Bilingual Lexicon Induction Framework for Closely Related Languages

Arbi Haza Nasution

Doctoral Thesis Series of
Ishida & Matsubara Laboratory
Department of Social Informatics
Kyoto University

Abstract

The objective of this thesis is to support the construction of a comprehensive set of bilingual dictionaries among closely related low-resource languages. To this end, it generalizes the existing constraint-based bilingual lexicon induction method to obtain many-to-many translation pairs, and a novel method to optimize a plan of comprehensive bilingual dictionaries creation that utilizes the constraint-based technique and a manual creation in order to reduce total creation time. This thesis proposes four contributions as follows:

1. A method to generate language similarity clusters for selecting closely related target languages.

   This method utilizes Automated Similarity Judgment Program database to generate a language similarity matrix, then the matrix is further used to create the clusters by using hierarchical clustering and k-means clustering. We further extract closely related clusters which consist of languages where the similarity between them is more than 50%. We applied this method to Indonesian ethnic languages to select target languages that are enriched in the following chapters as an example.

2. A method to induce translation pairs between the target languages via a pivot language.

   The existing constraint-based bilingual lexicon induction that induces bilingual dictionary A-C from two input bilingual dictionaries A-B and B-C via pivot language B has a low recall because it only identifies cognates (words with a common etymological origin) as translation pairs. To address this issue, we generalize the method by not only identifying cognates but also further identifying their synonyms based on a ratio of shared meaning with the cognates. Our generalized constraint method (64% average f-score) outperformed the existing constraint method (41% average f-score), which shows that our method is more practical for the low-resource languages.

3. A method to optimize a plan of comprehensive bilingual dictionaries creation from a set of target languages.

   In the previous chapter, we show that our generalized constraint method is practical for low-resource languages. However, in such languages, we do not have enough bilingual dictionaries as its input. Therefore, a manual creation of some bilingual dictionaries is required. Moreover, there is an uncertainty of precision of the generalized constraint method before its execution. To model the planning under uncertainty, we use Markov Decision Process (MDP) in formalizing a plan optimizer. A state consists of each bilingual dictionary and its current status (satisfies or dissatisfies a minimum dictionary size threshold). An action set consists of two bilingual dictionary creation methods: the generalized constraint method and the manual creation. A state transition probability refers the likelihood that an action

executed from a state satisfies the minimum dictionary size threshold. Lastly, a cost of action represents a completion time of the bilingual dictionary creation method. The MDP outputs an optimal policy for each state which is a mapping between a state and an optimal action to take. To estimate a total cost, starting from a start state, the cost of optimal action at the current state is added to the total cost. Further, the next state is chosen based on the highest state transition probability until the final state is reached. The cost estimation was validated in an experiment described in the following chapter.

4. A collaborative framework to conduct an experiment for validating the above proposals in this thesis.
   Bilingual native speakers of Indonesian and ethnic language are available even though they are often scattered throughout Indonesia. Therefore, this framework provides a tool to help them collaborate. We conducted an experiment on the target languages with a threshold of 2,000 translation pairs. As a result, the actual total cost was 97% close to the estimated total cost, which shows that our plan optimizer is reliable.

These contributions are beneficial for any party who want to enrich the low-resource languages to prevent them from language endangerment.

Acknowledgements

This thesis and my work during Ph.D. study would not be possible without Allah's permission and the help of many people. Firstly and foremost, words are not enough to express my gratitude to my supervisor Professor Toru Ishida who has given me the opportunity to forge myself as a Ph.D. student in Ishida & Matsubara Laboratory, Kyoto University. Thank you very much for giving so much persistent guidance that is very valuable for my future career as a lecturer. You also have taught me the skills and attitude that shape me to become a good researcher in the future.

I also would like to express my gratitude to my adviser Professor Masatoshi Yoshikawa. I always received insightful comments from different perspective that enriched my work. I also thank my adviser Professor Tatsuya Kawahara who always gave fundamental advice that significantly improve the quality of my research.

I also sincerely express my gratitude to Associate Professor Yohei Murakami who gave me so much time, encouragement and thoughtful advice for my work. I really appreciate his so much effort in guiding my research to the tiny details.

My PhD journey will not be so amazing without a companion of my beloved wife, Winda Monika, and my son, Akira Zafirarda Nasution, for their patient throughout the years and, at the same time, giving me more motivation to finish my Ph.D. study. Her continuous feedback helped me shaping my research to be more readable and logical to general audiences.

I sincerely thank the faculty at Ishida & Matsubara Laboratory for the wonderful environment and support to learn how to do high quality research, Associate Professor Shigeo Matsubara for his kind support, Associate Professor Donghui Lin for his valuable comments during lab seminars, Associate Professor Takao Nakaguchi for teaching me about Language Grid, and Masayuki Otani for introducing me to Internet of Things. I also would not forget to thank the coordinators of the laboratory: Terumi Kosugi and Hiroko Yamaguchi who gave so much help for both study-related matters and my personal matters as a foreigner living in Japan.

I sincerely express my gratitude to my fellow laboratory members and past members for being good friends and supporting each other. Special thanks to Kemas Muslim Lhaksmana who gave much help for both my academic and personal matters, especially on how to live as a muslim in Japan. Special thanks to Mairidan Wushouer for giving so much advice for my work. Special thanks to Shinsuke Goto, Nguyen Cao Hong Ngoc, and Mondheera Pituxcoosuvarn who accompanied the journey throughout Ph.D. study. Special thanks to Dai Jiapeng and Chou Hui Chen for helping me when I was staying at Tsukuba. Special thanks to Ryosuke Okuno for preparing halal food during lab party and translating Japanese documents. Sincere thanks to Amit Pariyar, Trang Mai Xuan, Xin Zhou, Andrew W. Vargo, Hiroaki Kingetsu, Xun Cao, Junta Koyama, Akihiko Itoh, Victoria Abou Khalil,

and many others. I am grateful to know you all.

My Ph.D. study and life in Kyoto were supported by the scholarship from Indonesia Endownment Fund for Education (LPDP). The research throughout my Ph.D. study was supported by the Grant-in-Aid for Scientific Research (A) (17H00759, 2017-2020) and a Grant-in-Aid for Young Scientists (A) (17H04706, 2017-2020) from Japan Society for the Promotion of Science (JSPS)

I sincerely thank Islamic University of Riau, Indonesia for their financial and moral support. I also thank the former rector, Professor Detri Karya, the current rector, Professor Syafrinaldi, the dean of Faculty of Engineering, Associate Professor Abdul Kudus Zaini, the former head of Department of Informatics Engineering, Akmar Efendi, and the current head of Department of Informatics Engineering, Ause Labellapansa, for giving me the chance to do my Ph.D. study. Special thanks to all my colleague in Department of Informatics Engineering for the moral support throughout my Ph.D. study.

Finally, I would not achieve anything without the support of my family and relatives, especially my parents, Habib Nasution and Zainab who always support me through prayers and many ways. I also express my gratitude to my brothers, Rehazain Agustian Nasution and Salhazan Nasution, and my sister, Hafiza Oktasia Nasution, my sisters-in-law, Moni Tanti Yolanda and Riona Ulfah, my nephew, Adiva Morein Nasution, my father-in-law, Misbah Kusman, my mother-in-law, Murniati, and my brother-in-law, Ahmad Rian Wibowo for their moral support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Historical linguistics is the scientific study of language change over time in term of sound, analogical, lexical, morphological, syntactic, and semantic information [Campbell, 2013]. Comparative linguistics is a branch of historical linguistics that is concerned with language comparison to determine historical relatedness and to construct language families [Lehmann, 2013]. Many methods, techniques, and procedures have been utilized in investigating the potential distant genetic relationship of languages, including lexical comparison, sound correspondences, grammatical evidence, borrowing, semantic constraints, chance similarities, sound-meaning isomorphism, etc [Campbell and Poser, 2008]. The genetic relationship of languages is used to classify languages into language families. Closely-related languages are those that came from the same origin or proto-language, and

belong to the same language family. Several machine translation studies focused on closely-related languages [Scannell, 2006, Nakov and Tiedemann, 2012, Tiedemann, 2009].

Glottochronology, one of lexical comparison method as formulated by [Swadesh, 1955], is a method for estimating the amount of time elapsed since related languages diverged from a common ancestral language. Glottochronology depends on basic, relatively culture-free vocabulary, which is known as Swadesh list. Automated Similarity Judgment Program (ASJP) was proposed by [Holman et al., 2011] with the main goal of developing a database of Swadesh lists [Swadesh, 1955] for all of the world's languages from which lexical similarity or lexical distance matrix between languages can be obtained by comparing the word lists. For example, Indonesia has 707 low-resource ethnic languages [Lewis et al., 2015] which are mostly belong to the same language family, i.e., Austronesian language family. The language similarity matrix can be generated by utilizing ASJP. It can be used as a reference in determining target languages.

Machine readable bilingual lexicons are very useful for natural language processing applications/researches such as cross-language information retrieval [Hull and Grefenstette, 1996] and machine translation [Brown et al., 1990], but are usually unavailable for low-resource languages. These lexicons are traditionally extracted from parallel corpora, a corpus that contains source texts and their translations. However, despite good results in the extraction of bilingual lexicons, parallel corpora remains scarce resources for low-resource languages. Thus, research in bilingual lexicon extraction is shifted to comparable corpora [Rapp, 1999, Fung, 2000, Haghighi et al., 2008] which consists of texts sharing common features such as domain,

genre, register, or sampling period without having a source text-target text relationship. The approach depends on the assumption that the term and its translation appear in similar contexts [Rapp, 1999, Fung, 2000], which means that a translation equivalent of a source word can be found by identifying a target word with the most similar context vector in a comparable corpus.

Nevertheless, bilingual lexicon extraction is still highly problematic for most low-resource languages due to the paucity or outright omission of parallel and comparable corpora. The approaches of pivot language and cognate recognition have been proven useful in inducing bilingual lexicons for low-resource languages with bilingual dictionary as a sole required language resource. Measuring the semantic distance between two words from the word translation topology based on structure of input bilingual dictionaries is the main research challenge in this area. The first work on bilingual lexicon induction to create bilingual dictionaries (language A and language C) via pivot language B is Inverse Consultation (IC) [Tanaka and Umemura, 1994]. It utilizes the structure of input dictionaries to measure the closeness of word meanings and then uses the results to prune erroneous translation pair candidates. The IC approach identifies equivalent candidates of language A words in language C by consulting dictionary A-B and dictionary B-C. These equivalent candidates will be looked up and compared in the inverse dictionary C-A. The pivot-based approach is very suitable for low-resource languages, especially when dictionaries are the only language resource required. Unfortunately, for some low-resource languages, it is often difficult to find machine-readable inverse dictionaries and corpora to identify and eliminate the erroneous translation pair candidates. To overcome

this limitation, our team proposed to treat pivot-based bilingual lexicon induction as an optimization problem [Wushouer et al., 2015]. The assumption was that lexicons of closely-related languages offer instances of one-to-one mapping and share a significant number of cognates (words with similar spelling/form and meaning originating from the same root language). However, the assumption of one-to-one mapping is too strong to induce the many translation pairs needed to offset resource paucity because few such pairs can be found. Therefore, a generalization of the constraint-based bilingual lexicon induction to attain more voluminous bilingual dictionary results with many-to-many translation pairs extracted is necessary.

