the recall value of a manually created Japanese-English dictionary is higher than any automatically generated dictionary's value (Table 1).

| method | recall |
|---|---|
| our method | 51.68 |
| initial candidates | 51.68 |
| Japanese-English(*) | 73.23 |

Table 1: Recall evaluation results (* marks a manually created dictionary)

### *(b) 1-to-1 precision evaluation*

We evaluated 2000 randomly selected translation pairs, manually scoring them as *correct* (the translation conveys the same meaning, or the meanings are slightly different, but in a certain context the translation is possible: 79.15%), *undecided* (the translation pair's semantic value is similar, but a translation based on them would be faulty: 6.15%) or *wrong* (the translation pair's two entries convey a different meaning: 14.70%).

### *(c) 1-to-multiple evaluation*

With 1-to-multiple evaluation we quantify the true reliability of the dictionary: when looking up the meanings or translations of a certain keyword, the user, whether he's a human or a machine, expects all translations to be accurate. We evaluated 2000 randomly selected Japanese entries from the initial translation candidates, scoring all Hungarian translations as *correct* (all translations are correct: 71.45%), *acceptable* (the good translations are predominant, but there are up to 2 erroneous translations: 13.85%), *wrong* (the number or wrong translations exceeds 2: 14.70%).

# 3 iChi

iChi is an implementation of our method. Programmed in Java, it is a platform-independent tool with a user friendly graphical interface (Image 1). Besides the MySql database it consists of: iChi.jar (java executable), iChi.cfg (configuration file), iChi.log (log file) and iChip.jar (parameter estimation tool). The major functions of iChi are briefly explained below.

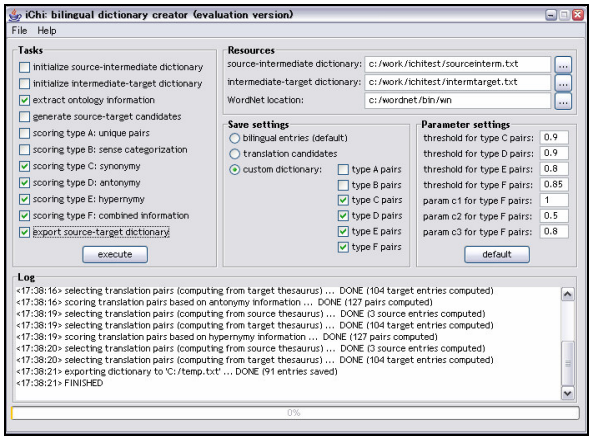


Image 1: User interface of iChi

## 3.1 Resources

The two bilingual dictionaries used as resources are text files, with a translation pair in each line:

```
source entry 1@pivot entry 1
source entry 2@pivot entry 2
```

The location of the pivot language's WordNet also needs to be specified. All paths are stored in the configuration file.

## 3.2 Parameter settings

iChip.jar estimates language dependent parameters needed for the selection methods. Its single argument is a text file that contains marked (correct: $+ or incorrect: $-) translation pairs:

```
$+source entry 1@correct target entry 1
$-source entry 2@incorrect target entry 2
```

The parameter estimation tool experiments with various threshold settings on the same (correct or incorrect) source entries. For example, with Hungarian-Japanese we considered all translation candidates whose Hungarian entry starts with "zs" (IPA: ʒ). 133 head words totalling 515 translation candidates comprise this set, 273 entries being marked as *correct*. iChip experimented with a number of thresholds to determine which ones provide with the best F-scores, e.g. retain most marked correct translations (Table 2). The F-scores were determined as follows: for example using synonymy information (type C) in case of threshold=0.85%, 343 of the 515 translation pairs were above the threshold. Among these, 221 were marked as correct, thus the precision being 221/343·100=64.43 and the recall being 221/273·100=80.95. F-score is the harmonic mean of precision and recall (71.75 in this case).

| **selection type** | **threshold value** (%) | | | | |
|---|---|---|---|---|---|
| | **0.75** | **0.80** | **0.85** | **0.90** | **0.95** |
| C | 70.27 | 70.86 | 71.75 | **72.81** | 66.95 |
| D | 69.92 | 70.30 | 70.32 | **70.69** | 66.66 |
| E | 73.71 | **74.90** | 72.52 | 71.62 | 65.09 |
| F | 78.78 | 79.07 | **79.34** | 78.50 | 76.94 |

Table 2: Selection type F-scores with varying thresholds (best scores in bold)

The output is saved into the configuration file. If no parameter estimation data is available, the parameters estimated using Hungarian-Japanese are used as default.

### 3.3 Save settings

The generated source-target dictionary is saved into a text file that uses the same format described in §3.1. The output can be customized by choosing the desired selection methods. The default value is a dictionary with selection types A, B and F; selection types C, D and E are used only indirectly with type F.

### 3.4 Tasks

The tasks are run sequentially, every step being saved in the internal database, along with being logged into the log file.

## 4 Discussion

If heavily unbalanced resources dictionaries are used, due to the bidirectional selection method many erroneous entries will be generated. If one polysemous pivot entry has multiple translations into the source, but only some of them are translated into the target languages, unique, but incorrect source-target pairs will be generated. For example, with an English pivoted dictionary that has multiple translation of *'bank'* onto the source (*'financial institution', 'river bank'*), but only one into the target language *('river bank')*, the incorrect source*('financial institution')*-target*('river bank')* pair will be generated, since target*('river bank')* has no other alternative.

Thorough discussion on recall and precision problems concerning the methodology of iChi, are available in Varga and Yokoyama (2009).

## 5 Conclusions

In this paper we presented iChi, a user friendly tool that uses two dictionaries into a third, intermediate language together with the WordNet of that third language to generate a new dictionary. We briefly described the methodology, together with the basic functions. The tool is freely available online (http://mj-nlp.homeip.net/ichi).

## References


Bond, F., Ogura, K. 2007. Combining linguistic resources to create a machine-tractable Japanese-Malay dictionary, *Language Resources and Evaluation*, 42(2), pp. 127-136.

Breen, J.W. 1995. Building an Electric Japanese-English Dictionary, *Japanese Studies Association of Australia Conference*, Brisbane, Queensland, Australia.

Isahara, H. (2007). EDR Electronic Dictionary – present status (EDR 電子化辞書の現状), NICT-EDR symposium, pp. 1-14. (in Japanese)

Miller G.A., Beckwith R., Fellbaum C., Gross D., Miller K.J. (1990). Introduction to WordNet: An Online Lexical Database, *Int J Lexicography* 3(4), pp. 235-244.

Sjöbergh, J. 2005. Creating a free Japanese-English lexicon, *Proceedings of PACLING*, pp. 296-300.

Shirai, S., Yamamoto, K. 2001. Linking English words in two bilingual dictionaries to generate another pair dictionary, *ICCPOL-2001*, pp. 174-179.

Tanaka, K., Umemura, K. 1994. Construction of a bilingual dictionary intermediated by a third language, *Proceedings of COLING-94*, pp. 297-303.

Varga, I., Yokoyama, S. 2009. Bilingual dictionary generation for low-resourced language pairs, *Proceedings of EMNLP 2009.*