A Constraint Approach to Pivot-Based Bilingual Dictionary Induction

MAIRIDAN WUSHOUER, DONGHUI LIN, and TORU ISHIDA, Department of Social Informatics, Kyoto University
KATSUTOSHI HIRAYAMA, Graduate School of Maritime Sciences, Kobe University

High-quality bilingual dictionaries are very useful, but such resources are rarely available for lower-density language pairs, especially for those that are closely related. Using a third language to link two other languages is a well-known solution and usually requires only two input bilingual dictionaries *A*-*B* and *B*-*C* to automatically induce the new one, *A*-*C*. This approach, however, has never been demonstrated to utilize the complete structures of the input bilingual dictionaries, and this is a key failing because the dropped meanings negatively influence the result. This article proposes a constraint approach to pivot-based dictionary induction where language *A* and *C* are closely related. We create constraints from language similarity and model the structures of the input dictionaries as a Boolean optimization problem, which is then formulated within the Weighted Partial Max-SAT framework, an extension of Boolean Satisfiability (SAT). All of the encoded CNF (Conjunctive Normal Form), the predominant input language of modern SAT/MAX-SAT solvers, formulas are evaluated by a solver to produce the target (output) bilingual dictionary. Moreover, we discuss alternative formalizations as a comparison study. We designed a tool that uses the Sat4j library as the default solver to implement our method and conducted an experiment in which the output bilingual dictionary achieved better quality than the baseline method.

Categories and Subject Descriptors: I. [**Computing Methodologies**]; I.2.6 [**Learning**]: Induction

General Terms: Algorithms

Additional Key Words and Phrases: Bilingual dictionary induction, Weighted Partial Max-SAT, constraint satisfaction problem, low-resource languages, pivot language

ACM Reference Format:
Mairidan Wushouer, Donghui Lin, Toru Ishida, and Katsutoshi Hirayama. 2015. A constraint approach to pivot-based bilingual dictionary induction. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 15, 1, Article 4 (November 2015), 26 pages.
DOI: http://dx.doi.org/10.1145/2723144

1. INTRODUCTION

Bilingual dictionaries, machine-readable resources used to translate a word or phrase from one language to another, are essential for many tasks in Natural Language Processing (NLP), such as machine translation [Brown et al. 1990] and cross-lingual information retrieval [Nie et al. 1999]. However, high-quality bilingual dictionaries are only available for well-resourced language pairs, such as English–French or English–Chinese; they remain sparse, dated, or simply unavailable for less resourced language pairs like Uyghur and Kazakh. Hence, researchers have investigated the issue of

This research was partially supported by the Service Science, Solutions and Foundation Integrated Research Program from JST RISTEX, and a Grant-in-Aid for Scientific Research (S) (24220002) from the Japan Society for the Promotion of Science.

Authors' addresses: M. Wushouer, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan; email: mardan@ia.soc.i.kyoto-u.ac.jp; D. Lin and T. Ishida, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan; emails: {lindh, ishida}@i.kyoto-u.ac.jp; K. Hirayama, 5-1-1, Fukae-minamimachi, Higashinada-ku, Kobe 658-0022, Japan; email: hirayama@maritime.kobe-u.ac.jp.

DOI: http://dx.doi.org/10.1145/2723144

automatic creation of a bilingual dictionary. For example, a bilingual dictionary has been induced from a large-scale parallel corpus using subsentential alignment techniques [Wu and Xia 1994]. More recently, the use of comparable corpora (e.g., Wikipedia) has drawn increasing attention [Dou and Knight 2012; Yu and Tsujii 2009; Haghighi et al. 2008] since the Internet era has made monolingual data readily available[1] while parallel corpus remains scarce.

From the viewpoint of the etymological closeness of languages, some studies directly tackled the creation of dictionaries for closely related language pairs such as Spanish and Portuguese [Schulz et al. 2004] by using specific heuristics such as spelling. These studies, however, describe techniques that are not language transparent. Another well-known approach, pivot-based induction, uses a widespread language as a bridge between less resourced language pairs. Its naive implementation proceeds as follows. For each word in language *A*, take its translations to the pivot language using bilingual dictionary *A*-*B*, and then for each such pivot translation, take its translations to the language *C* using *B*-*C*. This implementation yields an extremely noisy bilingual dictionary containing incorrect translation pairs as lexicons are generally intransitive. This intransitivity stems from polysemic and ambiguous words in the pivot language. Take Uyghur–English–Kazakh as an example. The English word *tear* is the translation of Uyghur word *yash*, but only in the sense of liquid from the eyes. Further translating *tear* into Kazakh yields both the correct translation *jash* and an incorrect one, *jirtiw* (to rip).

To cope with the issue of divergence, previous studies attempted to select correct translation pairs by using semantic distances from the structures of the input dictionaries [Tanaka and Umemura 1994] or by using additional resources such as part of speech [Bond and Ogura 2008], WordNet [István and Shoichi 2009], comparable corpora [Kaji et al. 2008; Shezaf and Rappoport 2010], and descriptions present in dictionary entities [Sjobergh 2005]. Although the technique of adding resources to pivot-based induction is promising for improving performance [Shezaf and Rappoport 2010], a basic method that uses the structures of the input dictionaries must be developed because (1) it is essential for inadequately resourced languages, (2) it is compatible with other approaches and so can be combined [Mairidan et al. 2013; Saralegi et al. 2012], and (3) there is a potential for improving quality by considering the missing meanings [Saralegi et al. 2011].

There has been growing interest in using constraint optimization problem formalism for ideally describing and solving many problems in NLP and Web Service Composition [Matsuno and Ishida 2011; Ravi and Knight 2008; Hassine et al. 2006], because these problems are (or could be) reformed as combinatorial problems that can be represented by a set of variables connected upon constraints. For instance, the word sense ambiguity in machine translation has been resolved efficiently by a proposal of a consistent word selection method based on constraint optimization [Matsuno and Ishida 2011], in which the authors considered constraints between words in the document based on their semantic relation and contextual distance. Moreover, Ravi and Knight [2008] presented an application of optimization by solving substitution ciphers using low-order letter n-gram models, where the authors enforced global constraints using integer programming [Wolsey 1998] and guaranteed that no decipherment key is overlooked.

In this article, we propose a constraint-based method for pivot-based dictionary induction to promote the quality of output dictionary *A*-*C*, where *A* and *C* are closely related languages (intrafamily), while pivot language *B* is distant.[2] More precisely,

[1] Especially, it is easier to obtain in-domain monolingual corpora [Dou and Knight 2012].

[2] This limitation on language selection is not just because the closeness of languages is useful for detecting correct translation pairs, but also because the significant importance of bilingual dictionaries in making

we try to obtain semantic distance by constraining the types of connection in the structures of the input dictionaries based on a one-to-one assumption of intrafamily language lexicons. Furthermore, instances of pivot-based dictionary induction are represented by graphs, to which weighted edges are added to represent missing meanings. In this context, the Weighted Partial Max-SAT framework (WPMax-SAT), an optimization extension of Boolean satisfiability, is used to encode the graphs that will generate the optimally correctly output dictionary. Meanwhile, we discuss an alternative formalization within the 0-1 Integer Linear Programming framework (0-1 ILP) and its computation performance over WPMax-SAT as a comparison study. The reasons for using the WPMax-SAT framework as a primary formalization are that (1) the hidden facts such as whether a word pair is a correct translation or whether a meaning of pivot word is missing for a dictionary have binary states when they are unknown to the machine; (2) automatic detection of correct translation pairs and missing meanings whose states are bounded by certain weights can be seen as an optimization problem, which is to find the most reliably correct translation pairs, while adding the most probable missing meaning(s); and (3) the constraints inferred from language similarity can easily be transformed into propositional expressions. In other words, a new bilingual dictionary is created in the following steps:

First, we make an assumption: *lexicons of intrafamily languages are in one-to-one relation*, which allows any word in language A to have a unique translation equivalent in language C, or vice versa. Such a word pair is called a *one-to-one translation pair* (or *one-to-one pair*) in this article.

Second, to incorporate the input bilingual dictionaries, we use graphs, in which vertex is a word, and an edge is the indication of shared meaning. Following Mausam et al. [2009], we call these *transgraph*, which is defined to be the maximum scope for discovering the one-to-one pairs. Moreover, we treat transgraphs as being incomplete because some translations might not have been covered (missing) in input bilingual dictionaries when they were created. Hence, we allow for the automatic addition of edges to transgraphs, but only with certain costs, the chance that a particular edge is NOT missing (endpoint words are not a translation pair). This value is obtained by analyzing the structures of the transgraphs.

Third, the one-to-one assumption is used to constrain the word pair candidates of A and C languages in the transgraphs, and the candidates are recognized as one-to-one pairs only if they satisfy these constraints. Each transgraph is encoded as an optimization problem formulated within the Weighted Partial Max-SAT framework.

Finally, an iterative algorithm is created to extract one-to-one pairs by evaluating CNF formulas, in which, at each iteration, the CNF formula corresponding to a transgraph is evaluated to extract only a single one-to-one pair from the optimal assignment. This CNF formula is modified for the next iteration with reference to the awareness of the availability of one-to-one pair(s) accumulated from the previous iteration.

We designed a tool to implement the proposal using an open-source SAT library[3] as the default solver. With this tool, we evaluated our approach by inducing a Uyghur–Kazakh bilingual dictionary from Chinese–Uyghur and Chinese–Kazakh dictionaries; Uyghur and Kazakh are members of the Turkic language family, while Chinese is a Sino-Tibetan language. The evaluation result revealed the efficiency of our proposal in case of a related language pair; details can be found in Section 7.

machine translation systems for intrafamily language pairs has been claimed by recent research [Nakov and Ng 2012]. As to restricting pivot language to be distant, we consider the likeliness of having more information from the structures of the input dictionaries.

[3] http://www.sat4j.org.

The rest of the article is organized as follows: In Section 2, we discuss existing works on dictionary creation. Section 3 briefly introduces dictionary induction and the pivot-based technique. Sections 4 and 5 describe the proposed method with Max-SAT formalization, while in Section 6, an alternative formulization is discussed. An experiment and analyses are covered in Section 7. Finally, we end with the conclusion.

2. RELATED WORK

2.1. Pivot Language and Bilingual Dictionary

A very early attempt to create bilingual dictionaries from existing dictionaries was by Tanaka and Iwasaki [1996], who used a pivot language. They used Inverse Consultation (IC) to tackle lexical intransitivity divergence. IC tries to measure the intersection of two pivot word sets: the set of pivot translations of a word w in language A, and the set of pivot translations of each word in language C, a candidate for a translation of w. The number of elements in the intersection indicates the nearness of the original word and its candidate. IC generally requires that the intersection contain at least two synonymous words. For example, the intersection of the English translations of French *printemps* and Spanish *resorte* contains only a single word, *spring*. The intersection for the correct translation pair *printemps* and *primavera* will include two synonymous words, *spring* and *springtime*. If only one pivot word is present in the intersection, the equivalent candidate is considered to be inadequate and could be discarded if the consultation limit is equal to one. If it is two, further consultation could be conducted: Spanish translation set of the French *resorte* is generated through English pivot words, and how many times the Spanish *resorte* is repeated in this set is determined. The French *printemps* will be discarded, or subjected to further ranking if it is repeated more than once. In the IC approach, inverse consultation can be conducted n times. A weakness of the IC method is that it relies on synonymous words to identify correct translations. In the previous example, if the relatively rare *springtime* did not exist or was missing from the input bilingual dictionaries, IC would not have been able to detect that *primavera* is a correct translation, which may result in low recall.