1.2 Objectives

The objective of this thesis is to provide a bilingual lexicon induction framework for closely related low-resource languages consisting of (1) a method to generate lexicostatistic and language similarity cluster which can be referred when selecting target languages, (2) a good quality constraint-based bilingual lexicon induction with many-to-many translation results and ultimately, a comprehensive implementation to create bilingual dictionaries from all combination of the target languages with (3) a plan optimizer to minimize the cost considering the inclusion of manual investment method by native speaker and (4) a collaborative tools to bridge the spatial gap. There are three motivations to achieve these goals:

1. Helping researchers, linguists, journalists, politicians, or government to obtain or generate lexicostatistic and language similarity cluster by

themselves. Indonesia has 707 low-resource ethnic languages where more than half of them are categorized as endangered languages. In order to save those languages, government need to take action in the prevention or revitalization of the languages. Lexicostatistic and language similarity cluster of Indonesian ethnic languages will be very useful in deciding where to start.

2. Improving the recall while maintaining a good precision of constraint-based bilingual lexicon induction by relaxing the one-to-one mapping assumption to obtain many-to-many translation pairs. A method with high precision but low recall is not suitable to enrich low-resource languages. We are not only focus on finding cognate, but also the cognate synonym since some linguistic studies show that the meaning of a word can be deduced via cognate synonym.

3. Persuading the stakeholder to invest in a comprehensive creation of bilingual dictionaries from all combination of target languages. Obviously, an effort to save low-resource languages from language endangerment requires a big amount of money. Even though some people may be skeptical on the importance of language endangerment prevention, but we believe that at least the government will not turn their back on this crucial problem. However, we need to persuade the government to invest by showing the estimated optimal plan with the least cost and show how do we address the current situation where it is difficult to find bilingual native speakers of two ethnic languages.

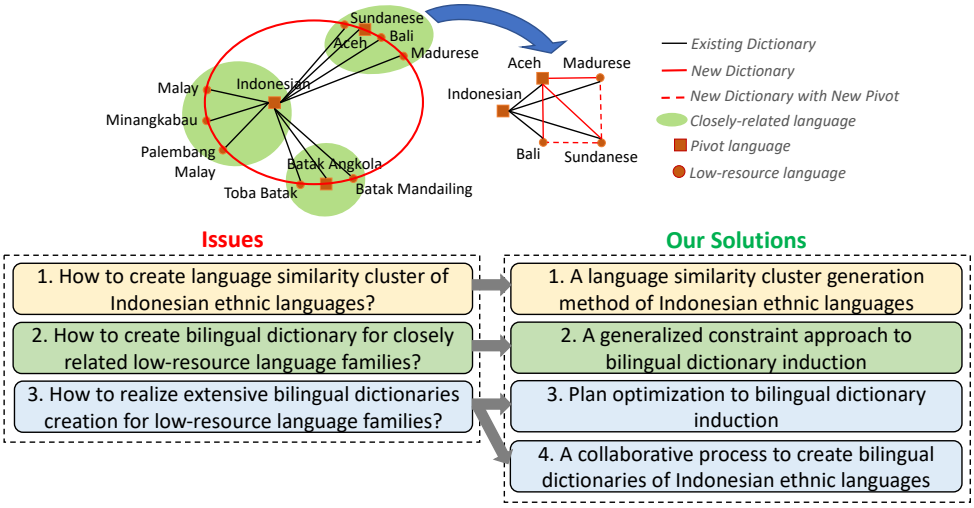


Figure 1.1: Overview of solutions to inducing bilingual dictionary for closely-related languages.

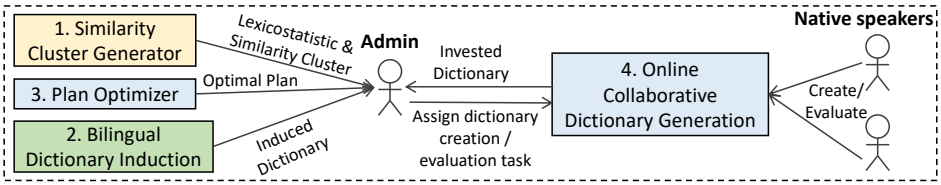


Figure 1.2: Bilingual Lexicon Induction Framework.

1.3 Issues and Approaches

In designing and implementing the bilingual lexicon induction framework for closely related languages we listed four approaches to deal with the following three issues as shown in Figure 1.1. The following issues are derived from the characteristic and availability of Indonesian ethnic languages. As shown in Figure 1.1, there are some bilingual dictionaries available between Indonesian, the official language, and some Indonesian ethnic languages.

However, there are no existing bilingual dictionary among Indonesian ethnic languages. We device solutions to comprehensively create bilingual dictionaries between Indonesian ethnic languages as an example. We depict the framework which consist of four solutions and the relation between them in Figure 1.2.

1. **How to create a language similarity cluster of Indonesian ethnic languages?** There are 341 out of 707 Indonesian ethnic languages facing various degree of language endangerment (trouble/dying) where some of the native speakers do not speak Indonesian well since they are living in a remote area. Unfortunately, there are 13 Indonesian ethnic languages which already extinct. In order to save low-resource languages like Indonesian ethnic languages from language endangerment, enriching the basic language resource, i.e., bilingual dictionary is very crucial. Lexicostatistic/language similarity clusters of Indonesian languages are needed as references to select the target languages. However, to the best of our knowledge, there are no published lexicostatistic/language similarity clusters of Indonesian ethnic languages available. To fill in the gap, we formalize an approach to create language similarity clusters by utilizing ASJP database to generate the language similarity matrix, then generate the hierarchical clusters, and further extract the stable clusters with high language similarities. The hierarchical stable clusters are evaluated utilizing our extended k-means clustering. Finally, the obtained k-means clusters are plotted to a geographical map.

2. **How to create bilingual dictionary for closely related low-resource language families?** Treating pivot-based bilingual lexicon induction

as an optimization problem introduced by our team [Wushouer et al., 2015] with bilingual dictionaries as the only required language resources is promising to enrich low-resource languages compared to previous work that requires parallel or comparable corpora. However, their assumption of one-to-one mapping is too strong to induce the many translation pairs needed to offset resource paucity because few such pairs can be found. Therefore, we generalized the constraint-based bilingual lexicon induction by extending constraints and translation pair candidates from the one-to-one approach to attain more voluminous bilingual dictionary results with many-to-many translation pairs extracted from connected existing and new edges while maintaining a good precision. We further enhance our generalized method by setting two steps utilizing language characteristic to obtaining translation pair results. First, we identify one-to-one cognates by incorporating more constraints and heuristics to improve the quality of the translation result. We then identify the cognates' synonyms to obtain many-to-many translation pairs. In each step, we can obtain more cognate and cognate synonym pair candidates by iterating the n-cycle symmetry assumption until all possible translation pair candidates have been reached. Our method works better on closely related languages. Utilizing a pivot language within each cluster can improve the constraint-based bilingual lexicon induction precision. As shown in Figure 1.1, we can shift a pivot language from Indonesian to an ethnic language within each cluster.

3. **How to realize bilingual dictionaries creation for low-resource language families?** To address this issue, we provide two approaches.

The first approach is about estimating the constraint-based bilingual lexicon induction precision and creating an optimal plan to minimize the cost. The constraint-based bilingual lexicon induction only need bilingual dictionaries as input rather than the expensive parallel and comparable corpora. However, in the actual implementation to comprehensively create bilingual dictionaries from set of target languages, we need to consider the inclusion of manual creation of the bilingual dictionaries by bilingual native speakers if no machine-readable dictionaries are available as input. In the earliest phase of the implementation, one may only have a very few existing machine readable bilingual dictionaries to begin with. Deciding which bilingual dictionary to be invested first by bilingual native speakers or induced with the constraint approach right from the start in order to obtain all possible combination of bilingual dictionaries with a satisfying size from the language set with the minimum total cost to be paid is very difficult because the constraint approach quality needs to be estimated with no input bilingual dictionaries available. A good estimation from the language characteristics included to a plan optimizer to obtain an estimated optimal plan with the least cost is indispensable. We model prior beta distribution of constraint-based bilingual lexicon induction precision with language similarity and polysemy of the topology as beta distribution parameters. We further formalize a plan optimization in creating bilingual dictionaries using Markov Decision Process. We model bilingual dictionary dependency with AND/OR graphs as states, constraint-based bilingual lexicon induction and manual dictionary creation by bilingual native speakers as actions, and utilizing beta distribution of constraint-based bilingual lexicon induction pre-

cision to model cost function and state transition probability.

The second approach focus on the manual creation or evaluation of bilingual dictionary by bilingual native speakers. There is a difficulty in finding a bilingual native speaker of two ethnic languages. One simple solution is to ask two bilingual native speakers of different ethnic languages to collaborate. However they need a tool that can bridge the spatial gap and help them collaborate. For example, to create bilingual dictionary between ethnic language Minangkabau and Sundanese, we can ask bilingual native speaker of Indonesia-Minangkabau and bilingual native speaker of Indonesia-Sundanese to collaborate by explaining the senses with Indonesian language. However, usually bilingual native speaker of Indonesia-Minangkabau can be found in Sumatra island while bilingual native speaker of Indonesia-Sundanese can be found in Java island. They need a tool that can bridge the spatial gap and help them collaborate. We implement the plan optimizer and the constraint approach in creating 10 bilingual dictionaries with 2,000 translation pairs from every combination of 5 languages, i.e., Indonesian, Malay, Minangkabau, Javanese, and Sundanese. We design an online collaborative dictionary generation to bridge spatial gap between native speakers when doing a collaborative creation or evaluation of bilingual dictionary.

1.4 Thesis Outline

This thesis is organized into seven chapters including Chapter 1. The content of the remaining chapters are summarized as follows. Chapter 2 presents the background of this thesis, various bilingual lexicon induction method for low-resource languages. There are three major methods being discussed: extraction from comparable corpora, pivot-based induction approach, and method utilizing language characteristics. Chapter 3 describes a method to generate lexicostatistic and language similarity cluster utilizing ASJP database. We present an example of implementation with Indonesian ethnic languages. We use the result as a reference to select target languages that we want to enrich from 707 Indonesian low-resource languages. Our generalized constraint-based bilingual lexicon induction is presented in Chapter 4. The approach focus on identifying cognate and cognate synonym that will be considered as translation pairs. The result is many-to-many translation pairs with a high recall and good precision. To utilize our constraint approach in enriching the target languages along side with the manual creation by bilingual native speakers, we introduce a plan optimizer to estimate an optimal plan with the least cost in Chapter 5. A collaborative tools to bridge the spatial gap and help bilingual native speakers to collaborate on bilingual dictionary creation and evaluation is presented in Chapter 6. Finally, Chapter 7 concludes this thesis by discussing the summary of contributions made and addressing the possible future directions.

Chapter 2

Bilingual Lexicon Induction Method

In this chapter, we will discuss about the recent bilingual lexicon induction method from comparable corpora with a comprehensive analysis and variation of monolingual signals of translation equivalent, some inexpensive pivot-based approaches that only require multilingual bilingual dictionaries as input, and methods utilizing language characteristics.

2.1 Introduction

Machine readable bilingual lexicons are very useful for natural language processing applications/researches such as cross-language information retrieval [Hull and Grefenstette, 1996] and machine translation [Brown et al., 1990] and being utilized in actual services [Ishida, 2006, Ishida, 2011] to

support intercultural collaboration [Ishida, 2016, Nasution et al., 2017d, Nasution, 2018], but are usually unavailable for low-resource languages. These lexicons are traditionally extracted from parallel corpora, a corpus that contains source texts and their translations. The are various techniques used to extract bilingual lexicons from parallel corpora other than the traditional sentence-aligned bilingual texts [Brown et al., 1990]. [Saers and Wu, 2009] proposed a better method of producing word alignment by training inversion transduction grammars, while recently [Beloucif et al., 2016] utilized English monolingual semantic role labeling to obtain more semantically correct bilingual correlations. An inductive chain learning method [Echizen-ya et al., 2005] can even automatically acquire bilingual rules from parallel corpora without utilizing bilingual dictionary or machine translation.

