Creating Reverse Bilingual Dictionaries

Khang Lam
Department of Computer Science
University of Colorado at Colorado Springs
Colorado Springs, USA
klam2@uccs.edu

Jugal Kalita
Department of Computer Science
University of Colorado at Colorado Springs
Colorado Springs, USA
jkalita@uccs.edu

Abstract

Bilingual dictionaries are expensive resources and not many are available when one of the languages is resource-poor. In this paper, we propose algorithms for creation of new reverse bilingual dictionaries from existing bilingual dictionaries in which English is one of the two languages. Our algorithms exploit the similarity between word-concept pairs using the English Wordnet to produce reverse dictionary entries. Since our algorithms rely on available bilingual dictionaries, they are applicable to any bilingual dictionary as long as one of the two languages has Wordnet type lexical ontology.

1 Introduction

The Ethnologue organization\(^1\) lists 6,809 distinct languages in the world, most of which are resource-poor. Most existing online bilingual dictionaries are between two resource-rich languages (e.g., English, Spanish, French or German) or between a resource-rich language and a resource-poor language. There are languages for which we are lucky to find a single bilingual dictionary online. For example, the University of Chicago hosts bilingual dictionaries from 29 Southeast Asian languages\(^2\), but many of these languages have only one bilingual dictionary online.

Existing algorithms for creating new bilingual dictionaries use intermediate languages or intermediate dictionaries to find chains of words with the same meaning. For example, (Gollins and Sanderson, 2001) use lexical triangulation to translate in parallel across multiple intermediate languages and fuse the results. They query several existing dictionaries and then merge results to maximize accuracy. They use four pivot languages, German, Spanish, Dutch and Italian, as intermediate languages. Another existing approach for creating bilingual dictionaries is using probabilistic inference (Mausam et al., 2010). They organize dictionaries in a graph topology and use random walks and probabilistic graph sampling. (Shaw et al., 2011) propose a set of algorithms to create a reverse dictionary in the context of single language by using converse mapping. In particular, given an English-English dictionary, they attempt to find the original words or terms given a synonymous word or phrase describing the meaning of a word.

The goal of this research is to study the feasibility of creating a reverse dictionary by using only one existing dictionary and Wordnet lexical ontology. For example, given a Karbi\(^3\)-English dictionary, we will construct an ENG-AJZ dictionary. The remainder of this paper is organized as follows. In Section 2, we discuss the nature of bilingual dictionaries. Section 3 describes the algorithms we propose to create new bilingual dictionaries from existing dictionaries. Results of our experiments are presented in Section 4. Section 5 concludes the paper.

2 Existing Online Bilingual Dictionaries

Powerful online translators developed by Google and Bing provide pairwise translations (including for individual words) for 65 and 40 languages, respectively. Wiktionary, a dictionary created by volunteers, supports over 170 languages. We find a

\(^1\)http://www.ethnologue.com/
\(^2\)http://dsal.uchicago.edu/dictionaries/list.html
\(^3\)Karbi is an endangered language spoken by 492,000 people (2007 Ethnologue data) in Northeast India, ISO 639-3 code AJZ. ISO 693-3 code for English is ENG.
large number of bilingual dictionaries at PanLex\(^4\) including an ENG-Hindi\(^5\) and a Vietnamese\(^6\)-ENG dictionary. The University of Chicago has a number of bilingual dictionaries for South Asian languages. Xobdo\(^7\) has a number of dictionaries, focused on Northeast India.

We classify the many freely available dictionaries into three main kinds.

- **Word to word dictionaries**: These are dictionaries that translate one word in one language to one word or a phrase in another language. An example is an ENG-HIN dictionary at Panlex.
- **Definition dictionaries**: One word in one language has one or more meanings in the second language. It also may have pronunciation, parts of speech, synonyms and examples. An example is the VIE-ENG dictionary, also at Panlex.
- **One language dictionaries**: A dictionary of this kind is found at dictionary.com.

We have examined several hundred online dictionaries and found that they occur in many different formats. Extracting information from these dictionaries is arduous. We have experimented with five existing bilingual dictionaries: VIE-ENG, ENG-HIN, and a dictionary supported by Xobdo with 4 languages: Assamese\(^8\), ENG, AJZ, and Dimasa\(^9\). We consider the last one to be a collection of 3 bilingual dictionaries: ASM-ENG, AJZ-ENG, and DIS-ENG. We choose these languages since one of our goals is to work with resource-poor languages to enhance the quantity and quality of resources available.