With the assumption that using more than one pivot language could provide more information to evaluate semantic distance of the cross-lingual pairs in the output dictionary, one method uses multiple input bilingual dictionaries [Mausam et al. 2009]. They represent the input bilingual dictionaries as an undirected graph, where vertices represent the words from all the input dictionaries, and edges represent translation pairs. The new translation pairs are induced based on cycles in the undirected graph, where a cycle indicates that there are multiple paths between a pair of words in different languages. In the previous example, if both English and German are used as pivots, *printemps* and *primavera* would be accepted as a correct translation pair because they are linked by both *spring* in English and *Fruehling* in German, while *printemps* and *resorte* are not linked by any German pivot. This multiple-pivot idea is similar to the IC method, but its use of multiple pivot languages eliminates IC's dependency on synonym-rich input bilingual dictionaries to some extent. However, the new problem is the need to find suitable multiple input dictionaries.

One work [Skoumalova 2001] presented a method for inducing a bilingual dictionary of cross-family languages based on transduction models of cognate pairs via bridge languages, where the bilingual dictionaries within language families are induced using probabilistic string edit distance models. Bilingual dictionaries for arbitrary distant language pairs are then generated by a combination of these intrafamily translation models and one or more cross-family online dictionaries. They reported relatively high precision when generating English–Portuguese, English–Norwegian, and English–Ukrainian bilingual dictionaries.

2.2. Parallel Corpus and Monolingual Corpora

Another line of research is extracting a bilingual dictionary from a large amount of parallel corpus or comparable corpora. This has been generally done in two ways: (1) Using cross-lingual co-occurrences obtained from parallel corpus or monolingual corpora to improve the bilingual dictionaries generated by using pivot language [Kaji and Aizono 1996; Bond et al. 2001; Ahn and Frampton 2006; Saralegi et al. 2011; Tanaka and Iwasaki 1996]. For example, Schafer and Yarowsky [2002] created bilingual dictionaries between English and a target local language (e.g., Gujarati) using a related language (e.g., Hindi) as the pivot. In their case, English pivot words were used in conjunction with the cognates of two closely related languages. Cross-lingual co-occurrences were used to remove errors, together with other cues such as edit distance and IDF scores. (2) Extraction from parallel corpus or monolingual corpora without using the pivot language [Nerima and Wehrli 2008; Sahlgren and Karlgren 2005].

Using parallel corpus to obtain word-oriented machine-readable bilingual dictionaries is a mature and very promising approach if adequate amounts of data are available. The general implementation aligns the texts to each other at the chunk or word level. The alignment is usually evaluated by consistency (source words should be translated to a small number of target words relative to the entire corpus) and minimum shifting (the source must be aligned to a nearby translation). However, parallel corpus is expensive and generally not available for every domain.

Relying on readily available monolingual corpora is an alternative with growing research interest. In this context, most researches were inspired by Fung [1998] and Rapp [1999]. Their main assumption is that the term and its translation share similar contexts. These methods consist of two steps: modeling of contexts and measuring the similarity between the contexts of two languages using a seed dictionary. The majority of approaches follow the bag-of-words paradigm and represent contexts as weighted collections of words using LL [Ismail and Manandhar 2010], TF-IDF [Fung 1998], or PMI [Shezaf and Rappoport 2010]. Furthermore, Haghighi et al. [2008] characterized word types in each language by multiple monolingual features, such as context counts and orthographic substrings. The translations are induced using a generative model based on canonical correlation analysis. Such settings have been considered in other works, most notably in Koehn and Knight [2002] and Fung [1995], but Haghighi et al. [2008] was the first to use a probabilistic model and present results across a variety of language pairs and data conditions. In their experiment to create an English–Spanish dictionary, relatively high precision (89.0%) but a very low recall (33%) were reported. However, this approach was revealed to have low efficiency with distant language pairs (such as English–Chinese) due to its heavy dependence on orthographic features of languages. One work [Bergsma and Van Durme 2011], considering the fact that a large number of annotated images are added to sites like Facebook and Flickr every month, efficiently eases the dependence on orthographic similarities while improving the performance of bilingual dictionary extraction by using labeled web images: cross-lingual pairs of words are proposed as translations if their corresponding images have similar visual features. They concluded that visual similarity provides substantial gains over orthographic similarity alone.

Recently, there has been growing interest in decipherment work [Dou and Knight 2012; Ravi and Knight 2008; Nuhn et al. 2012], in which researchers treat training and statistical-based translation without parallel data as a deciphering problem. It is worth mentioning these works because they produce bilingual dictionaries (more precisely, a probabilistic translation table) as a byproduct by training statistical machine translation with monolingual corpora. Dou and Knight [2012] used decipherment to learn a domain-specific translation table and use it to improve out-of-domain translations, because although it is hard to find parallel corpora for any specific domain, it is relatively easy to find domain-specific monolingual corpora. Their probabilistic decipherment model treats Spanish words as a simple word substitution for its French counterparts and learns a Spanish–French word-to-word lexicon via decipherment on domain-specific monolingual corpora, although their decipherment model does not assume deterministic relation and is able to discover multiple translations. Meanwhile, a similar work [Nuhn et al. 2012] took a big step toward large-scale and large-vocabulary unsupervised training of statistical translation models with large-scale monolingual corpora, breaking the constraints in vocabulary or data size that previous approaches faced.

Most methods for bilingual dictionary creation are promising when extra language resources are available for the given language pair to acquire word sense or to evaluate semantic distance between cross-lingual word pairs. The sole work to try to create bilingual dictionaries purely from two input dictionaries [Tanaka and Umemura 1994] utilizes pivot synonymous words as the only information supporting semantic distance (hence, it is often seen as a baseline method in evaluations). In our work, we also focus on creating a bilingual dictionary from just two input dictionaries, since there are still many world languages that lack useful languages resources. Our approach, modeling complete structures of input dictionaries as a constraint optimization problem to handle the incompleteness of input dictionaries to some extent, performs well when the target languages are closely related. Moreover, as language resources are gradually being accumulated for inadequately resourced language pairs, methods to combine many language resources to produce a bilingual dictionary are becoming promising [Mairidan et al. 2013]. In this sense, our approach can also be adopted as a useful heuristic for extracting semantic information from the bilingual dictionary resources available.

3. PIVOT-BASED BILINGUAL DICTIONARY INDUCTION

3.1. Using Pivot Language

Let $D_{l_1-l_2}$ denote the dictionary of l_1 and l_2 languages. It is a relation linking any word in one language to one or more words in the other language. The creation of a bilingual dictionary can be done manually or automatically. If the latter, it is the process of determining whether a word in one language has the same meaning as a word in the another language.

Using a pivot language is well known in research on machine translation [Tanaka et al. 2009] and service computing since a large number of language resources are being accumulated as web services, and the recent service computing technologies allow us to utilize existing services to create new composite services [Ishida 2011]. However, in this context, the pivot-based induction is used to induce new dictionary $D_{\text{A-C}}$ from existing $D_{\text{A-B}}$ and $D_{\text{B-C}}$, where a pair of words in languages A and C is added to $D_{\text{A-C}}$ if they have the same translation in B. Such a $D_{\text{A-C}}$ may include both correct and incorrect translation pairs. More precisely, if pivot word w^{B} is a translation of w^{A} with respect to a sense s and w^{C} is a translation of w^{B} with respect to the same sense s, we can say that w^{C} is a translation of w^{A}. This approach is based on the assumption of the transitive relation of translation pairs in two languages (see Figure 1(a) and Figure 1(d)). Assume that we are seeking translations of words in language A to those in language C using $D_{\text{A-B}}$ and $D_{\text{B-C}}$. If pivot word w^{B} is a translation of w^{A} in $D_{\text{A-B}}$ and w^{C} is a translation of w^{B} in $D_{\text{B-C}}$, we could say that w^{C} is hence a translation of w^{A}.

Note that this deduction is not correct because it does not take into account the word sense: in Figure 1(b), w^{C} (case of w_2^{C}) can be the translation of w^{B} (w_1^{B}) for sense s (s_3) different from the sense for which w^{B} (w_1^{B}) is the equivalent of w^{A} (w_1^{A}). This can happen when pivot word w^{B} is polysemous or ambiguous, and such entries are often present

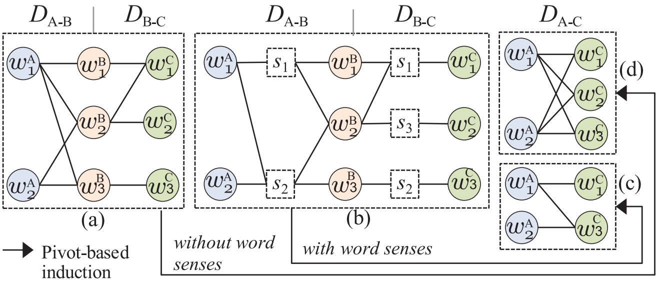


Fig. 1. Pivot-based bilingual dictionary induction and ambiguity problem.

in input bilingual dictionaries [Saralegi et al. 2011]. Taking Uyghur–English–Kazakh as an example, the English word *tear* is the translation of the Uyghur word *yash*, but only in the sense of liquid from the eyes. Further translating *tear* into Kazakh yields both the correct translation *jash* and an incorrect one, *jirtiw* (to rip). Identifying such an incorrect translation is challenging (see Figure 1), because, unfortunately, most dictionaries lack comparable information about senses in their entries. So it is not possible to map entries and translation equivalents according to their corresponding senses. As an alternative, most previous studies try to guide this mapping according to semantic distances extracted from the dictionaries themselves or external resources.

One can create a dictionary of two languages just by propagating their lexicons. This dictionary would have the highest recall and lowest precision. It is important to note that the basic pivot approach is often the first and easiest step to increasing the precision of such a dictionary. In many cases, the precision obtained from the first step is so low that the resulting dictionary is impractical.

3.2. Translation Graph

Merging two input bilingual dictionaries $D_{\text{A-B}}$ and $D_{\text{B-C}}$ via language B forms a big graph whose vertices are words and whose edges are the indication of common meaning between endpoint words. Such a graph has at least one connected component [Hopcroft and Tarjan 1973]—a subgraph in which any two vertices are connected to each other, and which is connected to no additional vertices in the supergraph—such as the one shown in Figure 1(a). We, following Mausam et al. [2009], call each connected component a *transgraph*.

Definition 1. A transgraph is defined as an undirected graph $G = \{V, E\}$, in which a vertex $w_i^l \in V$ is a word in language $l \in \{A, B, C\}$, and an edge $e(w_i^{\text{A}}, w_j^{\text{B}}) \in E$ or $e(w_h^{\text{C}}, w_j^{\text{B}}) \in E$ represents the belief that the word w_i^{A} and w_h^{C} shares at least one meaning with pivot word w_j^{B} , while $\neg e(w_i^{\text{A}}, w_j^{\text{B}})$ and $\neg e(w_h^{\text{C}}, w_j^{\text{B}})$ express the belief that there is no meaning in common. We also use V^{A}, V^{B}, and V^{C} to denote all the words (in terms of a *transgraph*) in language A, B, and C, respectively. For further use, $V_{w_i^{l1}}^{l2}$ denotes a set of meanings of $w_i^{l_1}$ in language l_2.

It should be noted that although $D_{\text{A-B}}$ and $D_{\text{B-C}}$ are usually directional, for example, $D_{\text{A-B}}$ was made with the intention of translating words in language A to language B, ignoring directionality is possible, because not only is this in accordance with the *reversibility principle* found in lexicographic literature [Tomaszczyk 1986] but also the initial noisy dictionary, $D_{\text{A-C}}$, would provide the most complete candidate set possible. However, one is allowed to merge $D_{l_1\text{-}l_2}$ and $D_{l_2\text{-}l_1}$ if they are available in order to increase

Table I. The Lexicostatistical Matrix of Turkic Languages

(%)	Kyrgyz	Kazakh	Uzbek	Uyghur	Tatar	Turkmen	Azeri
Kazakh	92						
Uzbek	82.9	82.8					
Uyghur	83.8	81.9	86.3				
Tatar	83.9	82.1	78	79.6			
Turkmen	71.2	71.9	75.9	71.7	69.8		
Azeri	66.9	67.8	70	68.8	68.4	78.2	
Turkish	64.9	64.8	67.2	66.7	65.6	73.6	86

the coverage of the output dictionary. However, doing so apparently creates additional ambiguities. The small transgraph in Figure 1(a) is used as an example in this article.