However, despite good results in the extraction of bilingual lexicons, parallel corpora remains scarce resources for low-resource languages. Thus, research in bilingual lexicon extraction is shifted to comparable corpora [Rapp, 1999, Fung, 2000, Haghighi et al., 2008] which consists of texts sharing common features such as domain, genre, register, or sampling period without having a source text-target text relationship. The approach depends on the assumption that the term and its translation appear in similar contexts [Rapp, 1999, Fung, 2000], which means that a translation equivalent of a source word can be found by identifying a target word with the most similar context vector in a comparable corpus. Identification of good similarity metrics as signals of translation equivalence is the main research challenge in this area. [Irvine and Callison-Burch, 2017] presented a discriminative model of bilingual lexicon induction from comparable corpora and showed experimental results on a wide variety of languages (many of

them are low-resource), for which a wide variety of monolingual corpora and seed bilingual dictionaries are available.

Nevertheless, bilingual lexicon extraction is still highly problematic for most low-resource languages due to the paucity or outright omission of parallel and comparable corpora. The approaches of pivot language [Tanaka and Umemura, 1994][Wushouer et al., 2015] and cognate recognition [Mann and Yarowsky, 2001] have been proven useful in inducing bilingual lexicons for low-resource languages with bilingual dictionary as a sole required language resource. Measuring the semantic distance between two words from the word translation topology based on structure of input bilingual dictionaries is the main research challenge in this area. Closely-related languages share cognates that share most of the semantic or meaning of the lexicons [Lehmann, 2013]. Some linguistics studies [Van Bezooijen and Gooskens, 2005, Gooskens, 2006] show that the percentage of shared cognates, either related directly or via a synonym, constitutes a highly accurate linguistic distance measure based on mutual intelligibility, i.e. the ability of speakers of one language to understand the other language. The higher the percentage of shared cognates between the languages, the lower the linguistic distance, the higher is the level of mutual intelligibility.

2.2 Extraction from Comparable Corpora

The bilingual lexicon induction approach utilizing comparable corpora depends on the assumption that the term and its translation appear in similar contexts [Rapp, 1999, Fung, 2000], which means that a translation equiv-

alent of a source word can be found by identifying a target word with the most similar context vector in a comparable corpus. Identification of good similarity metrics as signals of translation equivalence is the main research challenge in this area. Recently, [Irvine and Callison-Burch, 2017] presented a discriminative model of bilingual lexicon induction that significantly outperforms previous models. Their model is capable of combining a wide variety of features/signals that weakly identify translation equivalence when being used individually. When feature weights are discriminatively set, these signals produce dramatically higher translation quality than previous approaches that combined signals in an unsupervised fashion (e.g., using minimum reciprocal rank) where bilingual lexicon induction is considered as a binary classification problem; a pair of source and target language words are predicted as translations of one another or not. For a given source language word, all target language candidates were separately scored and then ranked. A variety of signals [Rapp, 1995, Fung, 1998, Schafer and Yarowsky, 2002, Klementiev and Roth, 2006, Klementiev et al., 2012] derived from source and target monolingual corpora were used as features and a supervision was used to estimate the strength of each.

All of the individual signals of translation equivalence are weak indicators by themselves, while combining diverse signals increases the translation accuracy [Irvine and Callison-Burch, 2017] which can be observed even using a simple baseline combination method like mean reciprocal rank. Their discriminative approach to combining the signals achieves dramatically improved performance. They used seed dictionary to empirically weight the contributions of the different signals. In this section, their 6 signals of translation equivalence are discussed.

2.2.1 Contextual Similarity

The first signal of translation equivalence proposed by [Irvine and Callison-Burch, 2017] is the common contextual similarity. Context vector representations can be used to compare the similarity of words across two languages in a similar way to how vector space models can be used to compute the similarity between two words in one language by creating vectors that represent their co-occurrence patterns with other words [Turney and Pantel, 2010]. The earliest work in bilingual lexicon induction by [Rapp, 1995] and [Fung, 1995] used the surrounding context of a given word as a clue to its translation. The key to using contextual similarity as a signal of translation equivalence is to find a mapping between the vector space of one language and the vector space of another language. To accomplish this, [Rapp, 1995] originally proposed creating two co-occurrence matrices for the source and target languages, where the co-occurrence between a pair of words is defined as follows:

$$A_{i,j} = \frac{(f(i,j))^2}{f(i)f(j)} \tag{2.1}$$

Where $f(i,j)$ is defined as the number of times words i and j, in the same language, occur in the same context in a large monolingual corpus (using context window of 11 words), and $f(i)$ is the total number of times word i appears in the same corpus. In this original formulation, no bilingual information was used to find the mappings between the vector spaces of the two languages. Instead, after computing the two co-occurrence matrices for the two languages, [Rapp, 1995] iteratively randomly permutes the word order of the matrix for one of the languages and calculates the similarity between the two matrices. The permutation is optimal when the similarity between

the matrices is maximal, which is when the ordered words in the two matrices are most likely to be translations of one another. Later formulations of the problem, including [Fung and Yee, 1998] and [Rapp, 1999], used small seed dictionaries to project word-based context vectors from the vector space of one language into the vector space of the other language. [Rapp, 1999] uses the same projection method as [Fung and Yee, 1998] but uses log-likelihood ratios instead. Once source and target language contextual vectors are built, each position in the source language vectors is projected onto the target side using a seed bilingual dictionary. Finally, contextual similarities are calculated where each projected vector is compared, using any vector comparison method, with the context vector of each target word. Word pairs with high contextual similarity are likely to be translations.

2.2.2 Temporal Similarity

Words' usage over time is also considered as signal of translation equivalence [Irvine and Callison-Burch, 2017]. The intuition is that news stories in different languages will tend to discuss the same world events on the same day and, correspondingly, source and target language words that are translations of one another will appear with similar frequencies over time in monolingual data. For instance, if the English word *World Cup* is used frequently during a particular time span, the Indonesian translation *Piala Dunia* is likely to also be used frequently during that time. The other things that can have periodic temporal signatures are words associated with the Olympics. To calculate temporal similarity, they collected online monolingual newswire over a multi-year period and associate each article with a

time stamp. Each document in the Web crawls of online news Web sites has an associated publication date. Following previous work [Schafer and Yarowsky, 2002, Klementiev and Roth, 2006, Alfonseca et al., 2009], they gather temporal signatures for each source and target language unigram from the time-stamped web crawl data in order to measure temporal similarity.

2.2.3 Orthographic Similarity

Words that have similar spelling are sometimes good translations, because they may be etymologically related, or borrowed words, or the names of people and places. [Irvine and Callison-Burch, 2017] compute the orthographic similarity between a pair of words as the third signal of translation equivalence. They use the Levenshtein edit distance between the two words, normalized by the average of the lengths of the two words. This is more complicated for languages that are written using different scripts. A variety of prior work has focused on the problem of learning mappings between character sets [Yamada and Knight, 1999, Tao et al., 2006, Yoon et al., 2007, Bergsma and Kondrak, 2007, Li et al., 2009, Snyder et al., 2010, Berg-Kirkpatrick and Klein, 2011]. For non-Roman script languages, words are transliterated into the Roman script before measuring orthographic similarity with their candidate English translations. Following previous work [Virga and Khudanpur, 2003, Irvine et al., 2010], transliteration is treated as a monotone character translation task and models are trained on the mined pairs of person names in foreign, non-Roman script languages and English [Irvine and Callison-Burch, 2017].

2.2.4 Topic Similarity

[Irvine and Callison-Burch, 2017] assume that articles that are written about the same topic in two languages are likely to contain words and their translations, even if the articles themselves are written independently and are not translations of one another. They associate articles about the same topic across two languages to compute a topic similarity score to help rank potential translations. Wikipedia articles are used to create topic signatures for words. In order to find a mapping of topics across languages, Wikipedia's interlingual links are used, in a way similar to that used with the small seed bilingual dictionaries to project across the vector spaces for two languages when computing contextual similarity. In order to score how likely a pair of words are to be translations, their topic signatures are compared by counting the words' occurrences in each topic, then the signatures are normalized and finally the resulting vectors are compared. The cosine distance between topic signatures are then computed.

2.2.5 Frequency Similarity

[Irvine and Callison-Burch, 2017] also assume that words that are translations of one another are likely to have similar relative frequencies in monolingual corpora. The frequency similarity of two words are measured as the absolute value of the difference between the log of their relative corpus frequencies. This helps prevent high-frequency closed-class words from being considered viable translations of less-frequent open-class words.

2.2.6 Burstiness Similarity

Burstiness is a measure of how peaked a word's usage is over a particular corpus of documents [Pierrehumbert, 2012]. Bursty words are topical words that tend to appear frequently in a document when some topic is discussed, but do not occur frequently across all documents in a collection. For example, *earthquake* and *election* are considered bursty. In contrast, non-bursty words are those that appear more consistently throughout documents discussing different topics such as *use* and *they*. Previous work [Church and Gale, 1995, Church and Gale, 1999] provide an overview of several ways to measure burstiness empirically. [Irvine and Callison-Burch, 2017] measure the burstiness of a given word in two ways following [Schafer and Yarowsky, 2002]. The first is based on inverse document frequency and the second is similar to that defined by [Church and Gale, 1995], which is the average frequency of a word divided by the percent of documents in which the word appears.

2.3 Pivot-based Induction Approach

A pivot language approach has been applied in machine translation [Tanaka et al., 2009] and service computing [Ishida, 2011] researches. In this section, three pivot-based induction approaches are discussed: the traditional inverse consultation method that only requires bilingual dictionaries, the constraint-based approach that treat bilingual lexicon induction problem as constraint satisfaction problem, and finally a recent work that use WordNets as intermediate resource to generate a new bilingual dictionary.

Figure 2.1: Inverse Consultation Method.

2.3.1 Inverse Consultation

The first work on bilingual lexicon induction to create bilingual dictionary between language A and language C via pivot language B is Inverse Consultation (IC) [Tanaka and Umemura, 1994] by utilizing the structure of input dictionaries to measure the closeness of word meanings and then use the results to prune erroneous translation pair candidates, taking into account that many world languages, especially the low-resource languages are still lack of language resources such as parallel copora and comparable corpora. The IC approach identifies equivalence candidates (EC) of language A words in language C by looking up language A words in $d_{(A,B)}$ and then looking up the resulting B words in $d_{(B,C)}$. These C words equivalence candidates will be looked up in the inverse dictionary $d_{(C,B)}$ and the resulting B words, i.e., Selection Area (SA) are compared to the B equivalences from $d_{(A,B)}$ (one-time inverse consultation), and then further the B words SA can be looked up in the inverse $d_{(B,A)}$ to obtain A words SA to be compared again to the original A words (two-times inverse consultation). For example, in Figure 2.1, a Japanese word 競争 is looked up in dictionary Japanese-English and the

resulting English equivalences (E) are *competition*, *contest*, and *race*. These English equivalences are further looked up in dictionary English-French to obtain the equivalence candidates, which are *compétition*, *concours*, *course*, *race*, and *hate*. *Race* and *hate* are further considered as irrelevant ECs since the pivot English word *race* falls into the following cases:

- The pivot word has multiple meaning with the same spelling. (in this example, the English word *race* has two meanings: *to compete* and *human race*)
- The pivot word has wider meaning than the source word. (in this example, the English word *race* has wider meaning *to hurry* which Japanese word 競争 does not have)
- There are mistakes in the dictionaries.