### 3 Proposed Solution Approach

A dictionary entry, called *LexicalEntry*, is 2-tuple `<LexicalUnit, Definition>`. A *LexicalUnit* is a word or a phrase being defined, also called *definiendum* (Landau, 1984). A list of entries sorted by the *LexicalUnit* is called a *lexicon* or a *dictionary*. Given a *LexicalUnit*, the *Definition* associated with it usually contains its class and pronunciation, its *meaning*, and possibly additional information. The meaning associated with it can have several *Senses*. A *Sense* is a discrete representation of a single aspect of the meaning of a word. Thus, a dictionary entry is of the form `<LexicalUnit, Sense\(_1\), Sense\(_2\), \ldots>`.

In this section, we propose a series of algorithms, each one of which automatically creates a reverse dictionary, or *ReverseDictionary*, from a dictionary that translates a word in language \(L_1\) to a word or phrase in language \(L_2\). We require that at least one of these two languages has a Wordnet type lexical ontology (Miller, 1995). Our algorithms are used to create reverse dictionaries from them at various levels of accuracy and sophistication.

#### 3.1 Direct Reversal (DR)

The existing dictionary has alphabetically sorted *LexicalUnits* in \(L_1\) and each of them has one or more *Senses* in \(L_2\). To create *ReverseDictionary*, we simply take every pair `<LexicalUnit, Sense>` in *SourceDictionary* and swap the positions of the two.

**Algorithm 1 DR Algorithm**

\[
\text{ReverseDictionary} := \emptyset \\
\text{for all } \text{LexicalEntry}_i \in \text{SourceDictionary} \text{ do} \\
\quad \text{for all } \text{Sense}_j \in \text{LexicalEntry}_i \text{ do} \\
\quad\quad \text{Add tuple } <\text{Sense}_j, \text{LexicalEntry}_i.\text{LexicalUnit}> \text{ to ReverseDictionary} \\
\quad \text{end for} \\
\text{end for}
\]

This is a baseline algorithm so that we can compare improvements as we create new algorithms. If in our input dictionary, the sense definitions are mostly single words, and occasionally a simple phrase, even such a simple algorithm gives fairly good results. In case there are long or complex phrases in senses, we skip them. The approach is easy to implement, and produces a high-accuracy *ReverseDictionary*. However, the number of entries in the created dictionaries are limited because this algorithm just swaps the positions of *LexicalUnit* and *Sense* of each entry in the *SourceDictionary* and does not have any method to find the additional words having the same meanings.
3.2 Direct Reversal with Distance (DRwD)

To increase the number of entries in the output dictionary, we compute the distance between words in the Wordnet hierarchy. For example, the words "hasta-lipi" and "likhavat" in HIN have the meanings "handwriting" and "script", respectively. The distance between "handwriting" and "script" in Wordnet hierarchy is 0.0, so that "handwriting" and "script" likely have the same meaning. Thus, each of "hasta-lipi" and "likhavat" should have both meanings "handwriting" and "script". This approach helps us find additional words having the same meanings and possibly increase the number of lexical entries in the reverse dictionaries.

To create a ReverseDictionary, for every LexicalEntry in the existing dictionary, we find all LexicalEntry\textsubscript{i}, i \neq j with distance to LexicalEntry\textsubscript{i} equal to or smaller than a threshold \(\alpha\). As results, we have new pairs of entries \(<\text{LexicalEntry}_i, \text{LexicalUnit}_i, \text{LexicalEntry}_j, \text{Sense}_j>\); then we swap positions in the two-tuples, and add them into the ReverseDictionary. The value of \(\alpha\) affects the number of entries and the quality of created dictionaries. The greater the value of \(\alpha\), the larger the number of lexical entries, but the smaller the accuracy of the ReverseDictionary.

The distance between the two LexicalEntries is the distance between the two LexicalUnits if the LexicalUnits occur in Wordnet ontology; otherwise, it is the distance between the two Senses. The distance between each phrase pair is the average of the total distances between every word pair in the phrases (Wu and Palmer, 1994). If the distance between two words or phrases is 1.00, there is no similarity between these words or phrases, but if they have the same meaning, the distance is 0.00.

We find that a ReverseDictionary created using the value 0.0 for \(\alpha\) has the highest accuracy. This approach significantly increases the number of entries in the ReverseDictionary. However, there is an issue in this approach. For instance, the word "tuhbi" in DIS means "crowded", "compact", "dense", or "packed". Because the distance between the English words "slow" and "dense" in Wordnet is 0.0, this algorithm concludes that "slow" has the meaning "tuhbi" also, which is wrong.

3.3 Direct Reversal with Similarly (DRwS)

The DRwD approach computes simply the distance between two senses, but does not look at the meanings of the senses in any depth. The DRwS approach represents a concept in terms of its Wordnet synset\textsuperscript{10}, synonyms, hyponyms and hypernyms. This approach is like the DRwD approach, but instead of computing the distance between lexical entries in each pair, we calculate the similarity, called simValue. If the simValue of a \(<\text{LexicalEntry}_i, \text{LexicalEntry}_j>, i \neq j\) pair is equal or larger than \(\alpha\), we conclude that the LexicalEntry\textsubscript{i} has the same meaning as LexicalEntry\textsubscript{j}.