4. PROPOSED METHOD

4.1. One-to-One Assumption

The key characteristic of intrafamily languages is that their lexicons are similar and share a significant number of cognates—words that are derived from the same origin and are similar in both spelling and meaning (e.g., *neveu*[Fr.] and *nephew* [Eng.]). Most cognate pairs have direct translations,[4] meaning that they correspond one to one.

A classical lexicostatistical study of 15 Turkic languages,[5] mostly used in central Asia, indicated that cognate pairs shared among members of the Turkic language family scales from 44% to 94% of their lexicons, and the majority of noncognates tend to be nouns. Although there are some studies on constructing a bilingual dictionary for intrafamily languages by recognizing these cognate pairs, they often turned out to be language-pair dependent since subsequent separate phonetic development of languages has made many cognates not identical in spelling. Our work, however, aims at generating a relatively complete bilingual dictionary of intrafamily language pairs.

Taking account of such facts, we make the following assumption: *lexicons of intrafamily languages offer one-to-one relation*. That is, if A and C are intrafamily, for any w_i^{A} there exists a unique w_j^{C}, such that they have exactly the same meaning. Such a pair is called a one-to-one pair and denoted by $\mathbb{O}(w_i^{\mathrm{A}}, w_j^{\mathrm{C}})$. Accordingly, $\neg\mathbb{O}(w_i^{\mathrm{A}}, w_j^{\mathrm{C}})$ denies, logically, the state of one-to-one relation. We sometimes use the term one-to-one pair candidate to refer to a pair of words whose state of one-to-one relation has yet to be determined.

Although such an assumption may be too strong for the general case, we consider it reasonable for the case of intrafamily languages, although the evolution of languages has different situations and is reached in different states. This may result in a less accurate dictionary in applying the proposed algorithm in other language pairs that are closely related. However, utilizing a one-to-one assumption can also be found in existing studies. For example, Melamed [1997] made a similar assumption for any language pair in trying to create a word-to-word model of translation. They presented a fast method for inducing accurate translation lexicons from a parallel corpus. This method assumes that words are translated one to one. They claim that such an assumption reduces the explanatory power of their model in comparison to the IBM models and also helps them to avoid indirect associations, a major source of errors in translation models. Koehn and Knight [2002] also made a similar assumption when they extracted an English–German dictionary from monolingual corpora. Another relevant line of using the one-to-one criterion is extracting cognates of closely related languages and using

[4] A direct translation [Melamed 2000] is an association between two words where these two words are indeed mutual translations.

[5] http://turkic-languages.scienceontheweb.net.

them to adapt resources (such as parallel corpus) from one language to another. For example, Hana et al. adapted Spanish resources to Brazilian Portuguese to train a part-of-speech tagger [Hana et al. 2006]. Moreover, creating and using one-to-one translation equivalents are often tackled in translation between dialects of the same language, for example, between Cantonese and Mandarin [Zhang 1998], or between a dialect of a language and a standard version of that language, for example, between some Arabic dialect (e.g., Egyptian) and Modern Standard Arabic [Bakr et al. 2008; Sawaf 2010; Salloum and Habash 2011].

In conclusion, our goal is not to create bilingual dictionaries with word-to-word relation but accurate bilingual dictionaries of closely related language pairs (such as Turkic languages), so that the result might be of use in translation systems like Apertium (an open-source machine translation platform at http://www.apertium.org/), which uses bilingual dictionaries and manual rules to translate between related languages, including Spanish–Catalan, Spanish–Galician, Occitan–Catalan, and Macedonian–Bulgarian. The one-to-one assumption in our proposal is a tool whose aim is higher precision while preventing significant drop in recall. However, it is important to handle exceptions by proper postprocessing, which we try to explore in later works.

The one-to-one assumption is realized as constraints that are imposed on pair (w_i^{A}, w_j^{C}) in a transgraph; all constraints must be satisfied before the pair is recognized as a one-to-one pair. The next section details the constraints.

4.2. Constraints and Formalization

In a *transgraph*, the one-to-one assumption is realized with two constraints, one of which demands symmetric connection entitles while the other guarantees their uniqueness. Actually, selecting candidates of one-to-one pairs also can be seen as a constraint, which is independently defined in this article.

4.2.1. One-to-One Pair Candidate. Theoretically, any word w_i^{A} can be a one-to-one equivalent to any w_j^{C} in a transgraph (or even in the lexicons) when it is unknown to the machine. As the initial step of pivot-based techniques, the possible translation pairs are selected to generate a noisy $D_{\text{A-C}}$ based on the structures of the input dictionaries. In our work, we also take such a step, so that whether word pair (w_i^{A}, w_j^{C}) can be a one-to-one pair candidate is decided by the following constraint.

Constraint 1 (*Candidate Existence*): *A pair of words, w_i^{A} and w_j^{C}, in a transgraph can be a one-to-one pair candidate if they are connected via at least one pivot word.*

That is, a word pair is taken to be a candidate and subjected to further evaluation only if they share at least one word in the pivot language. For instance, in Figure 1(a), the candidates are all six possible combinations between $\{w_1^{\text{A}}, w_2^{\text{A}}\}$ and $\{w_1^{\text{C}}, w_2^{\text{C}}, w_3^{\text{C}}\}$. This constraint may raise doubts on the potential one-to-one pair candidates that are hidden because of data incompleteness (missing pivot words or meanings). However, we ignore this for simplicity. Recall that $V_{w_i^{\text{A}}}^{\text{B}}$ and $V_{w_j^{\text{C}}}^{\text{B}}$ are the sets of meanings of w_i^{A} and w_j^{C} in language B, respectively. This constraint can be expressed mathematically by the following propositional expression, which states that if two sets, $V_{w_i^{\text{A}}}^{\text{B}}$ and $V_{w_j^{\text{C}}}^{\text{B}}$, have no member in common, then w_i^{A} and w_j^{C} should never be taken to be a one-to-one pair candidate:

$$\left(V_{w_i^{\text{A}}}^{\text{B}} \cap V_{w_j^{\text{C}}}^{\text{B}} = \emptyset\right) \rightarrow \neg\mathbb{O}\left(w_i^{\text{A}}, w_j^{\text{C}}\right). \tag{1}$$

4.2.2. Symmetry. A one-to-one pair is a pair of words that carry exactly the same meanings. This allows us to define the following constraint on one-to-one pairs.

Constraint 2 (*Symmetry*): *Given a pair of words, w_i^{A} and w_j^{C}, in a transgraph, if they are a one-to-one pair, then they should be symmetrically connected through pivot word(s).*

In other words, a one-to-one pair must share the same words in pivot language; the number of edges between w_i^{A} and pivot words should equal the number of edges between w_j^{C} and pivot words. Note that a path through a pivot word might maintain at least one common word sense along the edges. This constraint is written in the following propositional expression:

$$\mathbb{O}\left(w_i^{A}, w_j^{C}\right) \rightarrow \bigwedge_{w_h^{B} \in V^{B}_{w_i^{A} w_j^{C}}} \left[e\left(w_i^{A}, w_h^{B}\right) \wedge e\left(w_j^{C}, w_h^{B}\right)\right], \tag{2}$$

where $V^{B}_{w_i^{A}}$ and $V^{B}_{w_j^{C}}$ are the sets of meanings of w_i^{A} and w_j^{C} in language B, and $V^{B}_{w_i^{A} w_j^{C}} = V^{B}_{w_i^{A}} \cup V^{B}_{w_j^{C}}$. For example, in Figure 1(a), if (w_1^{A}, w_1^{C}) is one-to-one pair, then six edges, $e(w_1^{A}, w_1^{B})$, $e(w_1^{A}, w_2^{B})$, $e(w_1^{A}, w_3^{B})$, $e(w_1^{C}, w_1^{B})$, $e(w_1^{C}, w_2^{B})$, and $e(w_1^{C}, w_3^{B})$, must exist in the transgraph, where $e(w_1^{C}, w_3^{B})$ is, indeed, not present. We consider that such an edge may be missing, which means that the corresponding translation might not have been included in the input dictionary when it was built.

4.2.3. Uniqueness. Another consequence of one-to-one assumption is that the translation pairs of intrafamily languages should be unique, meaning any w_i^{A} can have only a single one-to-one equivalent in language C, and vice versa. It is possible that synonymous words in a language can share exactly the same meaning(s) and can be used as alternates in the translation. Such synonymous words apparently can be one-to-one translation equivalent to the same word, but since we are looking for a single equivalent under the one-to-one assumption, we need to prevent the selection of multiple equivalents. This needs a constrain, which can be stated as follows.

Constraint 3 (*Uniqueness*): *Given a pair of words, w_i^{A} and w_j^{C}, in a transgraph, if they are a one-to-one pair, then they should be unique, such that all other candidates involving w_i^{A} or w_j^{C} are not one-to-one pairs.*

For example, in Figure 1(a), if (w_1^{A}, w_1^{C}) is a one-to-one pair, then we assert that (w_1^{A}, w_2^{C}), (w_1^{A}, w_3^{C}), and (w_2^{A}, w_1^{C}) are not one-to-one pairs. This constraint is written as the following propositional expression:

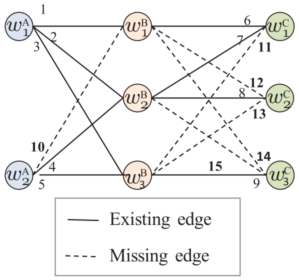
$$\mathbb{O}\left(w_i^{A}, w_j^{C}\right) \rightarrow \left[\bigwedge_{h \neq j} \neg\mathbb{O}\left(w_i^{A}, w_h^{C}\right)\right] \wedge \left[\bigwedge_{p \neq i} \neg\mathbb{O}\left(w_p^{A}, w_j^{C}\right)\right], \tag{3}$$

where the first AND operation iterates over the words w_h^{C} (excluding the given w_j^{C}) that are one-to-one candidates of w_i^{A}, and, likewise, the last AND operation iterates over the words w_p^{A} (excluding the given w_i^{A}) that are one-to-one candidates of w_j^{C}.

4.3. Data Incompleteness

The completeness of input dictionaries is seldom guaranteed: (1) a pivot word is missing so that some translation pair for $D_{A\text{-}C}$ is not identified, (2) a nonpivot word is missing (vertex $w_i^{A} \in V^{A}$ or $w_i^{C} \in V^{C}$ is missing in a transgraph), or (3) a translation (w_i^{A}, w_j^{B}) or (w_h^{C}, w_j^{B}) is missing (an edge is missing in a transgraph).

Apparently, the first two problems cannot be resolved without additional resources, so they are not considered further in this article. The third one, however, is vital because any missing edge may break a symmetric connection between w_i^{A} and w_j^{C}, so that $\mathbb{O}(w_i^{A}, w_j^{C})$ could not be detected as a one-to-one pair. This could harm the quality



An example where $e_{11} = e(w_1^C, w_3^B)$:

$$Weight(e_{11}) = P(w_1^A, w_1^C) = P' \cdot P(w_1^A|w_1^C) \cdot (w_1^C|w_1^A) = 0.666 \times 0.750 \times 0.500 = 0.250$$

where: $$P(w_1^A|w_1^C) = \sum_{i=1}^{2} P(w_1^A|w_i^B) \cdot (w_i^B|w_1^C) = \left[1 \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2}\right] = 0.750$$

$$P(w_1^C|w_1^A) = \sum_{i=1}^{2} P(w_1^C|w_i^B) \cdot (w_i^B|w_1^A) = \left[1 \times \frac{1}{3} + \frac{1}{2} \times \frac{1}{3}\right] = 0.500$$

Accordingly:

Edge	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}
Weight	0.166	0.250	0.056	0.083	0.111	0.166

Fig. 2. A transgraph with possible missing edges and their weights. Note that dotted lines represent the missing edges, $e_{10} \sim e_{15}$; $P(w_i^{l_1}|w_j^{l_2})$ is the probability that $w_i^{l_1}$ is the translation of $w_j^{l_2}$; and P' is the maximum probability of w_i^A or w_j^C having a one-to-one equivalent in a transgraph.

of induction. Moreover, missing edges are hard to avoid since the input dictionaries are usually independently created, and their completeness is seldom guaranteed. However, adding missing edges into the transgraph makes it complete and, thus, makes induction more accurate. This is why $D_{l_1-l_2}$ and $D_{l_2-l_1}$ are merged when dictionaries are available for either direction [Tanaka and Umemura 1994]. We assign a weight to missing edge e, indicating the chance of it being incorrectly missed.