In Figure 2.1, the equivalence candidates *compétition* is looked up in inverse dictionary French-English to obtain selection area which consist of *contest*, *competition*, and *match*. Then the English words in SA are compared with English words in equivalences (E) of Japanese word 競争. The number of elements in common between SA and E indicate the nearness of the meaning between the EC and the original word.

To analyze the method used to filter wrong translation pair candidates induced via the pivot-based approach, [Saralegi et al., 2011] explored distributional similarity measure (DS) in addition to IC. The analysis showed that IC depends on significant lexical variants in the dictionaries for each meaning in the pivot language, while DS depends on distributions or contexts across two corpora of the different languages. Their analysis also showed that the combination of IC and DS outperformed each used individually.

There are many prior work extending the IC method by utilizing various other language resources such as semantic classes and part-of-speech information [Bond et al., 2001, Bond and Ogura, 2008], WordNet [István and Shoichi, 2009], monolingual corpora [Shezaf and Rappoport, 2010], etc. There are also various techniques proposed to better identify the equivalence candidates. [Sjöbergh, 2005] compared full definitions in order to detect words corresponding to the same sense. However, not all the dictionaries provide this kind of information. [Mausam et al., 2009] researched the use of multiple languages as pivots by using Wiktionary for building a multilingual lexicon, with hypothesis that the more languages used, the more evidences will be found to find translation equivalences. [Tsunakawa et al., 2008] used parallel corpora to estimate translation probabilities between possible translation pairs and setting a minimum threshold to accept equivalence candidates as correct translations. However, even if this approach achieves the best results, it is not adequate to be implemented to low-resource languages due to the scarcity of the parallel corpora.

Even though the IC method seems very suitable for low-resource languages, especially when dictionaries are the only language resource required, unfortunately, for some low-resource languages, it is often difficult to find machine-readable inverse dictionaries and corpora to filter the wrong translation pair candidates. Moreover, since IC relies on pivot language synonyms to identify correct translations, if the relatively rare used meanings had not existed or was missing from the input bilingual dictionaries, IC would not have been able to detect the correct translations. This may result in low recall.

2.3.2 Constraint-based Approach

Our team proposed to treat pivot-based bilingual lexicon induction as an optimization problem[Wushouer et al., 2015]. The assumption was that lexicons of closely-related languages offer one-to-one mapping and share a significant number of cognates (words with similar spelling/form and meaning originating from the same root language). With this assumption, they developed a constraint optimization model to induce an Uyghur-Kazakh bilingual dictionary using Chinese language as the pivot, which means that Chinese words were used as intermediates to connect Uyghur words in an Uyghur-Chinese dictionary with Kazakh words in a Kazakh-Chinese dictionary. They used a graph whose vertices represent words and edges indicate shared meanings; they called this a transgraph following [Soderland et al., 2009]. The steps in their approach are as follows: (1) use two bilingual dictionaries as input, (2) represent them as transgraphs where w_1^A and w_2^A are non-pivot words in language A, w_1^B and w_2^B are pivot words in language B, and w_1^C, w_2^C and w_3^C are non-pivot words in language C, (3) add some new edges represented by dashed edges based on the one-to-one assumption, (4) formalize the problem into conjunctive normal form (CNF) and use the Weighted Partial MaxSAT (WPMaxSAT) solver [Ansótegui et al., 2009] to return the optimized translation results, and (5) output the induced bilingual dictionary as the result. These steps are shown in Figure 2.2.

The one-to-one approach depends only on semantic equivalence, one of the closely-related language characteristics that permit the recognition of cognates between languages assuming that lexicons of closely-related languages offer the one-to-one relation. If language A and C are closely related, for any word in A there exists a unique word in C such that they have exactly

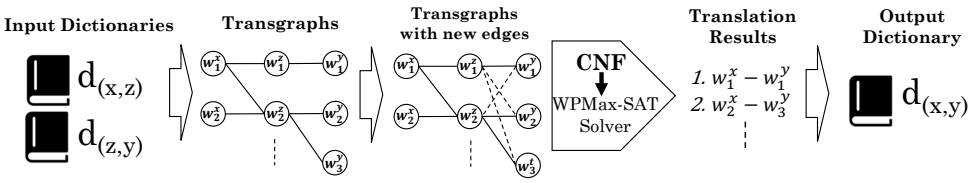


Figure 2.2: One-to-one constraint approach to pivot-based bilingual dictionary induction.

the same meaning, and thus are symmetrically connected via pivot word(s). Such a pair is called a one-to-one pair. They realized that this assumption may be too strong for the general case, but they believed that it was reasonable for closely-related languages like Turkic languages. They believe that their method works best for languages with high-similarity. They tried to improve the precision by utilizing multiple input dictionaries [Wushouer et al., 2014] while still applying the same one-to-one assumption. However, this assumption is too strong to be used for the induction of as many translation pairs as possible to offset resource paucity because the few such pairs are yielded.

2.3.3 Using WordNets as Intermediate Resource (IW)

Recently, there is an interesting work [Lam et al., 2015] which use WordNets as intermediate resource to generate a new bilingual dictionary. To create new bilingual dictionaries $d_{(S,D)}$ where a word in source language S can be translated to a word or multiword expression in a destination language D, they start with existing bilingual dictionary $d_{(S,R)}$, where S is the source language and R is an "intermediate helper" language with a condition that the language R has an available Wordnet linked to the Princeton WordNet.

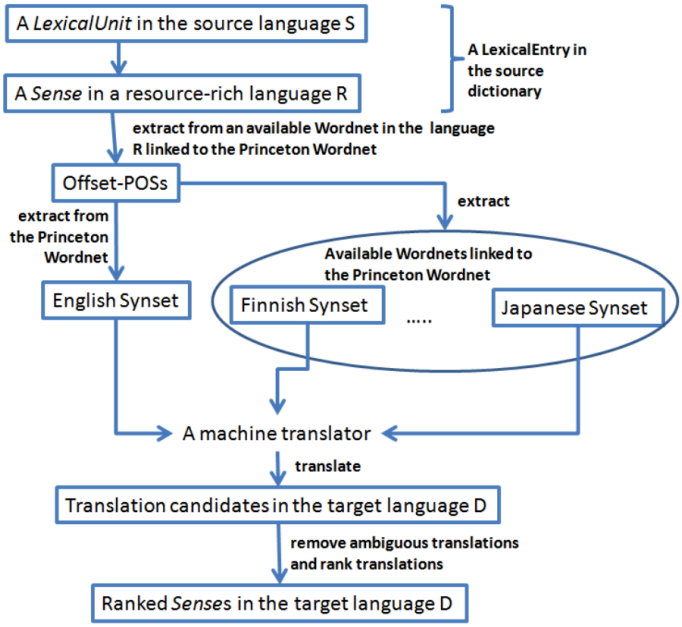


Figure 2.3: The IW approach in Bilingual Dictionary Creation.

The second step is to apply their proposed direct translation approach which uses transitivity to create new bilingual dictionaries from existing dictionaries and a machine translator. They simply translate all senses of a word in language S to language R which can yields ambiguities in the resulting $d_{(S,R)}$. To handle the ambiguities, they extract all offset-POS synsets (cognitive synonyms with common part-of-speech) of the senses from multiple WordNets and further translate those offset-POS synsets to language D with machine translator. They rank the translation candidate of language D based on the occurrence of the offset-POS synset. The IW approach is depicted in Figure 2.3.

This work requires three language resources which are bilingual dictionaries, WordNets and a machine translator which is not always available for

low-resource languages. Nevertheless, since WordNets has richer semantic relation and information than bilingual dictionary such as part-of-speech, hypernym, hyponym, etc., the use of WordNets can potentially solve the ambiguity problem in the pivot-based approach. Even though some low-resource languages have available WordNets, but mostly the size is not as big as Princeton WordNet yet. Moreover, most of other low-resource languages have no available WordNets at all.

2.4 Utilizing Language Characteristics

Cognates are words with similar spelling/form and meaning that have a common etymological origin. For instance, the words *night* (English), *nuit* (French), *noche* (Spanish), *nacht* (German) and *nacht* (Dutch) have the same meaning which is "night" and derived from the Proto-Indo-European **nók^{w}ts* with the same meaning of "night". The closer the etymological relation between languages, the bigger number of cognates can be found [Mann and Yarowsky, 2001]. Since most linguists believe that lexical comparison alone is not a good way to recognize cognates [Campbell, 2013], a more general and basic characteristic of closely-related languages need to be utilized such as cognate pair mostly maintain the semantic or meaning of the lexicons. Even though there is a possibility of a change in one of the meanings of a word in a language, within the families where the languages are known to be closely-related, the possibility of a change is smaller.

Some linguistic studies show that the meaning of a word can be deduced via cognate synonym [Van Bezooijen and Gooskens, 2005, Gooskens, 2006].

Nevertheless, it remains a challenge to find the cognate synonyms when the input dictionaries do not have information about senses/meanings. Therefore, majority of researches on bilingual dictionary creation of intra-family languages have been put high effort on approaches to detecting cognates. There are two approaches from prior work in detecting cognates. The first approach is to make a description on orthography changes of words. In other word, the researcher want to analyze how orthography of a borrowed word should change when it has been introduced into another language. A work [Koehn and Knight, 2000] expanded a list of English-German cognate words by applying well-established transformation rules (e.g. substitution of *k* or *z* by *c* and of *–tat* by *–ty*, as in German *Elektizitat* – English *electricity*). The second approach measure the spelling similarity (also known as form similarity and orthographic similarity) between the given two words. The most well-known approach to measure spelling is edit distance which corresponds to the minimum number of edit operations such as substitution, deletion and insertion required to transform one word into another [Levenshtein, 1966]. A prior work [Mann and Yarowsky, 2001] of cognate recognition using edit distance induce bilingual dictionaries between cross-family languages via a intra-family pivot language. The identified cognate pairs are considered as correct translations. Other related techniques of measuring spelling similarity are the longest common subsequent ratio, which counts the number of letters shared by two strings divided by the length of the longest string [Melamed, 1995], and another method [Danielsson and Muehlenbock, 2000] compare two words by calculating the number of matching consonants. A further extension claimed the importance of genetic cognates by comparing the phonetic similarity of lexemes with the semantic similarity of the glosses [Inkpen et al., 2005].

Chapter 3

A Language Similarity Cluster Generation Method of Indonesian Ethnic Languages

3.1 Introduction

Indonesia has a population of 221,398,286 and 707 living languages which cover 57.8% of Austronesian Family and 30.7% of languages in Asia [Lewis et al., 2015]. There are 341 Indonesian ethnic languages facing various degree of language endangerment (trouble/dying) where some of the native speakers do not speak Indonesian well since they are in remote areas. Unfortunately, there are 13 Indonesian ethnic languages which already extinct. In order to save low-resource languages like Indonesian ethnic languages from language endangerment, prior works tried to enrich the basic language

resource, i.e., bilingual dictionary [Tanaka and Umemura, 1994, Wushouer et al., 2015]. The lexicostatistic/language similarity clusters of the low-resource languages can be used to select the target languages. However, to the best of our knowledge, there are no published lexicostatistic/language similarity clusters of Indonesian ethnic languages. To fill the void, we address this research goal: Formulating an approach of creating a language similarity cluster. We first obtain 40-item word lists from the Automated Similarity Judgment Program (ASJP), further generate the language similarity matrix, then generate the hierarchical and k-means clusters, and finally plot the generated clusters to a geographical map [Nasution et al., 2017c].