To calculate simValue between two phrases, we obtain the ExpansionSet for every word in each phrase from the WordNet database. An ExpansionSet of a phrase is a union of synset, and/or synonym, and/or hypernym, and/or hyponym of every word in it. We compare the similarity between the ExpansionSets. The value of \(\alpha\) and the kinds of ExpansionSets are changed to create different ReverseDictionaries. Based on experiments, we find that the best value of \(\alpha\) is 0.9, and the best ExpansionSet is the union of synset, synonyms, hyponyms, and hypernyms. The algorithm for computing the simValue of entries is shown in Algorithm 3.

\textsuperscript{10}Synset is a set of cognitive synonyms.
Algorithm 3 \(\text{simValue}(\text{LexicalEntry}_i, \text{LexicalEntry}_j)\)

\[
\begin{align*}
simWords & := \emptyset \\
\text{if } & \text{LexicalEntry}_i.\text{LexicalUnit} \land \text{LexicalEntry}_j.\text{LexicalUnit} \text{ have a Word-} \\
\text{net lexical ontology} \text{ then} \\
\text{for all } (\text{LexicalUnit}_u \in \text{LexicalEntry}_i) \land (\text{LexicalUnit}_v \in \text{LexicalEntry}_j) \text{ do} \\
& \quad \text{Find } \text{ExpansionSet} \text{ of every} \\
& \quad \text{LexicalEntry} \text{ based on } \text{LexicalUnit} \\
\text{end for} \\
\text{else} \\
\text{for all } (\text{Sense}_u \in \text{LexicalEntry}_i) \land (\text{Sense}_v \in \text{LexicalEntry}_j) \text{ do} \\
& \quad \text{Find } \text{ExpansionSet} \text{ of every} \\
& \quad \text{LexicalEntry} \text{ based on } \text{Sense} \\
\text{end for} \\
\text{end if} \\
\text{simWords} & \leftarrow \text{ExpansionSet}(\text{LexicalEntry}_i) \cap \text{ExpansionSet}(\text{LexicalEntry}_j) \\\n\text{n} & \leftarrow \text{ExpansionSet}(\text{LexicalEntry}_i).\text{length} \\
\text{m} & \leftarrow \text{ExpansionSet}(\text{LexicalEntry}_j).\text{length} \\
\text{simValue} & \leftarrow \min\left\{\frac{\text{simWords}.\text{length}}{\text{n}}, \frac{\text{simWords}.\text{length}}{\text{m}}\right\}
\end{align*}
\]

4 Experimental results

The goals of our study are to create the high-precision reverse dictionaries, and to increase the numbers of lexical entries in the created dictionaries. Evaluations were performed by volunteers who are fluent in both source and destination languages. To achieve reliable judgment, we use the same set of 100 non-stop word ENG words, randomly chosen from a list of the most common words\(^{11}\). We pick randomly 50 words from each created ReverseDictionary for evaluation. Each volunteer was requested to evaluate using a 5-point scale, 5: excellent, 4: good, 3: average, 2: fair, and 1: bad. The average scores of entries in the ReverseDictionaries are presented in Figure 1. The DRwS dictionaries are the best in each case. The percentage of agreements between raters is in all cases is around 70%.

The DRwD approach significantly increases the number of entries, but the accuracy of the created dictionaries is much lower. The DRwS approach us-ing a union of synset, synonyms, hyponyms, and hypernyms of words, and \(\alpha \geq 0.9\) produces the best reverse dictionaries for each language pair. The DRwS approach increases the number of entries in the created dictionaries compared to the DR algorithm as shown in Figure 2.

Figure 1: Average entry score in ReverseDictionary

![Figure 1: Average entry score in ReverseDictionary](image)

Figure 2: Number of lexical entries in ReverseDictionaries generated from 100 common words

![Figure 2: Number of lexical entries in ReverseDictionaries generated from 100 common words](image)

We also create the entire reverse dictionary for the AJZ-ENG dictionary. The total number of entries in the ENG-AJZ dictionaries created by using the DR algorithm and DRwS algorithm are 4677 and 5941, respectively. Then, we pick 100 random words from the ENG-AJZ created by using the DRwS algorithm for evaluation. The average score of every entry in this created dictionary is 4.07.

5 Conclusion

We proposed approaches to create a reverse dictionary from an existing bilingual dictionary using Wordnet. We show that a high precision reverse dictionary can be created without using any other intermediate dictionaries or languages. Using the Wordnet hierarchy increases the number of entries in the created dictionaries. We perform experiments with several resource-poor languages including two that are in the UNESCO’s list of endangered languages.
References