Although many methods are proposed for this calculation, we employ a simple statistical method [Nakov and Ng 2012] for the sake of simplicity; see Formula 5.[6] However, one can extend our method by adopting a different formula or even using external knowledge to gain more accurate weights. Notice that the weight of an existing edge (which exists when the transgraph is formed) is predefined as 1, which means that the chance of an existing edge to be missed is zero. For a possible missing edge, $e(w_i^A, w_{jh}^B)$, its weight equals the chance of a word pair $(w_i^A, w_j^C) \in \{(w_i^A, w_{i_m}^C)\}$ to be a one-to-one pair whose value is the maximum among all pairs, $\{(w_i^A, w_{i_m}^C)\}$, which are relevant to $e(w_i^A, w_j^B)$ (in other words, any pair in this set needs $e(w_i^A, w_j^B)$ to be added in order to be identified as a one-to-one pair). The same calculation is applicable for a possible missing edge $e(w_j^C, w_h^B)$:

$$\begin{aligned} Weight\left(w_i^A, w_h^B\right) &= max\left\{P\left(w_i^A, w_{i_m}^C\right)\right\} \\ Weight\left(w_j^C, w_h^B\right) &= max\left\{P\left(w_{j_m}^A, w_j^C\right)\right\}, \end{aligned} \tag{4}$$

where the likelihood of a pair (w_i^A, w_j^C) to be a one-to-one pair is calculated by the following equation:

$$\begin{aligned} P\left(w_i^A, w_j^C\right) &= P' \cdot P(w_i^A|w_j^C) \cdot P\left(w_j^C|w_i^A\right) \\ P' &= \tfrac{Min(|V_A|,|V_C|)}{Max(|V_A|,|V_C|)} \\ where\ P\left(w_i^A|w_j^C\right) &= \sum P\left(w_i^A|w_h^B\right) \cdot P\left(w_h^B|w_j^C\right) \\ P\left(w_j^C|w_i^A\right) &= \sum P\left(w_j^C|w_h^B\right) \cdot P\left(w_h^B|w_i^A\right). \end{aligned} \tag{5}$$

P' represents a maximum weight value of $w_i^A \in V^A$ or $w_j^C \in V^C$ having its one-to-one equivalent in transgraph; h is the index of pivot words shared by w_i^A and w_j^C. Figure 2

[6] Formula can yield a value that exceeds 1. To handle this case, the obtained probability values of all the candidates are normalized to the range of 0 to 1 by dividing by number of pivot words in the corresponding transgraph.

shows the edges that are considered to be missing in the sample transgraph (given in Figure 1(a)), and an example of their weights is calculated by Formula 5.

4.4. Objective Function

A transgraph is the maximum scope of extracting one-to-one pairs. In a transgraph, however, as mentioned in the previous section, a missing edge can make a one-to-one pair undetectable. For instance, in Figure 1(a), assume that w_1^{A} and w_1^{C} are a one-to-one pair, but this fails to be detected because any automatic attempt to classify it as a one-to-one pair needs to ignore the absence of the edge $e(w_1^{\text{C}}, w_3^{\text{B}})$ in that transgraph, which is actually a violation of Constraint 2.

Therefore, an edge is allowed to be added to the transgraph if it has nonzero probability, p, of having been missed. If it is added, then a certain cost, $1 - p$, is to be paid. We define the process of extracting one-to-one pairs from a transgraph as an optimization problem; the objective is to extract as many one-to-one pairs as possible while minimizing the cost of edge addition, where the cost is the penalty for adding an edge that should not exist.

We used a Boolean optimization framework, WPMax-SAT, to formulate the induction to generate the optimally correct one-to-one pair set, since the facts that whether a pair has one-to-one relation and whether an edge is actually missing, and constraints can be easily represented by Boolean variables and expressions.

In the next section, we will describe how we formalize this problem within the WPMax-SAT framework and then evaluate CNF (Conjunctive Normal Form) formulas to generate one-to-one pairs.

5. SAT-BASED FORMULATION

This section describes our proposal of formalizing the problem within the WPMax-SAT framework and process of extracting one-to-one pairs from a transgraph in detail.

5.1. Preliminaries

Boolean Satisfiability (SAT) is the problem of finding, if it exists, an assignment to the set of Boolean variables $\mathbb{V}$ that satisfies the Boolean formula expressed in CNF (Conjunctive Normal Form) [Biere et al. 2009]. A *literal* is a Boolean variable υ or its negation $\neg\upsilon$; a *clause* is a disjunction (logical OR) of literals (e.g., $\upsilon_1 \vee \neg\upsilon_2 \vee \neg\upsilon_3$). Each clause consists of ORed literals. A CNF φ is the conjunction (logical AND) of m clauses $c_1, \ldots, c_m$, where c_i is a disjunction of k_i literals. φ is *satisfied* if it evaluates to 1 (TRUE), such that all $c_i \in \varphi$ evaluate to 1.

There are several extensions to the SAT problem. One such extension of interest is Weighted partial Max-SAT (WPMax-SAT) [Fu and Malik 2006], which aims to satisfy a partial set of clauses. In a WPMax-SAT problem, clauses are assigned weights (natural number in most cases, though real numbers are also widely used) and are separated into hard and soft types. Hard clauses have maximum weights (represented by infinity ∞) and all must be satisfied, while soft clauses need to be satisfied such that the sum of the weights of the satisfied soft clauses is maximized or the sum of the weights of the unsatisfied (falsified) is minimized.

Formally, a WPMax-SAT is a multiset of weighted clauses $\varphi = \{(c_1, \omega_1), \ldots, (c_m, \omega_m), (c_{m+1}, \infty), (c_{m+m'}, \infty)\}$, where the first m clauses (φ^+) are soft and the last m' clauses (φ^∞) are hard. WCNF formula (weighted extension of CNF) φ is the problem of finding an assignment to the set of Boolean variables $\mathbb{V}$ that minimizes the cost of the assignment on φ. If the cost is infinity, it means that we must falsify a hard clause and say that the multiset is unsatisfiable.

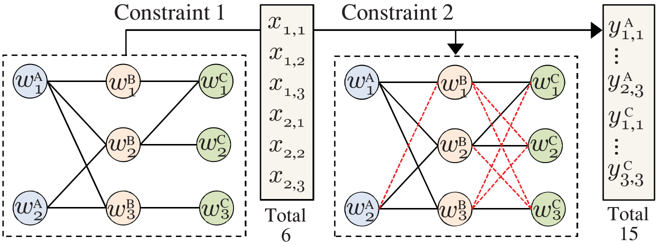


Fig. 3. The process of creating variables for a *transgraph*.

5.2. Encoding

As a first step of casting the problem in WPMax-SAT form, we apparently need a variable to denote whether a given word pair is a one-to-one pair. Moreover, another variable is also needed to represent whether an edge is missing, since the identification of a one-to-one pair requires the existence of particular edges. Overall, we use x and y to denote one-to-one pair candidates and edges in the transgraph, respectively:

—$x_{i,j}$, representing a pair (w_i^A, w_j^C), turns TRUE if it is a one-to-one pair and turns FALSE otherwise. (It is easily estimated that the number of x variables of a transgraph never exceeds $|V^A| \times |V^C|$). X denotes a set of x variables in a given problem instance.
—$y_{i,j}^A$, representing an edge $e(w_i^A, w_j^B)$, turns TRUE if it exists and turns FALSE otherwise.
—$y_{h,j}^C$, represents an edge $e(w_h^C, w_j^B)$, turns TRUE if it exists and turns FALSE otherwise.

Note that $Y_{existing}^l$ and $Y_{missing}^l$, $l \in \{A, C\}$, denote the set of existing and missing edges, respectively. Figure 3 illustrates how variables are created for a transgraph.

Before evaluating a WPMax-SAT problem by using a solver, it must be encoded to CNF. There are several ways of encoding most problems [Biere et al. 2009], yet the choice of encoding can be as important as the choice of search algorithm. However, for our problem, we use a resolution approach based on simple Boolean algebra rules such as $\upsilon_1 \rightarrow \upsilon_2 \wedge \upsilon_3 \Leftrightarrow (\neg\upsilon_1 \vee \upsilon_2) \wedge (\neg\upsilon_1 \vee \upsilon_3)$, because it is the most appropriate way to encode the constraints in our problem within the WPMax-SAT framework.

We use hard clauses to encode all the constraints that must be satisfied, and an apparent constraint: *an existing edge cannot be deleted* (this constraint is added because preexisting edges need to be protected from being deleted, since they are assumed to be created by humans). Meanwhile, the missing edges are encoded with soft clauses since adding an edge is not mandatory. In the following clause formulations, φ^∞ indicates hard, while φ^+ indicates soft.

5.2.1. Hard Clauses in Encoding to Prevent Edge Deletion.

$$\varphi_1^\infty = \Big[\bigwedge (y_{i,j}^A, \infty)\Big] \wedge \Big[\bigwedge (y_{h,j}^C, \infty)\Big], \tag{6}$$

where $y_{i,j}^A \in Y_{existing}^A$, $y_{h,j}^C \in Y_{existing}^C$.

5.2.2. Soft Clauses in Encoding for Addition of Missing Edges.

$$\varphi^+ = \Big[\bigwedge (\neg y_{i,j}^A, 1 - \omega_{i,j}^A)\Big] \wedge \Big[\bigwedge (\neg y_{h,j}^C, 1 - \omega_{h,j}^C)\Big], \tag{7}$$

where $y_{i,j}^A \in Y_{missing}^A$, $y_{h,j}^C \in Y_{missing}^C$; ω is the weight.

5.2.3. Hard Clauses Encoding Symmetry Constraint.

$$\varphi_2^\infty = \Big[\bigwedge (\neg x_{i,j} \vee y_{i,h}^A, \infty)\Big] \wedge \Big[\bigwedge (\neg x_{i,j} \vee y_{j,h}^C, \infty)\Big] \tag{8}$$

5.2.4. Hard Clauses Encoding Uniqueness Constraint.

$$\varphi_3^\infty = \left[\bigwedge_{j \neq h} (\neg x_{i,j} \vee \neg x_{i,h}, \infty) \right] \wedge \left[\bigwedge_{i \neq p} (\neg x_{i,j} \vee \neg x_{p,j}, \infty) \right] \tag{9}$$

5.3. Finding Solution

CNF formula $\varphi = \varphi^+ \wedge \varphi_1^\infty \wedge \varphi_2^\infty \wedge \varphi_3^\infty$ can be evaluated by a Max-SAT solver to output an optimal variable assignment (solution). However, any satisfiable assignment on φ ends up with minimum cost, equally zero, because no hard clause in φ requires x variables to evaluate to `TRUE` (doing so may need edge addition that eventually increases the cost of assignment). However, we resolve this by adding a new hard clause whose constraint is that at least one x variable must evaluate to `TRUE`. This clause is simply the disjunction of all the x variables.

Constraint 4: *In a transgraph, at least one one-to-one pair should be extracted*. It is encoded as follows:

$$\varphi_4^\infty = \bigvee (x_{i,j}, \infty). \tag{10}$$

Therefore, the complete CNF becomes $\varphi = \varphi^+ \wedge \varphi_1^\infty \wedge \varphi_2^\infty \wedge \varphi_3^\infty \wedge \varphi_4^\infty$; solving it returns an optimal assignment with minimum cost,[7] which equals 0 when no edge is added or exceeds 0 when the most probable missing edge(s) is added.