3.2 Automated Similarity Judgment Program

Historical linguistics is the scientific study of language change over time in term of sound, analogical, lexical, morphological, syntactic, and semantic information [Campbell, 2013]. Comparative linguistics is a branch of historical linguistics that is concerned with language comparison to determine historical relatedness and to construct language families [Lehmann, 2013]. Many methods, techniques, and procedures have been utilized in investigating the potential distant genetic relationship of languages, including lexical comparison, sound correspondences, grammatical evidence, borrowing, semantic constraints, chance similarities, sound-meaning isomorphism, etc [Campbell and Poser, 2008]. The genetic relationship of languages is used to classify languages into language families. Closely-related languages are those that came from the same origin or proto-language, and belong to the same language family.

Swadesh List is a classic compilation of basic concepts for the purposes of historical-comparative linguistics. It is used in lexicostatistics (quantitative comparison of lexical cognates) and glottochronology (chronological relationship between languages). There are various version of swadesh list with a number of words equal 225 [Swadesh, 1950], 215 & 200 [Swadesh, 1952], and lastly 100 [Swadesh, 1955]. To find the best size of the list, Swadesh states that "The only solution appears to be a drastic weeding out of the list, in the realization that quality is at least as important as quantity. Even the new list has defects, but they are relatively mild and few in number." [Swadesh, 1955]

A widely-used notion of string/lexical similarity is the edit distance or also known as Levenshtein Distance (LD): the minimum number of insertions, deletions, and substitutions required to transform one string into the other [Levenshtein, 1966]. For example, LD between "kitten" and "sitting" is 3 since there are three transformations needed: kitten $\rightarrow$ sitten (substitution of "s" for "k"), sitten $\rightarrow$ sittin (substitution of "i" for "e"), and finally sittin $\rightarrow$ sitting (insertion of "g" at the end).

There are a lot of previous works using Levenshtein Distances such as dialect groupings of Irish Gaelic [Kessler, 1995] where they gather the data from questionnaire given to native speakers of Irish Gaelic in 86 sites. They obtain 312 different Gaelic words or phrases. Another work is about dialect pronunciation differences of 360 Dutch dialects [Heeringa, 2004] which obtain 125 words from Reeks Nederlandse Dialectatlassen. They normalize LD by dividing it by the length of the longer alignment. [Tang and Van Heuven, 2015] measure linguistic similarity and intelligibility of 15 Chinese dialects and obtain 764 common syllabic units. [Petroni and Serva,

2008] define lexical distance between two words as the LD normalized by the number of characters of the longer of the two. [Wichmann et al., 2010] extend Petroni definition as LDND and use it in Automated Similarity Judgment Program (ASJP).

The ASJP, an open source software was proposed by [Holman et al., 2011] with the main goal of developing a database of Swadesh lists [Swadesh, 2017] for all of the world's languages from which lexical similarity or lexical distance matrix between languages can be obtained by comparing the word lists. The classification is based on 100-item reference list of Swadesh [Swadesh, 1955] and further reduced to 40 most stable items [Holman et al., 2008]. The item stability is a degree to which words for an item are retained over time and not replaced by another lexical item from the language itself or a borrowed element. Words resistant to replacement are more stable. Stable items have a greater tendency to yield cognates (words that have a common etymological origin) within groups of closely related languages.

3.3 Language Similarity Clustering Approach

We formalize an approach to create language similarity clusters by utilizing ASJP database to generate the language similarity matrix, then generate the hierarchical clusters, and further extract the stable clusters with high language similarities. The hierarchical stable clusters are evaluated utilizing our extended k-means clustering. Finally, the obtained k-means clusters are plotted to a geographical map. The flowchart of the whole process is shown in Figure 3.1.

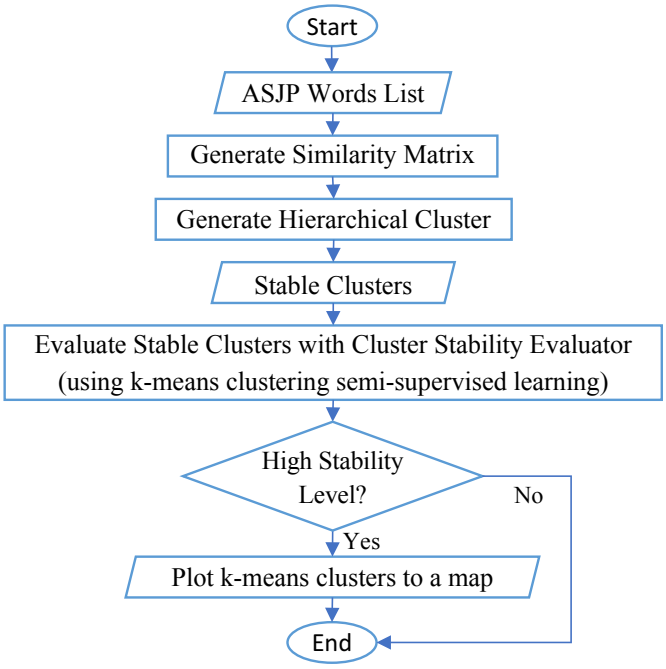


Figure 3.1: Flowchart of Generating Language Similarity Clusters

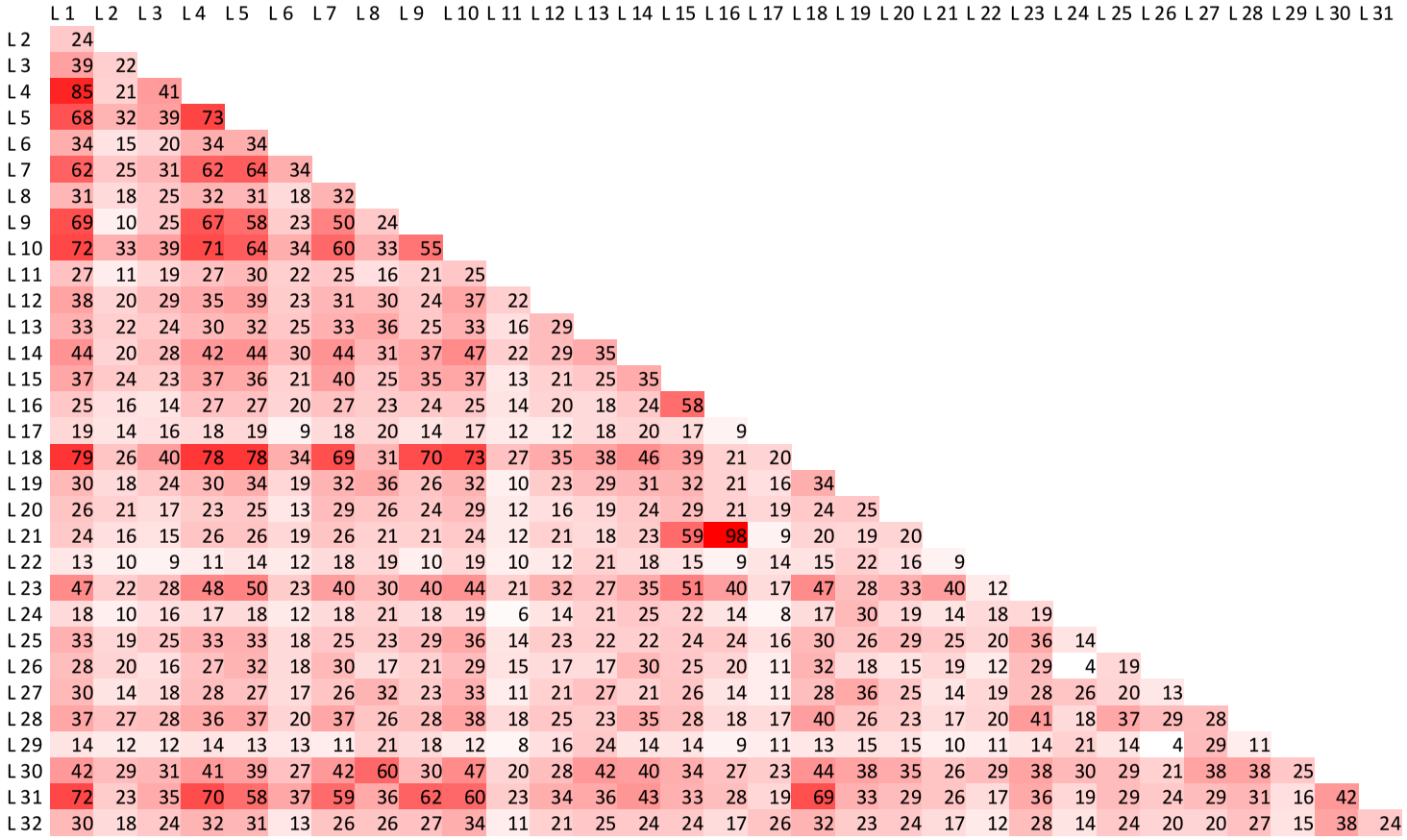
We choose Indonesian ethnic languages as the target languages. We obtain words list of 119 Indonesian ethnic languages with the number of speakers at least 100,000. However, it is difficult to classify 119 languages and obtain a valuable information from the generated clusters, therefore, we further filtered the target languages based on the number of speaker and availability of the language information in Wikipedia. We obtain 32 target languages as shown in Table 3.1 from the intersection between 46 Indonesian ethnic languages with number of speaker above 300,000 provided by Wikipedia and 119 Indonesian ethnic languages with number of speaker above 100,000 provided by ASJP.

We further generate the similarity matrix of those 32 languages as shown

Table 3.1: List of 32 Indonesian Ethnic Languages Ranked by Population According to ASJP database

Code	Population	Language
L 1	232004800	INDONESIAN
L 2	84300000	OLD OR MIDDLE JAVANESE
L 3	34000000	SUNDANESE
L 4	15848500	MALAY
L 5	15848500	PALEMBANG MALAY
L 6	6770900	MADURESE
L 7	5530000	MINANGKABAU
L 8	5000000	BUGINESE
L 9	5000000	BETAWI
L 10	3502300	BANJARESE MALAY
L 11	3500032	ACEH
L 12	3330000	BALI
L 13	2130000	MAKASAR
L 14	2100000	SASAK
L 15	2000000	TOBA BATAK
L 16	1100000	BATAK MANDAILING
L 17	1000000	GORONTALO
L 18	1000000	JAMBI MALAY
L 19	900000	MANGGARAI
L 20	770000	NIAS NORTHERN
L 21	750000	BATAK ANGKOLA
L 22	700000	UAB METO
L 23	600000	KARO BATAK
L 24	500000	BIMA
L 25	470000	KOMERING
L 26	350000	REJANG
L 27	331000	TOLAKI
L 28	300000	GAYO
L 29	300000	MUNA
L 30	250000	TAE
L 31	245020	AMBONESE MALAY
L 32	230000	MONGONDOW

in Figure 3.2. We added a white-red color scale where white color means the two languages are totally different (0% similarity) and the reddest color means the two languages are exactly the same (100% similarity). For a better clarity and to avoid redundancy, we only show the bottom-left part of the table. The headers follow the language code in Table 3.1.



	L 1	L 2	L 3	L 4	L 5	L 6	L 7	L 8	L 9	L 10	L 11	L 12	L 13	L 14	L 15	L 16	L 17	L 18	L 19	L 20	L 21	L 22	L 23	L 24	L 25	L 26	L 27	L 28	L 29	L 30	L 31
L 2	24																														
L 3	39	22																													
L 4	85	21	41																												
L 5	68	32	39	73																											
L 6	34	15	20	34	34																										
L 7	62	25	31	62	64	34																									
L 8	31	18	25	32	31	18	32																								
L 9	69	10	25	67	58	23	50	24																							
L 10	72	33	39	71	64	34	60	33	55																						
L 11	27	11	19	27	30	22	25	16	21	25																					
L 12	38	20	29	35	39	23	31	30	24	37	22																				
L 13	33	22	24	30	32	25	33	36	25	33	16	29																			
L 14	44	20	28	42	44	30	44	31	37	47	22	29	35																		
L 15	37	24	23	37	36	21	40	25	35	37	13	21	25	35																	
L 16	25	16	14	27	27	20	27	23	24	25	14	20	18	24	58																
L 17	19	14	16	18	19	9	18	20	14	17	12	12	18	20	17	9															
L 18	79	26	40	78	78	34	69	31	70	73	27	35	38	46	39	21	20														
L 19	30	18	24	30	34	19	32	36	26	32	10	23	29	31	32	21	16	34													
L 20	26	21	17	23	25	13	29	26	24	29	12	16	19	24	29	21	19	24	25												
L 21	24	16	15	26	26	19	26	21	21	24	12	21	18	23	59	98	9	20	19	20											
L 22	13	10	9	11	14	12	18	19	10	19	10	12	21	18	15	9	14	15	22	16	9										
L 23	47	22	28	48	50	23	40	30	40	44	21	32	27	35	51	40	17	47	28	33	40	12									
L 24	18	10	16	17	18	12	18	21	18	19	6	14	21	25	22	14	8	17	30	19	14	18	19								
L 25	33	19	25	33	33	18	25	23	29	36	14	23	22	22	24	24	16	30	26	29	25	20	36	14							
L 26	28	20	16	27	32	18	30	17	21	29	15	17	17	30	25	20	11	32	18	15	19	12	29	4	19						
L 27	30	14	18	28	27	17	26	32	23	33	11	21	27	21	26	14	11	28	36	25	14	19	28	26	20	13					
L 28	37	27	28	36	37	20	37	26	28	38	18	25	23	35	28	18	17	40	26	23	17	20	41	18	37	29	28				
L 29	14	12	12	14	13	13	11	21	18	12	8	16	24	14	14	9	11	13	15	15	10	11	14	21	14	4	29	11			
L 30	42	29	31	41	39	27	42	60	30	47	20	28	42	40	34	27	23	44	38	35	26	29	38	30	29	21	38	38	25		
L 31	72	23	35	70	58	37	59	36	62	60	23	34	36	43	33	28	19	69	33	29	26	17	36	19	29	24	29	31	16	42	
L 32	30	18	24	32	31	13	26	26	27	34	11	21	25	24	24	17	26	32	23	24	17	12	28	14	24	20	20	27	15	38	24

Figure 3.2: Lexicostatistic / Similarity Matrix of 32 Indonesian Ethnic Languages by ASJP (%)

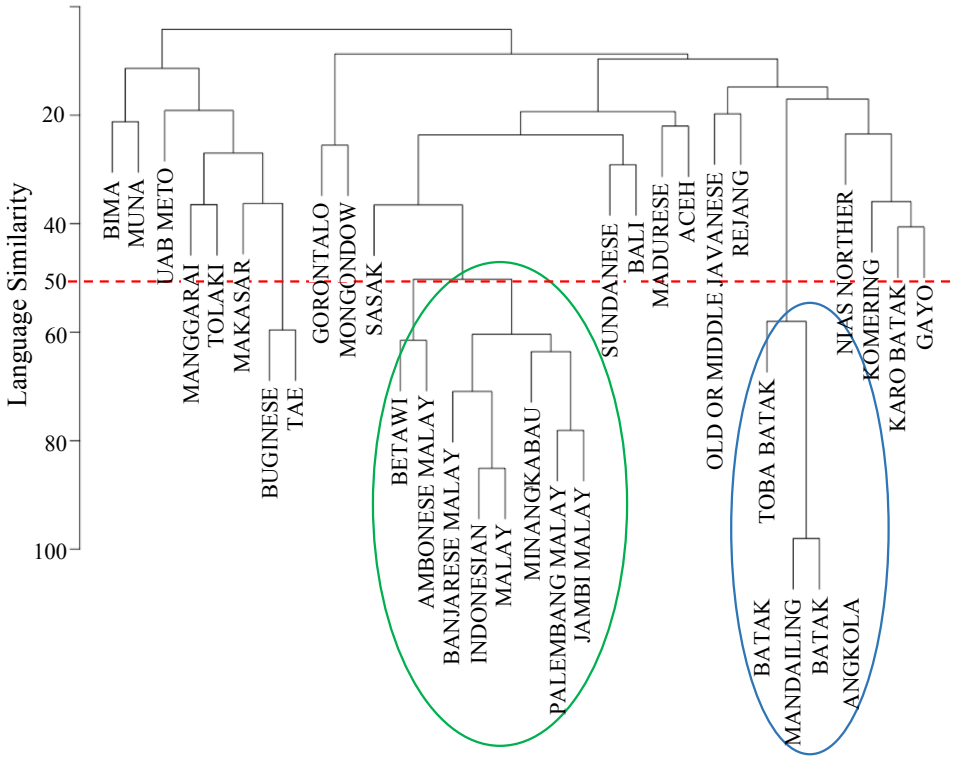


Figure 3.3: Hierarchical Clusters Dendogram of 32 Indonesian Ethnic Languages; Method: Complete.

Hierarchical clustering is an approach which builds a hierarchy from the bottom-up, and does not require us to specify the number of clusters beforehand. The algorithm works as follows: (1) Put each data point in its own cluster; (2) Identify the closest two clusters and combine them into one cluster; (3) Repeat the above step until all the data points are in a single cluster. Once this is done, it is usually represented by a dendrogram like structure. There are a few ways to determine how close two clusters are: (1) Complete linkage clustering: find the maximum possible distance between points belonging to two different clusters; (2) Single linkage clustering: find the min-

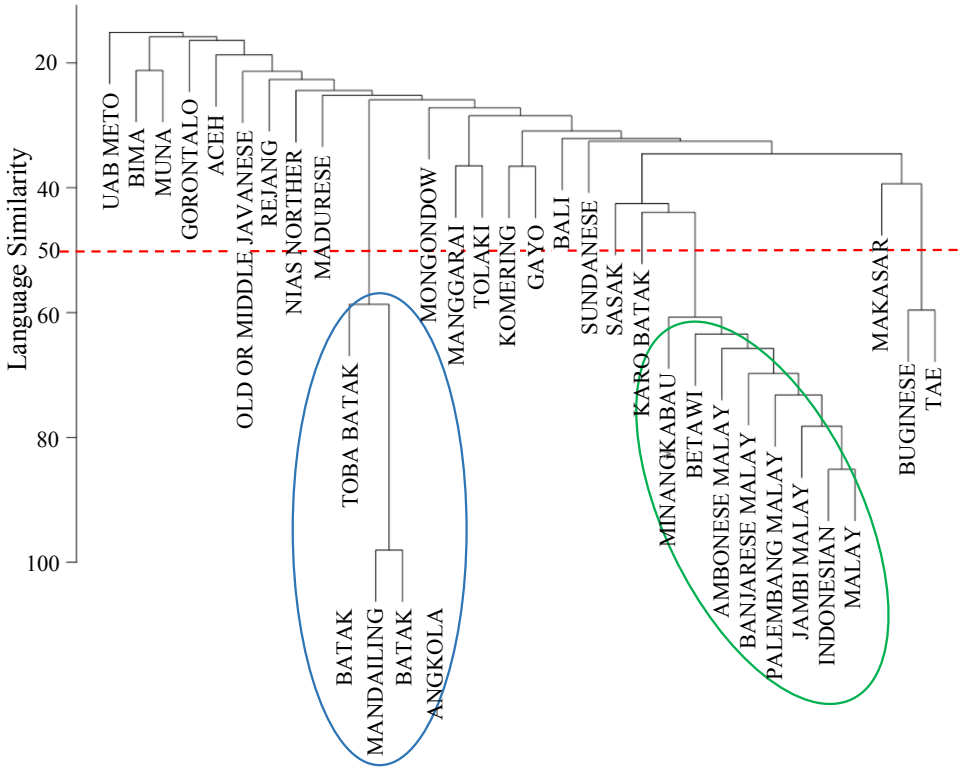


Figure 3.4: Hierarchical Clusters Dendogram of 32 Indonesian Ethnic Languages; Method: Average.

imum possible distance between points belonging to two different clusters; (3) Mean/Average linkage clustering: find all possible pairwise distances for points belonging to two different clusters and then calculate the average; (4) Centroid linkage clustering: find the centroid of each cluster and calculate the distance between centroids of two clusters. Complete linkage and mean (average) linkage clustering are the ones used most often. We generate the distance matrix from the similarity matrix shown in Figure 3.2 and further generate the hierarchical clusters with hclust function with a complete linkage clustering method as shown in Figure 3.3 and a mean linkage clustering

method as shown in Figure 3.4 using R, a free software environment for statistical computing and graphics.

From those two hierarchical clusters, we select two stable clusters that always grouped together despite of changing the linkage clustering method. The first cluster consists of TOBA BATAK, BATAK MANDAILING, and BATAK ANGKOLA, while the second cluster consists of MINANGKABAU, BETAWI, AMBONESE MALAY, BANJARESE MALAY, PALEMBANG MALAY, JAMBI MALAY, MALAY, and Indonesia. Since the two stable custers have language similarities above 50% between the languages, they are good clusters to be referred when selecting target languages for computational linguistic researches that depends on language similarity or cognate recognition for inducing bilingual lexicons from the target languages [Mann and Yarowsky, 2001, Wushouer et al., 2015]. The two clusters are actually enough for selecting the target languages for those researches. However, we still need to evaluate the stability of those clusters and we also need to identify the low language similarities clusters in order to grasp the whole picture of Indonesian ethnic languages. Thus, we utilize the alternative clustering approach which is a k-means clustering.

K-means clustering is an unsupervised learning algorithm that tries to cluster data based on their similarity. Unsupervised learning means that there is no outcome to be predicted, and the algorithm just tries to find patterns in the data. In k-means clustering, we have to specify the number of clusters we want the data to be grouped into. The algorithm works as follows: (1) The algorithm randomly assigns each observation to a cluster, and finds the centroid of each cluster; (2) Then, the algorithm iterates through two steps: (2a) Reassign data points to the cluster whose centroid is closest; (2b) Cal-

ALGORITHM 1: Cluster Stability Evaluator

Input: *similarityMatrix*, *stableClusters*, *minimumK*, *maximumTrial*;
Output: *stabilityLevel*

```
1  trial ← 1;
2  currentK ← minimumK;
3  maximumK ← length(similarityMatrix);
4  scale2D ← cmdscale(similarityMatrix); // multidimensional to
     2D scaling
5  while currentK <= maximumK do
6      successfulTrial ← 0;    // initialized for each currentK
7      while trial <= maximumTrial do
8          kClusters ← kmeans(scale2D, currentK);
9          if stableClusters distinctly found in kClusters then
10             successfulTrial++;
11             trial++;    // try again with the same number
                 of cluster (currentK)
12         end
13     end
14     stabilityLevel[currentK] ← successfulTrial/maximumTrial;
15     currentK++;     // increase the number of clusters
16     trial ← 1                // reset the number of trial
17 end
18 return stabilityLevel;
```

culate new centroid of each cluster. These two steps are repeated until the within cluster variation cannot be reduced any further. The within cluster variation is calculated as the sum of the euclidean distance between the data points and their respective cluster centroids.