An optimal assignment can have a single variable $x_{m,k} \in X$ evaluated to `TRUE`, while all others, if available, are falsified. In this case, the corresponding pair $(w_m^{\text{A}}, w_k^{\text{C}})$ is considered to be the most reliably correct one-to-one pair. We add it into output dictionary $D_{\text{A-C}}$ and regenerate φ by reflecting the awareness of $\mathbb{O}(w_m^{\text{A}}, w_k^{\text{C}})$, which can be encoded by a new hard clause $(x_{m,k}, \infty)$. The regenerated φ is again evaluated by the solver to identify one more one-to-one pair. The same process is iterated until φ becomes unsatisfiable, at which point the extraction of the output dictionary completes (as in Algorithm 1).

ALGORITHM 1: Extracting one-to-one pairs from a transgraph

```
Input: $G$ – a transgraph
Output: $R$ – Set of one-to-one pairs
($\varphi$, map) ← encode $G$ to CNF /* $\varphi = \varphi^+ \wedge \varphi_1^\infty \wedge \varphi_2^\infty \wedge \varphi_3^\infty \wedge \varphi_4^\infty$ where $\varphi_4^\infty = (\bigvee x_{i,j}, \infty)$*/
$X \leftarrow \emptyset$
while $\varphi$ is satisfied do
    $\mathcal{A} \leftarrow$ take an optimal assignment on $\varphi$
    $x_{m,k} \leftarrow$ take $x_{i,j} \in \mathcal{A}$ ,where $x_{i,j} \notin X$, and $x_{i,j} = 1$
    $X \leftarrow X \cup \{x_{m,k}\}$
    $\varphi_4^\infty \leftarrow \varphi_4^\infty - x_{m,k}$ /*exclude $x_{m,k}$ from $\bigvee x_{i,j}$*/
    $\varphi \leftarrow \varphi \wedge (x_{m,k}, \infty)$ /* create a new hard clause and insert it into $\varphi$ */
end while
return $R \leftarrow map(X)$
```

We describe how two one-to-one pairs are extracted from the example transgraph in Figure 1(a) after three iterations (as illustrated in Figure 4). Before solving the problem, a corresponding $\varphi = \varphi^+ \wedge \varphi_1^\infty \wedge \varphi_2^\infty \wedge \varphi_3^\infty$ is formed, where $\varphi_4^\infty = x_{1,1} \vee x_{1,2} \vee x_{1,3} \vee x_{2,1} \vee x_{2,2} \vee x_{2,3}$.

[7] Notice that the optimal assignment may not be unique, since more than one assignment may have equally minimum cost. If it is the case, the solver selects one randomly based on its designated behavior.

Iteration	$\varphi=\varphi^{+} \wedge \varphi^{\infty}{}_{1} \wedge \varphi^{\infty}{}_{2} \wedge \varphi^{\infty}{}_{3} \wedge \varphi^{\infty}{}_{4}$		Optimal Assignment		
	$\varphi^{\infty}{}_{4}$	New clause	Detected One-to-one Pair	Cost	Added edge
1st	$x_{1,1} \vee x_{1,2} \vee x_{1,3} \vee x_{2,1} \vee x_{2,2} \vee x_{2,3}$	-	$x_{1,1}$, $\mathbb{O}(w_1^{A}, w_1^{C})$	0.750	$y^{C}_{1,3}$, $e\,(w_1^{C}, w_3^{B})$
2nd	$x_{1,2} \vee x_{1,3} \vee x_{2,1} \vee x_{2,2} \vee x_{2,3}$	$(x_{1,1}, \infty)$	$x_{2,3}$, $\mathbb{O}(w_2^{A}, w_3^{C})$	0.834	$y^{C}_{3,2}$, $e\,(w_3^{C}, w_2^{B})$
3rd	$x_{1,2} \vee x_{1,3} \vee x_{2,1} \vee x_{2,2}$	$(x_{1,1} \wedge x_{2,3}, \infty)$	Unsatisfiable		

Fig. 4. Detail of solving the transgraph given in Figure 1(a).

(1) φ is evaluated: an optimal solution is found, where $x_{1,1}$ is assigned to TRUE, since the cost, $1 - 0.250 = 0.750$ (where 0.250 is the weight of edge $e(w_1^{C}, w_3^{B})$ which is incorrectly missed as given in Figure 2), of adding the edge $e(w_1^{C}, w_3^{B})$ is the minimum possible. The fact that $x_{1,1}$ is TRUE represents a new hard constraint and forms corresponding clause $(x_{1,1}, \infty)$, which becomes a part of φ. Meanwhile, φ_4^{∞} updates to $x_{1,1} \vee x_{1,2} \vee x_{1,3} \vee x_{2,1} \vee x_{2,2} \vee x_{2,3}$ to prevent deadlock. This iteration produces the one-to-one pair $\mathbb{O}(w_1^{A}, w_1^{C})$.

(2) φ is evaluated: an optimal solution is found, where the variable $x_{2,3}$ is assigned to TRUE, since the cost, $1 - 0.166 = 0.834$ (where 0.166 is the weight of an edge $e(w_3^{C}, w_2^{B})$ which is incorrectly missed as given in Figure 2), of adding edge $e(w_3^{C}, w_2^{B})$ is the minimum possible. Likewise, $x_{2,3} =$ TRUE represents a new hard constraint and forms corresponding clause $(x_{2,3}, \infty)$, which is attached to φ. Meanwhile, φ_4^{∞} becomes $x_{1,2} \vee x_{1,3} \vee x_{2,1} \vee x_{2,2}$. This iteration produces the one-to-one pair $\mathbb{O}(w_2^{A}, w_3^{C})$.

(3) φ is evaluated: no solution is found (problem is unsatisfiable) because, in this case, any attempt to have an x variable assigned TRUE violates the Uniqueness Constraint imposed by φ_3^{∞}.

6. ALTERNATIVE FORMALIZATIONS

Cardinality constraints—expressing numerical bounds on discrete quantities—arise frequently out of the encoding of real-world problems. Due to the progress made over the last years in solving propositional satisfiability instances, interest has increased in tackling problems that include cardinality constraints using SAT solvers. This, however, requires the encoding of cardinality constraints in the language of purely propositional logic or, more specifically, in CNF. However, Boolean cardinality constraints put numerical restrictions on the number of propositional variables that are allowed to be TRUE at the same time. Expressing such constraints by pure CNF leads to more complex SAT instances [Aloul et al. 2002]. Its typical expression is that not more than k out of the n Boolean variables $x_1, \ldots, x_n$ are allowed to be TRUE, and the common way of converting such a constraint using purely propositional logic is to explicitly exclude all possible combinations of $k + 1$ variables being simultaneously TRUE, which requires $(\frac{n}{k+1})$ clauses of length $k + 1$. In the worst case of $k = [n/2] - 1$, this amounts to $O(2^n/\sqrt{n/2})$ clauses [Sinz 2005].

For instance, in the example transgraph of this article, there are three one-to-one pair candidates that include the word w_1^{A}: (w_1^{A}, w_1^{C}), (w_1^{A}, w_2^{C}), and (w_1^{A}, w_3^{C}); their corresponding variables are $x_{1,1}$, $x_{1,2}$, and $x_{1,3}$, respectively. In the problem instance, at most one of them is allowed to be TRUE due to the *Uniqueness* constraint. The propositional logic to express it is as follows: $(x_{1,1} \rightarrow \neg x_{1,2} \wedge \neg x_{1,2}) \wedge (x_{1,2} \rightarrow \neg x_{1,1} \wedge \neg x_{1,3}) \wedge (x_{1,3} \rightarrow \neg x_{1,1} \wedge \neg x_{1,2})$, which is to be transformed into six clauses in CNF: $(\neg x_{1,1} \vee \neg x_{1,2})$, $(\neg x_{1,1} \vee \neg x_{2,1})$, $(\neg x_{1,2} \vee \neg x_{1,1})$, $(\neg x_{1,2} \vee \neg x_{2,1})$, $(\neg x_{2,1} \vee \neg x_{1,1})$, and $(\neg x_{2,1} \vee \neg x_{1,2})$.

Our approach to bilingual dictionary induction involves a large amount of cardinality constraints, and unfortunately, the weakness of SAT in handling them negatively affects the computation performance of the proposal (details are given in Experiment section). Therefore, we consider that it is reasonable to discuss some alternative formalizations with the goal of improving performance.

Integer Linear Programming[8] (ILP) handles such constraints efficiently (but generic ILP solvers may ignore the Boolean nature of 0-1 variables) [Aloul et al. 2002]. For example, in the previous case, the cardinality can be tackled by the single inequality $x_{1,1} + x_{1,2} + x_{1,2} \geq 0$. Therefore, a specialized 0-1 ILP formalization can be a reasonable candidate. In this section, we briefly compare Max-SAT and ILP formalizations. A full comparison of computation performance is given in Section 7.4.

6.1. ILP-Based Formulization

An integer programming problem is a mathematical optimization or feasibility program in which some or all of the variables are restricted to be integers. In many settings, the term refers to ILP, in which the objective function and the constraints (other than the integer constraints) are linear. Integer programming is NP-hard, while its special case, 0-1 Integer Linear Programming, in which unknowns are binary, is an NP-complete problem [Wolsey 1998]. Moreover, 0-1 techniques tend to outperform generic ILP on Boolean optimization problems [Aloul et al. 2002].

SAT (as well as its Max-SAT extension) problem can be easily transformed to its ILP equivalent [Li et al. 2004]. This provides an alternative tool for solving SAT by using ILP. On the other hand, given an ILP problem, we can also transform it to an SAT problem in polynomial time by the NP-complete theory [Cook 1971].

We formalize each constraint imposed by one-to-one assumption in both Max-SAT and 0-1 ILP as a comparison study.

6.1.1. Preventing Edge Deletion.

WPMax – SAT formulization:

$$[\bigwedge (y^{\text{A}}_{i,j}, \infty)] \wedge [\bigwedge (y^{\text{C}}_{h,p}, \infty)] \quad \text{where } y^{\text{A}}_{i,j} \in Y^{\text{A}}_{existing},\ y^{\text{C}}_{h,p} \in Y^{\text{C}}_{existing}$$

ILP formulization:

$$\sum y^{\text{A}}_{i,j} + \sum y^{\text{C}}_{h,p} = |Y^{\text{A}}_{existing}| + |Y^{\text{C}}_{existing}| \quad \text{where } y^{\text{A}}_{i,j} \in Y^{\text{A}}_{existing},\ y^{\text{C}}_{h,p} \in Y^{\text{C}}_{existing}$$

6.1.2. Encoding Symmetry Constraint. Given a word pair $(w^{\text{A}}_i, w^{\text{C}}_j)$, where w^{A}_i and w^{C}_j have $V^{\text{B}}_i = \{w^{\text{B}}_{i_1}, \ldots, w^{\text{B}}_{i_n}\}$ and $V^{\text{B}}_j = \{w^{\text{B}}_{j_1}, \ldots, w^{\text{B}}_{j_m}\}$ meanings preexist in a transgraph, respectively. If $(w^{\text{A}}_i, w^{\text{C}}_j)$ is a one-to-one pair, then w^{A}_i and w^{C}_j share the same meanings $V^{\text{B}}_{i,j} = V^{\text{B}}_i \cup V^{\text{B}}_j$, where $|V^{\text{B}}_{i,j}| < n + m$. The *Symmetry Constraint* for this pair is formalized as follows:

WPMax – SAT formulization:

$$[\bigwedge (\neg x_{i,j} \vee y^{\text{A}}_{i,h}, \infty)] \wedge [\bigwedge (\neg x_{i,j} \vee y^{\text{C}}_{j,h}, \infty)]$$