It is well known that standard agglomerative hierarchical clustering techniques are not tolerant to noise [Nagy, 1968, Narasimhan et al., 2006]. There are many previous works on finding clusters which robust to noise [Balcan et al., 2014, Guha et al., 1999, Langfelder and Horvath, 2012]. However,

to evaluate the stability of the hierarchical stable clusters, we introduced a simple approach of calculating their stability level of being grouped together despite of changing the number of k-means clusters. We extend the k-means clustering unsupervised learning to a k-means clustering semi-supervised learning as shown in Algorithm 1 by labeling the two hierarchical stable clusters beforehand.

3.4 Result and Analysis

Initially, we manually conduct several trials to estimate the minimum and maximum number of k-means cluster to obtain clusters which consist of the stable clusters distinctly. Based on the initial trials, we estimate the $minimum_k = 4$ and $maximum_k = 21$. Then, we calculate the stability level of the two hierarchical stable clusters where the number of clusters ranging from $minimum_k = 4$ to $maximum_k = 21$ following Algorithm 1. We have five sets of experiments with the $maximum_{trial}$ equals 50, 500, 5,000, 50,000, and 500,000. In each experiment, a stability level of the two hierarchical stable clusters is measured for each number of k-means clusters by calculating the success rate of obtaining the two hierarchical stable clusters in the generated k-clusters.

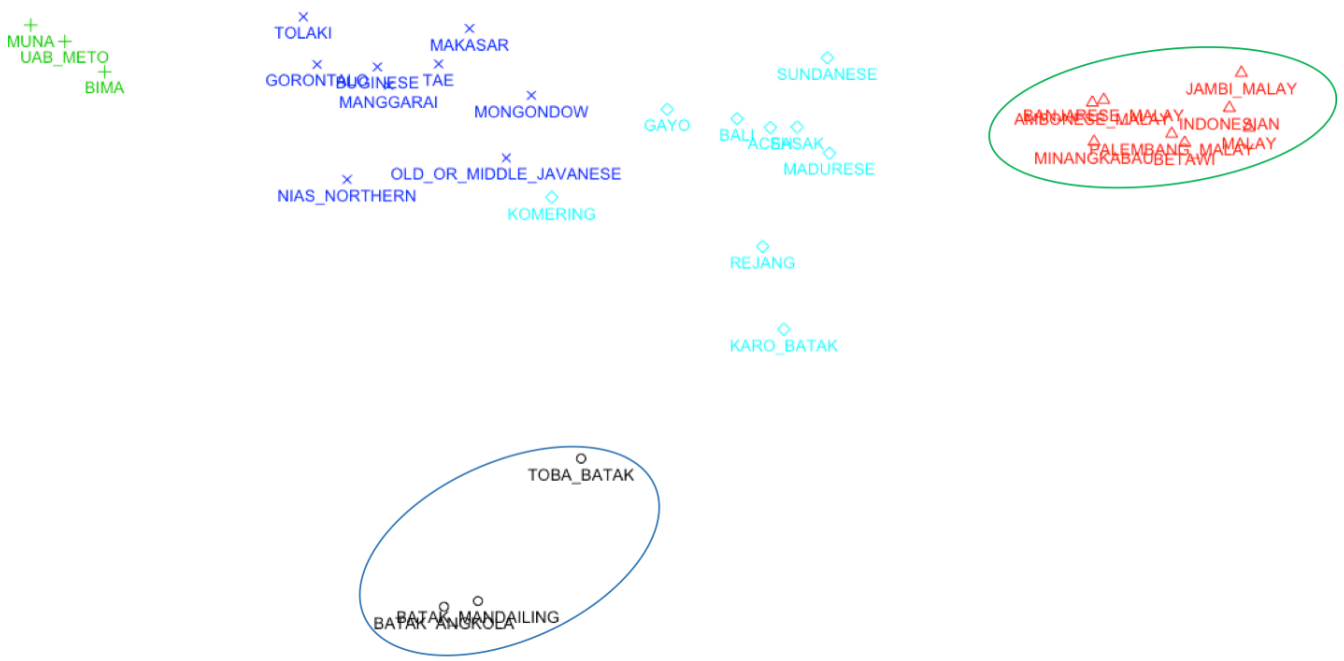


Figure 3.5: K-means Clusters of 32 Indonesian Ethnic Languages – 5 Clusters

Figure 3.6: Similarity Clusters Map of 32 Indonesian Ethnic Languages – 5 Clusters

The higher the number of the trial, the more likely we can distinctly find the two hierarchical stable clusters in the generated k-clusters with a big number of clusters. For example, within 50 trials, we can not find the two hierarchical stable clusters distinctly in the generated k-clusters for big number of clusters ($k > 14$). However, within 50,000 and 500,000 trials, we can find the two hierarchical stable clusters distinctly in the generated k-clusters for all number of clusters between the $minimum_k = 4$ and the $maximum_k = 21$, even though the success rate is getting lower as the number of clusters increases. For all five experiments, the stability level of the two hierarchical stable clusters is the highest (0.78) on 5 clusters. Therefore, we take the 5 clusters as shown in Figure 3.5 as the best clusters of Indonesian ethnic languages to be referred when selecting target languages for computational linguistic researches that depends on language similarity or cognate recognition. We further plot the 5 clusters to a geographical map as shown in Figure 3.6.

3.5 Conclusion

We utilized ASJP database to generate the language similarity matrix, then generate the hierarchical clusters with complete linkage and mean linkage clustering, and further extract two stable clusters with the highest language similarities. We apply our extended k-means clustering semi-supervised learning to evaluate the stability level of the hierarchical stable clusters being grouped together despite of changing the number of clusters. The higher the number of the trial, the more likely we can distinctly find the two hierarchical stable clusters in the generated k-clusters. However, for all five exper-

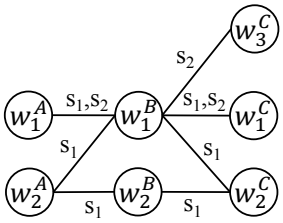
iments, the stability level of the two hierarchical stable clusters is the highest (0.78) on 5 clusters. Therefore, we take the 5 clusters as the best clusters of Indonesian ethnic languages to be referred to select target languages for computational linguistic researches that depends on language similarity or cognate recognition. Our algorithm can be used to find and evaluate other stable clusters of Indonesian ethnic languages or other language sets.

Chapter 4

A Generalized Constraint Approach to Bilingual Dictionary Induction

4.1 Introduction

Treating pivot-based bilingual lexicon induction as an optimization problem with bilingual dictionaries as the only required language resources is promising to enrich low-resource languages [Wushouer et al., 2015]. However, the assumption of one-to-one mapping is too strong to induce the many translation pairs needed to offset resource paucity because few such pairs can be found. Therefore, we generalized the constraint-based bilingual lexicon induction by extending constraints and translation pair candidates from the one-to-one approach to attain more voluminous bilingual dictionary re-



Step 1 - Recognize Cognates:
- $w_1^A - w_1^C$ (s_1, s_2)
- $w_2^A - w_2^C$ (s_1)

Step 2 - Recognize Cognate Synonyms:
- $w_1^A - w_1^C$ (s_1, s_2) → $w_1^A - w_2^C$ (s_1), $w_1^A - w_3^C$ (s_2)
- $w_2^A - w_2^C$ (s_1) → $w_2^A - w_1^C$ (s_1)

Figure 4.1: Strategy to recognize cognates and cognate synonyms.

sults with many-to-many translation pairs extracted from connected existing and new edges [Nasution et al., 2016]. We further enhance our generalized method by setting two steps to obtaining translation pair results. First, we identify one-to-one cognates by incorporating more constraints and heuristics to improve the quality of the translation result. We then identify the cognates' synonyms to obtain many-to-many translation pairs. In each step, we can obtain more cognate and cognate synonym pair candidates by iterating the n-cycle symmetry assumption until all possible translation pair candidates have been reached [Nasution et al., 2017a].

4.2 Cognate and Cognate Synonym Recognition

By utilizing linguistic information, we establish a strategy to obtain many-to-many translation pairs from a transgraph. The first step is to recognize one-to-one cognates in the transgraph which share all their senses. Once a list of cognates is obtained, the next step is to recognize cognate synonyms in the transgraph; those that share part/all senses with the cognate and so are mutually connected to some/all pivot words. Those two steps are easy tasks when the input dictionaries have sense/meaning information as

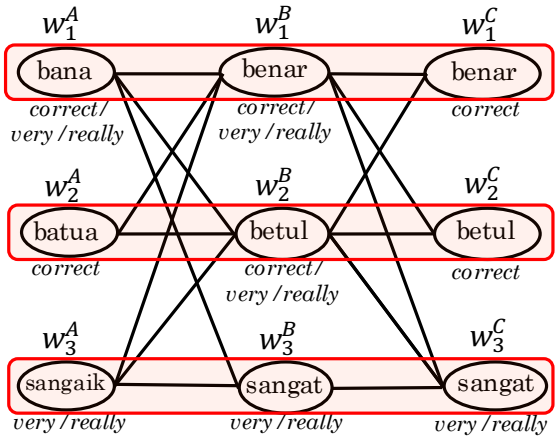


Figure 4.2: Cognate and cognate synonym example.

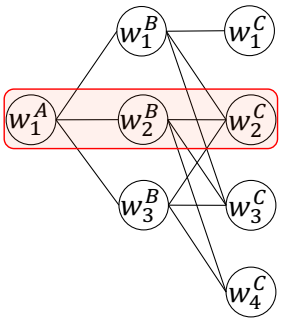
shown in Figure 4.1 where a cognate pair (w_1^A, w_1^C) share two senses, i.e., s_1 and s_2 through pivot word w_1^B and a cognate pair (w_2^A, w_2^C) only share s_1 through pivot word w_1^B and w_2^B. Since for low-resource languages, a machine-readable bilingual dictionary with sense information is scarce, we regard connected words share at least one sense/meaning. Thus, we assume that non-pivot words which are symmetrically connected via pivot word(s) potentially share all their senses and so being a cognate.

We want to utilize a general and basic characteristic of closely-related languages, which is: a cognate pair mostly maintain the semantic or meaning of the lexicons, because a probability of a change in one of the meanings of a word in a language within the families where the languages are known to be closely-related is smaller. Since our approach targets the closely-related languages, it is safe to make the following assumption based on the semantic characteristic of closely-related languages: *Given a pair of words, w_i^A of language A and w_k^C of language C, if they are cognates, they share all of their*

senses/meanings and are symmetrically connected through pivot word(s) from language B. We call this the symmetry assumption. Unfortunately, in some cases, symmetry assumption is inadequate to eliminate wrong cognate from the cognate pair candidates when a pivot-word has multiple indegree/outdegree. To correctly find cognates, not only the meaning (which is predicted by shared edges), but also the form need to be considered. We add form-similarity/lexical distance rate as a new heuristic in finding cognates following [Melamed, 1995] using the Longest Common Subsequence Ratio (LCSR).

Some linguistic studies show that the meaning of a word can be deduced via cognate synonym [Van Bezooijen and Gooskens, 2005, Gooskens, 2006]. For instance, in Figure 4.2, w_1^A, w_2^A and w_3^A are words in Minangkabau language (min), w_1^B, w_2^B and w_3^B are words in Indonesian language (ind) and w_1^C, w_2^C and w_3^C are words in Malay language (zlm). When we connect words in non-pivot language A and C via pivot words B based on shared meaning between the words, we can get translation results from language A to C. In this example, we have information about senses/meanings for all words in input dictionaries and there are three cognates which are (w_1^A, w_1^B, w_1^C), (w_2^A, w_2^B, w_2^C), and (w_3^A, w_3^B, w_3^C) as indicated within the same box in Figure 4.2. A cognate $w_1^A - w_1^C$ and non-cognates $w_1^A - w_2^C$ and $w_1^A - w_3^C$ are correct translations since w_1^C, w_2^C and w_3^C are synonymous.