ILP formulization:

$$\sum_{y^{\text{A}}_{i,j} \in Y'} y^{\text{A}}_{i,j} + \sum_{y^{\text{C}}_{i,j} \in Y'} y^{\text{C}}_{i,j} - 2|V^{\text{B}}_{i,j}| x_{i,j} \geq 0,$$

where Y' is a set of variables corresponding to the symmetric edges (both existing and missing) that connect w^{A}_i and w^{C}_j to $V^{\text{B}}_{i,j}$. With these formulas, WPMax-SAT produces $|V^{\text{B}}_{i,j}|$ number of clauses when encoded in CNF. By contrast, ILP handles the constraint with just a single inequality. For example, given the pair $(w^{\text{A}}_1, w^{\text{C}}_1)$ and

[8] For an overview and example of integer linear programming, refer to Schrijver [1998]

$V^{\mathrm{B}}_{1,1} = \{w^{\mathrm{B}}_1, w^{\mathrm{B}}_2, w^{\mathrm{B}}_3\}$, the formalization yields

WPMax – SAT formulization:
$x_{1,1} \rightarrow y^{\mathrm{A}}_{1,1} \wedge y^{\mathrm{A}}_{1,2} \wedge y^{\mathrm{A}}_{1,3} \wedge y^{\mathrm{C}}_{1,1} \wedge y^{\mathrm{C}}_{1,2} \wedge y^{\mathrm{C}}_{1,3} \Leftrightarrow$
$(\neg x_{1,1} \vee y^{\mathrm{A}}_{1,1}) \wedge (\neg x_{1,1} \vee y^{\mathrm{A}}_{1,2}) \wedge (\neg x_{1,1} \vee y^{\mathrm{A}}_{1,3}) \wedge (\neg x_{1,1} \vee y^{\mathrm{C}}_{1,1}) \wedge (\neg x_{1,1} \vee y^{\mathrm{C}}_{1,2})$
$\wedge (\neg x_{1,1} \vee y^{\mathrm{C}}_{1,3})$

ILP formulization:
$y^{\mathrm{A}}_{1,1} + y^{\mathrm{A}}_{1,2} + y^{\mathrm{A}}_{1,3} + y^{\mathrm{C}}_{1,1} + y^{\mathrm{C}}_{1,2} + y^{\mathrm{C}}_{1,3} - 6x_{1,1} \geq 0,$

where, in case of ILP, if the solver assigns 1 to $x_{1,1}$, it also has to assign 1 to all the six y variables; if the solver assigns 0 to $x_{1,1}$, then there will be no restriction on the values of y variables to make this inequality valid.

6.1.3. Encoding Uniqueness Constraint. Given a set of pairs where all items include a common word w^{A}_i or w^{C}_j, let X' denote the Boolean variable set corresponding to these pairs, while X denotes a set of Boolean variables representing one-to-one pair candidates generated from G. The Uniqueness constraint is written as follows:

WPMax – SAT formulization:
$[\bigwedge_{j \neq k} (\neg x_{i,j} \vee \neg x_{i,k}, \infty)] \wedge [\bigwedge_{i \neq k} (\neg x_{i,j} \vee \neg x_{k,j}, \infty)]$ where $x_{i,j}, x_{k,j} \in X$

ILP formulization:
$\sum x_{i,j} \geq 0$ where $x_{i,j} \in X'$.

6.1.4. Complete Formalization. Given transgraph G, let X denote a set of Boolean variables representing one-to-one pair candidates generated from G; the problem of extracting one-to-one pairs from G is formalized, within the 0-1 ILP framework, as follows:

maximize:

$$\mu_1 \cdot \sum x_{i,j} - \mu_2 \cdot \left\{ \sum (1 - \omega^{\mathrm{A}}_{i,h}) \cdot y^{\mathrm{A}}_{i,h} + \sum (1 - \omega^{\mathrm{C}}_{j,p}) \cdot y^{\mathrm{C}}_{j,p} \right\}$$

where $x_{i,j} \in X, y^{\mathrm{A}}_{i,h} \in Y^{\mathrm{A}}_{missing}, y^{\mathrm{C}}_{j,k} \in Y^{\mathrm{C}}_{missing}$

subject to:

(1) $\sum y^{\mathrm{A}}_{i,j} = 1; \sum y^{\mathrm{C}}_{i,j} = 1;$
where $y^{\mathrm{A}}_{i,j} \in Y^{\mathrm{A}}_{existing}, y^{\mathrm{C}}_{i,j} \in Y^{\mathrm{C}}_{existing}$

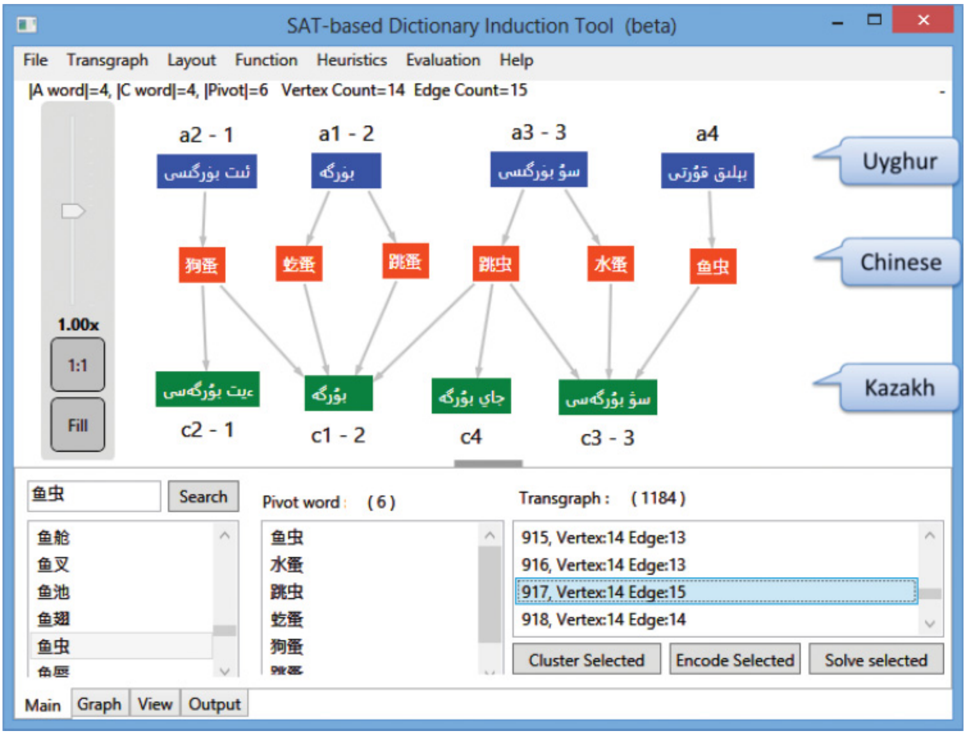
(2) $\forall x_{i,j} \quad \sum y^{\mathrm{A}}_{i,j} + \sum y^{\mathrm{C}}_{i,j} - 2|Y^{'}| \cdot x_{i,j} \geq 0$
where $Y' \subset Y^{\mathrm{A}}_{missing},\ y^{\mathrm{A}}_{i,j} \subset Y^{'},\ y^{\mathrm{C}}_{i,j} \subset Y^{'}$

(3) $\forall x_{i,j} \quad \sum x_{i,j} \geq 0$
where
$x_{i,j} \in X'$ and
X' is the of variables that representing the word pairs which share w^{A}_i or w^{C}_j.

(4) $x_{i,j}, y^{\mathrm{A}}_{i,j}, y^{\mathrm{C}}_{i,j} \in \{0, 1\}$.

Notice that (1) in the case of ILP formalization, the weight of missing edges is used in an objective function, while in Max-SAT, it is assigned to soft clauses; and (2) it is possible to balance precision against recall of the output dictionary to some extent by

[9] Library of SAT and Boolean optimization solver: http://www.sat4j.org.



|A word|=4, |C word|=4, |Pivot|=6 Vertex Count=14 Edge Count=15

鱼虫

鱼舱
鱼叉
鱼池
鱼翅
鱼虫
鱼…

鱼虫
水蚤
跳虫
虼蚤
狗蚤
跳蚤

915, Vertex:14 Edge:13
916, Vertex:14 Edge:13
917, Vertex:14 Edge:15
918, Vertex:14 Edge:14

Fig. 5. A screenshot: evaluation of a transgraph with 14 vertices, which resulted in 3 one-to-one pairs with full precision and 75% recall.

Table II. Structure of Input Dictionaries

Dictionary	*zh* Words	*ug*/*kk* Words	Pair
$D_{zh\text{-}ug}$	52,478	70,989	118,805
$D_{zh\text{-}kk}$	52,478	102,426	232,589

adjusting coefficients μ_1 and μ_2 in the expression of the objective function. However, in our experiment, in order to prevent Max-SAT and ILP formalizations from yielding different optimal solutions, we set μ_1 to a certain number so that it can be guaranteed for any $x_{i,j}$ that the value of $\mu_1 \cdot x_{i,j}$ should be greater than $\mu_2 \cdot \{\sum(1-\omega^{\mathrm{A}}_{i,h}) \cdot y^{\mathrm{A}}_{i,h} + \sum(1-\omega^{\mathrm{C}}_{j,p}) \cdot y^{\mathrm{C}}_{j,p}\}$. Otherwise, some one-to-one pairs may not be detected because assigning their corresponding variables TRUE may decrease the objective function's value due to excessive edge addition costs.

7. EXPERIMENT

We designed a tool (see a screenshot in Figure 5) to implement the proposal using Sat4j[9] as the default solver due to its flexibility in integration with third-party software. With this tool, we evaluated our approach by inducing $D_{ug\text{-}kk}$ from $D_{zh\text{-}ug}$ and $D_{zh\text{-}kk}$ (see Table II for details), where *ug* (Uyghur) and *kk* (Kazakh) are Turkic languages, while *zh* (Chinese) belongs to the Sino-Tibetan language family.

7.1. Tool Implementation

The highlights of the tool's main features are as follows:

—Provides many options for preprocessing the input dictionaries

Table III. Detail of Transgraphs

Transgraph		$\lvert V_{zh}\rvert$	$\lvert V_{ug}\rvert$	$\lvert V_{kk}\rvert$	Edge
#1 ~ #1,183	Smallest	2	2	3	6
	Largest	13	21	27	71
#1184		35,539	47,893	66,693	287,966

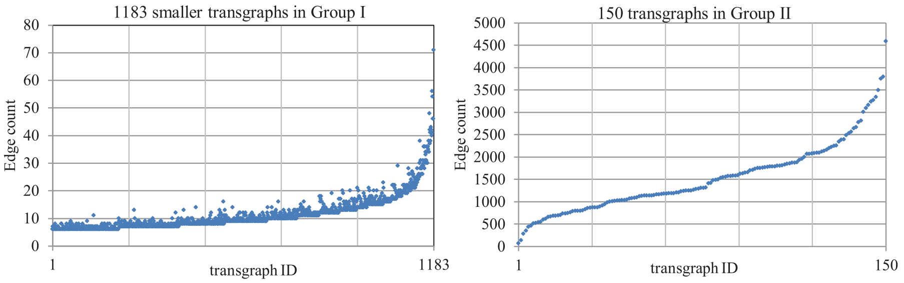


Fig. 6. Distribution of transgraphs in Group I and Group II, which are ordered by size.

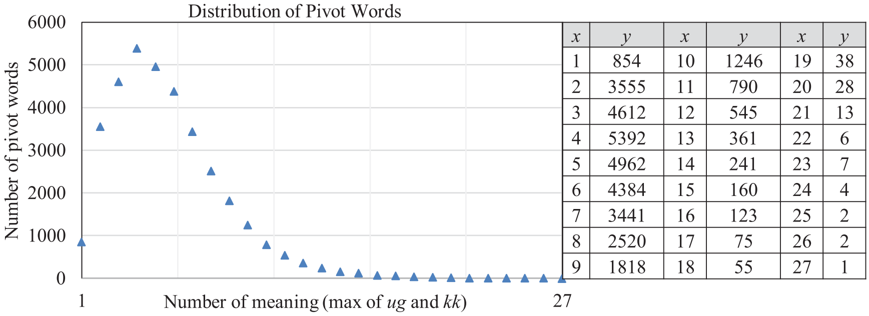
—Displays transgraphs using dynamic graph components (see Figure 5) so that users can easily observe the induction process and permits interaction with transgraphs to manually modify their structure (e.g., annotating known one-to-one pairs or adding missing edges)
—Produces comprehensive statistics of the structures of input dictionaries, transgraphs, CNF encoding, solutions, and some other details such as computation performance
—Supports two different problem-solving frameworks: Max-SAT and ILP (is expected to support more)
—Bilingual human experts can use it to evaluate automatically selected sample pairs easily

7.2. Experiment Settings

Table II shows structural information yielded by preprocessed $D_{zh\text{-}ug}$ and $D_{zh\text{-}kk}$. Connecting them resulted in 12,393 transgraphs. Among them, we selected only 1,184, each of which involves at least two pivot words (see Table III). In theory, our approach does not make sense to others (all the possible assignments always have equal cost, so a random selection is valid). However, these 1,184 transgraphs involve 52,218 *ug* and 73,093 *kk* words, which make up 73.5% and 71.4% of total *ug* an *kk* words in the input dictionaries, respectively. This means that our approach affects the majority part of input data. For others, approaches that consider the orthographic similarity may work [Haghighi et al. 2008], which is left as a future work.

One of them, #1,184, is remarkably large; it contains 69% of all vertices and 82% of all edges, while the rest are distributed to 1,183 smaller transgraphs (see Figure 6 for details). We examined why there was such a large difference in the percentage of input dictionaries. Figure 7 shows the distribution of a number of meanings of 35,235 pivot words contained in the transgraph #1,184, from which we can draw a rough conclusion that the large number of polysemic words in the pivot language explains the difference. However, different language pairs will need to be examined to see whether this imbalance in transgraphs is common.

Encoding this large transgraph resulted in a CNF formula with 16,879,348 variables and 46,059,686 clauses. We were unable to evaluate it using the Sat4j solver in our



x	*y*	*x*	*y*	*x*	*y*
1	854	10	1246	19	38
2	3555	11	790	20	28
3	4612	12	545	21	13
4	5392	13	361	22	6
5	4962	14	241	23	7
6	4384	15	160	24	4
7	3441	16	123	25	2
8	2520	17	75	26	2
9	1818	18	55	27	1

Fig. 7. Quantity distribution of pivot words over the number of meanings, which implies the availability of a relatively large number of ambiguous words in the pivot language.

Table IV. Overview of the Induction Result

Transgraphs	Maximum Expected	Actual Extracted	Precision	Recall
Group I	3,877	3,708 (95.6%)	79%	75.5%
Group II	47,893	39,907 (83%)	84%	70%
Overall	51,770	43,615 (84.2%)	83.7%	70.5%

experimental hardware environment[10] due to its high computation complexity. Hence, for experimental purposes, we partitioned transgraph #1,184 into 150 smaller subgraphs (see Figure 6 for their distribution) using a graph partitioning algorithm [Dhillon et al. 2005]. Since the goal of graph partitioning is to minimize the number of edges that cross from one subgroup of vertices to another, we consider adopting such an algorithm reasonable. However, as one can implement our proposal with more efficient SAT solvers and rerun the experiment using stronger hardware, the step of partitioning may not be needed. Also, since graph partitioning is not a focus in our work, we have not made any comparison study on relevant algorithms with our data. Instead, one that offered easy integration with our tool was preferred.

We independently processed these two groups of transgraphs (1,183 in Group I and 150 in Group II) and evaluated the induction result of each group. The overall values were also calculated by averaging. To measure the recall, we set an upper-bound value that represents the maximum number of possible one-to-one pairs available in a transgraph. It is given by $Min(|V^{\mathrm{A}}|, |V^{\mathrm{C}}|)$ for a transgraph with $|V^{\mathrm{A}}|$ words in language A and $|V^{\mathrm{C}}|$ in language C. Moreover, if there are n transgraphs, $g_1, \ldots, g_n$, the overall upper-bound value should equal $\sum_{i=1}^{n} Min(|V^{\mathrm{A}}_{g_i}|, |V^{\mathrm{C}}_{g_i}|)$. The overall recall is obtained by dividing this value by the maximum number of possible one-to-one pairs. To evaluate precision, samples were evaluated by bilingual human experts.

7.3. Results and Analysis

Figure 8 illustrates the distribution of maximum expected and actual extracted one-to-one pairs from transgraphs in each group; we can observe extraction with relatively high coverage in almost every transgraph. Overall, however, 84.2% of maximum expected one-to-one pairs were extracted as the details show in Table IV.

In order to evaluate the precision of the extracted one-to-one pairs, we randomly selected 3×100 samples from the sets of one-to-one pairs extracted from each group, receptively, and asked a *ug-kk* bilingual human to judge whether they are indeed

[10] Hardware – CPU: Intel(R) Core(TM) i5 2.40GHz; 8GB RAM Software – Dictionary induction tool with Sat4j 2.3 & Java 1.7 & .Net 4.0.

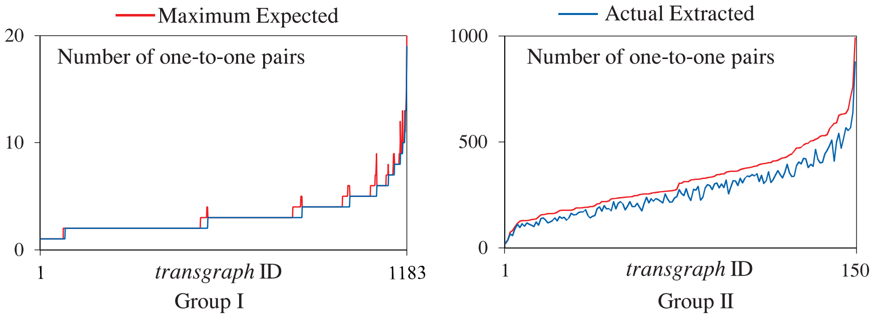


Fig. 8. Distribution of number of maximum expected and actual extracted one-to-one pairs among ordered transgraphs in Group I & II.

correctly mapped as one to one. As a result, 237 (79%) out of 300 for Group I and 251 (84%) out of 300 for Group II were determined to be correct. Thus, our method roughly yielded 70.5% overall recall as shown in Table IV. Notice that we did not ask a *ug*-*kk* bilingual human to judge whether the sample pairs are unique (in other words, to check whether a word in a one-to-one pair has an alternative translation equivalent that is also in one-to-one relation), because it is possible in practical translations. However, in our work, uniqueness represents a feature of output dictionary that it consists of an equal number of distinct *ug* and *kk* words that are in word-to-word relation.

Nonetheless, it is not reasonable to directly compare these numbers with those in related works and reach a conclusion on the efficiency of our approach, since the experimental language pairs and resources chosen in each similar research work are not quite the same. In response, we processed our 1,183+150 transgraphs with the IC method as it is used as a baseline in related works [Shezaf and Rappoport 2010]. It is a well-known approach to creating a new dictionary from just two input dictionaries with no extra information.

IC examines the two pivot word sets: the set of pivot translations of word w_i^{A} and the set of pivot translations of each w_j^{C} word that is a candidate for being a translation to w_i^{A}. The more closely they match, the better the candidate is. Since the IC has no one-to-one constraint on translation pairs, it allows multiple translations for a word through induction setting. However, in our implementation of IC, we only leave a top-ranked translation candidate in language C for each word in language A to make the two methods consistent. This does not harm the performance of IC as long as the top-ranked candidate is selected. As a result, output of the IC method was roughly 10.5% lower than the result of our proposal with similar recall, 72%.

Recall that in our proposal we did not allow one-to-many translations in the output dictionary in accordance with the one-to-one assumption, because relaxing the one-to-one constraint resulted in relatively higher recall but the loss in precision was remarkable. We consider such an output dictionary less useful than a more accurate dictionary with lower coverage. However, we acknowledge that being high in recall has the potential to overcome the precision problem if further processing is made, such as using parallel or comparable corpora to eliminate wrong translations [Nerima and Wehrli 2008; Otero and Campos 2010] and considering spelling similarity [Schulz et al. 2004]. Moreover, imposing the one-to-one restriction well controlled the runtime addition of possibly missing edges in the transgraphs, while relaxing it gives the solver to add more edges, which indeed further escalates the reduction in precision. However, we consider that allowing one-to-many translation by disabling the uniqueness

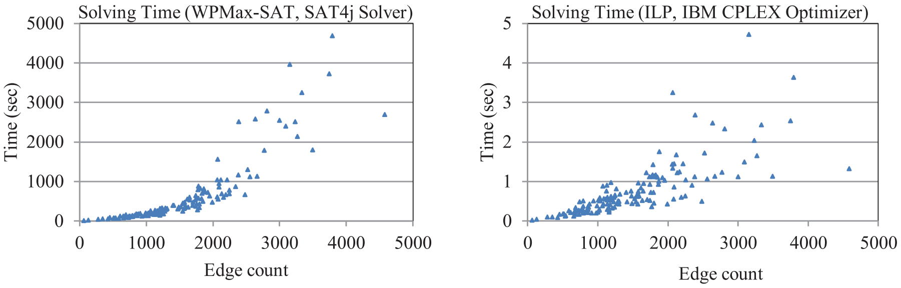


Fig. 9. Distribution of solving time of 150 transgraphs with Max-SAT and ILP formalizations.

constraint is a word try for controlling the balance between recall and precision of an automatically created dictionary.

7.4. Computation Performance

The computation performance might be another concern when implementing our method or using our tool to create one's own bilingual dictionaries. Therefore, in order to evaluate the speed of processing a transgraph and the memory space required to store CNF expressions, we recorded relevant data during the experiment.

The computation environment used was selected to suit ordinary users. Two experiments were conducted to compare Max-SAT and ILP. For ILP, the CPLEX[11] is utilized as the solver (it is widely used in the public domain due to its stability and high computation efficiency).

Figure 9 illustrates the distribution of completion time (time spent on evaluating transgraph by solver) among 150 transgraphs in Group II. With Max-SAT formalization, roughly 6 hours[12] are spent to finish solving all the transgraphs (running with four threads and 100% CPU utilization), while with ILP, only 116 seconds were spent to produce the same output dictionary, a remarkably speedy enhancement. As for the memory space used, which largely depends on the number of clauses in CNF in the case of Max-SAT and linear constraints in the case of ILP, 523MB CNF encodes were produced, while about 400MB were used to store linear constraints. More specifically, 10,354,815 clauses and 3,488,206 linear constraints were generated to encode 150 transgraphs.

The main reason for this big difference in completion time is the iterative mechanism of our SAT-based algorithm: in order to extract n number of one-to-one pairs from a *transgraph*, the corresponding problem instance needs to be evaluated n times by the SAT solver, while a single evaluation is enough to produce the same result in the case of ILP. Moreover, although the experiments were done in the same hardware environment, there might be many other factors, from underlining algorithms to programming implementation in software, that contribute to this big difference in completion time. For example, the IBM Cplex ILP solver, written in C language, is a commercial software that is designed to select a best-match algorithm based on the problem's features and size dynamically during the solution; it offers strong parallelism and can well utilize whatever number of CPU cores are available. It is hard for us to identify the effects of such factors in our experiment, and there is a need to do a more detailed comparison

[11] IBM ILOG CPLEX, http://www.ilog.com/products/cplex.

[12] Notice that this value does not include (1) the time spent on creating the transgraphs because it is negligible compare to the time spent by the SAT solver (it is a fast process that batch reads the dictionary database into memory and creates graph objects), (2) the time spent on feeding the solver with CNF encoding, and (3) the time for graph partitioning.

using different types of solvers to make a precise conclusion. As new open-source SAT solvers are emerging[13] each year with improved performance, it is possible that the performance of the SAT implementation of our proposal could be largely improved.

Notice that theoretic calculation of clauses used to encoding a transgraph is hard to formulate, since its value largely depends on graph structure. However, a maximum possible number of clauses can be obtained by Formula 7.4 for any given transgraph assuming that it constrains $|V^{\mathrm{B}}| \times (|V^{\mathrm{A}}| + |V^{\mathrm{C}}|)$ edges.

$$\mathbb{N} = \overbrace{2(a+c)}^{\text{for } \varphi_1^\infty \& \varphi^+} + \overbrace{2abc}^{\text{for } \varphi_2^\infty} + \overbrace{\frac{ac(a+c)}{2} - 1}^{\text{for } \varphi_3^\infty} + \overbrace{ab}^{\text{for Iterations}},$$

where $|V_{\mathrm{A}}| = a$, $|V_{\mathrm{B}}| = b$, and $|V_{\mathrm{C}}| = c$, respectively.

8. CONCLUSION

Bilingual dictionaries have yet to be created for many languages. Such work is challenging because many language pairs lack useful language resources like a parallel corpus and even comparable corpora. To provide an efficient, robust, and accurate dictionary creation method for poorly resourced language pairs, we presented a constraint approach to pivot-based dictionary induction, where a new dictionary of closely related language pairs is induced from two existing dictionaries using a distant language as a pivot. In our approach, the lexical intransitivity divergence is tackled by modeling an instance of induction as an optimization problem, where the new dictionary is produced as the solution of the problem. We also considered data incompleteness to some extent. An experiment showed the feasibility of our approach. However, we note the following points: (1) The problem may also be tackled by maximum weighted bipartite matching [Cheng et al. 1996] as well as other optimization frameworks other than Max-SAT and Integer Linear Programming. This is left as a future work, as we will continue to explore more efficient modeling approaches and algorithms for dictionary induction. (2) There is the potential for including spelling as additional information. (3) More comparisons are expected to find whether the method can indeed rely purely on the structure and still outperform the methods that utilize cheap external resources such as monolingual data. (4) The one-to-one assumption may be too strong for the general case, but we consider it reasonable for the case of intrafamily languages as it greatly reduces the complexity of the problem. (5) Applying the proposal to extrafamily language pairs is also promising and should be explored.

REFERENCES

Kisuh Ahn and Matthew Frampton. 2006. Automatic generation of translation dictionaries using intermediary languages. In *Proceedings of the International Workshop on Cross-Language Knowledge Induction*. Association for Computational Linguistics, 41–44.

Fadi A. Aloul, Arathi Ramani, Igor L. Markov, and Karem A. Sakallah. 2002. Generic ILP versus specialized 0-1 ILP: An update. In *Proceedings of the 2002 IEEE/ACM International Conference on Computer-Aided Design*. ACM, 450–457.

Hitham Abo Bakr, Khaled Shaalan, and Ibrahim Ziedan. 2008. A hybrid approach for converting written Egyptian colloquial dialect into diacritized Arabic. In *Proceedings of the he 6th International Conference on Informatics and Systems (INFOS'08). Cairo University*.

Shane Bergsma and Benjamin Van Durme. 2011. Learning bilingual lexicons using the visual similarity of labeled web images. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'*11), Vol. 22. 1764.

Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh (Eds.). 2009. *Handbook of Satisfiability*. Frontiers in Artificial Intelligence and Applications, Vol. 185. IOS Press.

[13] http://www.satcompetition.org/.

Francis Bond and Kentaro Ogura. 2008. Combining linguistic resources to create a machine-tractable Japanese-Malay dictionary. *Language Resources and Evaluation* 42, 2 (2008), 127–136.

Francis Bond, Takefumi Yamazaki, Ruhaida Binti Sulong, and Kentaro Okura. 2001. Design and construction of a machine–tractable Japanese-Malay lexicon. In *Annual Meeting of the Association for Natural Language Processing,* Vol. 7. 1.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics* 16, 2 (1990), 79–85.

Y. Cheng, Victor Wu, Robert Collins, A. Hanson, and E. Riseman. 1996. Maximum-weight bipartite matching technique and its application in image feature matching. In *SPIE Conference on Visual Communication and Image Processing*. 1358–1379.

Stephen A. Cook. 1971. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*. ACM, 151–158.

Inderjit Dhillon, Yuqiang Guan, and Brian Kulis. 2005. A fast kernel-based multilevel algorithm for graph clustering. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. ACM, 629–634.

Qing Dou and Kevin Knight. 2012. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 266–275.

Zhaohui Fu and Sharad Malik. 2006. On solving the partial MAX-SAT problem. In *Theory and Applications of Satisfiability Testing (SAT'06)*. Springer, 252–265.

Pascale Fung. 1995. Compiling bilingual lexicon entries from a non-parallel English-Chinese corpus. *Proceedings of the 3rd Annual Workshop on Very Large Corpora*.

Pascale Fung. 1998. A statistical view on bilingual lexicon extraction: From parallel corpora to non-parallel corpora. In *Machine Translation and the Information Soup*. Springer, 1–17.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. *Proceedings of ACL-08: HLT* (2008), 771–779.

Jirka Hana, Anna Feldman, Chris Brew, and Luiz Amaral. 2006. Tagging Portuguese with a Spanish tagger using cognates. In *Proceedings of the International Workshop on Cross-Language Knowledge Induction*. Association for Computational Linguistics, 33–40.

Ahlem Ben Hassine, Shigeo Matsubara, and Toru Ishida. 2006. A constraint-based approach to horizontal web service composition. In *The Semantic Web-ISWC 2006*. Springer, 130–143.

John Hopcroft and Robert Tarjan. 1973. Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM* 16, 6 (1973), 372–378.

Toru Ishida. 2011. *The Language Grid*. Springer.

Azniah Ismail and Suresh Manandhar. 2010. Bilingual lexicon extraction from comparable corpora using in-domain terms. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 481–489.

Varga István and Yokoyama Shoichi. 2009. Bilingual dictionary generation for low-resourced language pairs. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*. Association for Computational Linguistics, 862–870.

Hiroyuki Kaji and Toshiko Aizono. 1996. Extracting word correspondences from bilingual corpora based on word co-occurrences information. In *Proceedings of the 16th Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 23–28.

Hiroyuki Kaji, Shin'ichi Tamamura, and Dashtseren Erdenebat. 2008. Automatic construction of a Japanese-Chinese dictionary via English. In *LREC*, Vol. 2008. 699–706.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition-Volume 9*. Association for Computational Linguistics, 9–16.

Ruiming Li, Dian Zhou, and Donglei Du. 2004. Satisfiability and integer programming as complementary tools. In *Proceedings of the 2004 Asia and South Pacific Design Automation Conference*. IEEE Press, 879–882.

Wushouer Mairidan, Lin Donghui, and Toru Ishida. 2013. A heuristic framework for pivot-based bilingual dictionary induction. In *Proceedings of 3rd International Conference on Culture and Computing*.

Jun Matsuno and Toru Ishida. 2011. Constraint optimization approach to context based word selection. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence-Volume Volume 3*. AAAI Press, 1846–1851.

I. Dan Melamed. 1997. A word-to-word model of translational equivalence. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 490–497.

I. Dan Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics* 26, 2 (2000), 221–249.

Preslav Nakov and Hwee Tou Ng. 2012. Improving statistical machine translation for a resource-poor language using related resource-rich languages. *Journal of Artificial Intelligence Research* 44, 1 (2012), 179–222.

Luka Nerima and Eric Wehrli. 2008. Generating bilingual dictionaries by transitivity. In *LREC*.

Jian-Yun Nie, Michel Simard, Pierre Isabelle, and Richard Durand. 1999. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 74–81.

Malte Nuhn, Arne Mauser, and Hermann Ney. 2012. Deciphering foreign language by combining language models and context vectors. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 156–164.

Pablo Gamallo Otero and José Ramom Pichel Campos. 2010. Automatic generation of bilingual dictionaries using intermediary languages and comparable corpora. In *Computational Linguistics and Intelligent Text Processing*. Springer, 473–483.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 519–526.

Sujith Ravi and Kevin Knight. 2008. Attacking decipherment problems optimally with low-order n-gram models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 812–819.

Magnus Sahlgren and Jussi Karlgren. 2005. Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Natural Language Engineering* 11, 03 (2005), 327–341.

Wael Salloum and Nizar Habash. 2011. Dialectal to standard Arabic paraphrasing to improve Arabic-English statistical machine translation. In *Proceedings of the 1st Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*. Association for Computational Linguistics, 10–21.

Xabier Saralegi, Iker Manterola, and Iñaki San Vicente. 2012. Building a Basque-Chinese dictionary by using english as pivot. In *LREC*. 1443–1447.

Xabier Saralegi, Iker Manterola, and Iñaki San Vicente. 2011. Analyzing methods for improving precision of pivot based bilingual dictionaries. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 846–856.

Hassan Sawaf. 2010. Arabic dialect handling in hybrid machine translation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA'10)*.

Charles Schafer and David Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In *Proceedings of the 6th Conference on Natural Language Learning-Volume 20*. Association for Computational Linguistics, 1–7.

Alexander Schrijver. 1998. *Theory of Linear and Integer {rogramming*. John Wiley & Sons.

Stefan Schulz, Kornél Markó, Eduardo Sbrissia, Percy Nohama, and Udo Hahn. 2004. Cognate mapping: A heuristic strategy for the semi-supervised acquisition of a Spanish lexicon from a Portuguese seed lexicon. In *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics, 813.

Daphna Shezaf and Ari Rappoport. 2010. Bilingual lexicon generation using non-aligned signatures. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 98–107.

Carsten Sinz. 2005. Towards an optimal CNF encoding of boolean cardinality constraints. In *Principles and Practice of Constraint Programming (CP'05)*. Springer, 827–831.

Jonas Sjobergh. 2005. Creating a free digital Japanese-Swedish lexicon. In *Proceedings of PACLING*. Citeseer, 296–300.

Hana Skoumalova. 2001. Bridge dictionaries as bridges between languages. *International Journal of Corpus Linguistics, 6, Special Issue* 95, 105 (2001), 11.

Mausam, Stephen Soderland, Oren Etzioni, Daniel S. Weld, Michael Skinner, and Jeff Bilmes. 2009. Compiling a massive, multilingual dictionary via probabilistic inference. In *Proceedings of the Joint Conference* *of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*. Association for Computational Linguistics.

Kumiko Tanaka and Hideya Iwasaki. 1996. Extraction of lexical translations from non-aligned corpora. In *Proceedings of the 16th Conference on Computational Linguistics-Volume 2*. Association for Computational Linguistics, 580–585.

Kumiko Tanaka and Kyoji Umemura. 1994. Construction of a bilingual dictionary intermediated by a third language. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1 (COLING'94)*. Association for Computational Linguistics, Stroudsburg, PA, 297–303. DOI:http://dx.doi.org/10.3115/991886.991937

Rie Tanaka, Yohei Murakami, and Toru Ishida. 2009. Context-based approach for pivot translation services. In *IJCAI*. 1555–1561.

Jerzy Tomaszczyk. 1986. The bilingual dictionary under review. In *Zurilex'86 Proceedings: Papers Read at the Euralex International Congress, University of Zurich*. 289–297.

Laurence A. Wolsey. 1998. *Integer Programming*. Vol. 42. Wiley, New York.

Dekai Wu and Xuanyin Xia. 1994. Learning an English-Chinese lexicon from a parallel corpus. In *Proceedings of the 1st Conference of the Association for Machine Translation in the Americas*. Citeseer, 206–213.

Kun Yu and Junichi Tsujii. 2009. Bilingual dictionary extraction from wikipedia. *Proceedings of Machine Translation Summit XII* (2009), 379–386.

Xiaoheng Zhang. 1998. Dialect MT: A case study between Cantonese and Mandarin. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*. Association for Computational Linguistics, 1460–1464.

Received August 2013; revised October 2014; accepted January 2015