Nevertheless, it remains a challenge to find the cognate synonyms when the input dictionaries do not have information about senses/meanings. As shown in Figure 4.3, to recognize cognate synonyms, firstly, we need to recognize synonyms of w_2^C based on ratio of shared connectivity with the pivot word(s), since we assume that synonymous words are connected to common



For example, from step 1, cognate pair is identified: $w_1^A - w_2^C$
Step 2 – Recognize cognate synonyms:
a. Recognize synonyms of w_2^C based on ratio of shared connectivity with the pivot word(s):
- Probability of $w_2^C - w_3^C$ being synonym: 3/3 = 1
- Probability of $w_2^C - w_4^C$ being synonym: 2/3 = 0.67
- Probability of $w_2^C - w_1^C$ being synonym: 1/3 = 0.33

b. Pair w_1^A with the synonyms of w_2^C as cognate synonym pairs:
- $w_1^A - w_3^C, w_1^A - w_4^C, w_1^A - w_1^C$

Figure 4.3: Cognate Synonym Recognition.

pivot word(s). Then, w_1^A will be paired with the recognized synonyms of w_2^C to obtain cognate synonym pairs. The higher the ratio of shared connectivity between a synonym of w_2^C with the pivot words (w_1^B, w_2^B, w_3^B), the higher the probability of the synonym being a translation pair with w_1^A.

Finally, by recognizing both cognate pairs and cognate synonym pairs, we can obtain many-to-many translation results.

4.3 Generalization of Constraint-based Lexicon Induction Framework

We generalize the constraint-based lexicon induction framework by extending the existing one-cycle symmetry assumption into the n-cycle symmetry assumption and identify cognates and cognate synonyms by utilizing four heuristics to improve the quality and quantity of the translation pair results.

4.3.1 Tripartite Transgraph

To model translation connectivity between language A and C via pivot language B, we define the tripartite transgraph, which is a tripartite graph in which a vertex represents a word and an edge represents the indication of shared meaning(s) between two vertices. Two tripartite transgraphs can be joined if there exists at least one edge connecting a pivot vertex in one tripartite transgraph to one non-pivot vertex in the other tripartite transgraph. To maintain the basic form of a tripartite transgraph with n number of pivot words (at least 1 pivot per transgraph), each pivot word must be connected to at least one word in every non-pivot language, and there has to be a path connecting all pivot words via non-pivot words. Hereafter, we abbreviate the tripartite transgraph to transgraph.

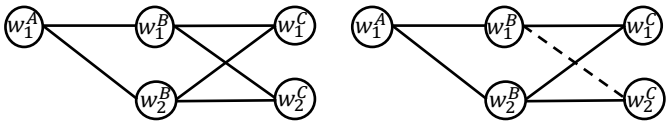
In this research, we assume that the input dictionaries contain no sense information. After modeling the translation connectivity from the input dictionaries as transgraphs, we further analyze the shared edges between the non-pivot vertices and the pivot vertices to predict the shared meanings between them. We then formalize the problem into Conjunctive Normal Form (CNF) and using WPMaxSAT solver to return the most probable correct translation results.

Sometimes, for high-resource languages where the input dictionaries have many shared meanings via the pivot words, a big transgraph can be generated which potentially leads to excessive computational complexity when we formalize and solve it. Nevertheless, for low-resource languages where we can expect the input dictionaries to have just a few shared meanings via the pivot words, transgraph size is small enough to make its formalization

and solution feasible. Therefore, for the sake of simplicity, we ignore big transgraphs in our experiments.

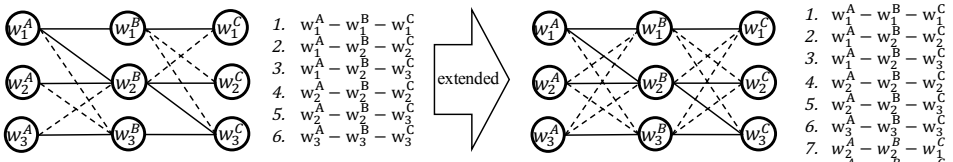
4.3.2 N-cycle Symmetry Assumption

Machine-readable bilingual dictionaries are rarely available for low-resource languages like Indonesian ethnic languages. It is even difficult to find sizable printed bilingual dictionary with acceptable quality for Indonesian ethnic languages. In the currently available machine-readable or printed dictionaries, we can expect to find missed senses/meanings that would lead to asymmetry in the transgraph. The expected missed senses are represented as dashed edges in the transgraph as depicted in Figure 4.4b. The one-to-one approach only considers translation pair candidates from existing connected solid edges in the transgraph as shown in Figure 4.5a. To fully satisfy symmetry constraint in the transgraph, we extend the existing one-cycle symmetry assumption to the n-cycle symmetry assumption while considering new translation pair candidates from the new dashed edges. As shown in Figure 4.5b, during the second cycle, the previously new dashed edges developed in the first cycle are taken to exist, therefore, we can extract translation pair candidates not only from the solid edges but also from the previously added dashed-edges. Users can input the maximum cycle for the experiment.



(a) Symmetry Transgraph (b) Asymmetry Transgraph

Figure 4.4: Symmetry and Asymmetry Transgraphs.



(a) Translation pair candidates from **existing solid-edges** in 1-cycle symmetry assumption

(b) Translation pair candidates from **existing solid edges** and ***previously added dashed-edges*** in 2-cycle symmetry assumption

Figure 4.5: N-cycle symmetry assumption extension.

4.3.3 Formalization

Constraint optimization problem formalism has been used in solving many natural language processing and web service composition related problems [Matsuno and Ishida, 2011, Hassine et al., 2006]. Our team [Wushouer et al., 2015] formalized bilingual lexicon induction as a WPMaxSAT problem. In this thesis, we follow the same formulation. A literal is either a Boolean variable x or its negation $\neg x$. A clause C is a disjunction of literals $x_1 \vee ... \vee x_n$. A unit clause is a clause consisting of a single literal. A weighted clause is a pair (C, ω), where C is a clause and ω is a natural number representing the penalty for falsifying the clause C. If a clause is hard, the corresponding weight is infinity. The propositional formula φ_c^{ω} in CNF [Biere et al., 2009] is a conjunction of one or more clauses $C_1 \wedge ... \wedge C_n$. CNF formula with soft clauses is represented as φ_c^{+} and φ_c^{∞} represents a

CNF formula with hard clauses. The WPMaxSAT problem for a multiset of weighted clauses C is the problem of finding an optimal assignment to the variables of C that minimizes the cost of the assignment on C. Let w_i^A, w_j^C and w_k^C represents words from language A, B and C. We define seven propositions as Boolean variables between a pair of words w_i^A, w_j^B and w_k^C as follows:

- $e(w_i^A, w_j^B)$ and $e(w_j^B, w_k^C)$ represents edge existence between word pair from language A and B and from language B and C respectively,
- $c(w_i^A, w_k^C)$, $c(w_i^A, w_n^C)$ and $c(w_m^A, w_k^C)$ represents whether the word pair from language A and C is a cognate pair, and
- $s(w_i^A, w_n^C)$ and $s(w_m^A, w_k^C)$ represents whether the word pair from language A and C is a cognate synonym pair

To encode some of the constraints to CNF, we use a resolution approach based on the Boolean algebra rule of $p \rightarrow q \wedge r \Leftrightarrow (\neg p \vee q) \wedge (\neg p \vee r)$. In the framework, we define E_E as a set of word pairs connected by existing edges, E_N as a set of word pairs connected by new edges, D_C as a set of translation pair candidates, D_{Co} as a set of cognate pairs, D_{NCo} as a set of non-cognate pairs, D_{PCo} as a set of pivot words from language B which are connecting the current cognate pair, and D_R as a set of all translation pair results returned by the WPMaxSAT solver.

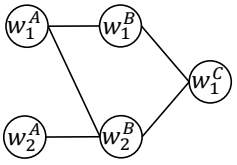
4.3.4 Heuristics to Find Cognate

We define four heuristics to find cognates in the transgraph: cognate pair coexistence probability, missing contribution rate toward cognate pair coex-

istence, polysemy pivot ambiguity rate, and cognate form similarity. Based on our symmetry assumption, when w_i^A and w_k^C in a transgraph share all of their senses through pivot word(s) from language B, they are a potential cognate pair, where the cognate pair coexistence probability equals 1, the missing contribution equals 0 and the polysemy pivot ambiguity rate equals 0. When w_i^A and w_k^C have the same spelling, they are a potential cognate pair, where the cognate form similarity equals 1. Thus, when w_i^A and w_k^C are satisfying the symmetry assumption and also have the same spelling, we take them as the highest potential cognate pair in the transgraph.

Cognate Pair Coexistence Probability

Cognate pairs of language A and C are induced from two input bilingual dictionaries, i.e., $d_{(A,B)}$ and $d_{(B,C)}$. We define two sets of event for $d_{(A,B)}$ (w_i^A and w_j^B) where event w_i^A represents connecting word w_i^A of language A to words of language B represented by edges based on shared meaning between them. Similarly, event w_j^B represents connecting word w_j^B of language B to words of language A. We also define two sets of event for $d_{(B,C)}$ (w_j^B and w_k^C) where event w_j^B represents connecting word w_j^B of language B to words of language C and event w_k^C represents connecting word w_k^C of language C to words of language B. A marginal probability $P(w_i^A)$ is a probability of w_i^A connected to words of language B. A conditional probability $P(w_i^A|w_j^B)$ is a probability of w_i^A connected to w_j^B considering other words of language A that also connected to w_j^B. A joint probability $P(w_i^A, w_j^B)$ is a probability of w_i^A interconnected to w_j^B. For example, in Figure 4.6, $P(w_1^A) = 2/3$, since w_1^A has two connected edges with words of language B out of 3 existing connected edges between words of language A and words of language B. The



	w_1^B	w_2^B	Total
w_1^A	$1/3^*$	$1/3^*$	$2/3^{**}$
w_2^A	0^*	$1/3^*$	$1/3^{**}$
Total	$1/3^{**}$	$2/3^{**}$	1

	w_1^C	Total
w_1^B	$1/2^*$	$1/2^{**}$
w_2^B	$1/2^*$	$1/2^{**}$
Total	1^{**}	1

* Joint Probability $P(w_i^X, w_j^Y)$
** Marginal Probability $P(w_i^X)$

Figure 4.6: Example of Marginal and Joint Probability.

joint probability $P(w_1^A, w_1^B) = 1/3$, since any word from language A and any word from language B are only interconnected with 1 edge out of 3 existing connected edges between words of language A and words of language B.

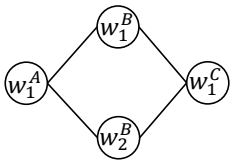
To calculate the possibility of a translation pair candidate $t(w_i^A, w_k^C)$ being a cognate pair $c(w_i^A, w_k^C)$, we calculate $t(w_i^A, w_k^C).H_{coex}$, a cognate coexistence probability of translation pair candidate $t(w_i^A, w_k^C)$. We firstly utilize a chain rule to obtain Equation (4.1) and Equation (4.2) . By multiplying them, we can get Equation (4.3) . Event w_i^A and event w_k^C are independent since they are from a different input bilingual dictionary, thus, $P(w_k^C, w_i^A) = P(w_i^A)P(w_k^C)$ and Equation (4.3) can be rewritten as Equation (4.4) . We use a generative probabilistic process which commonly used in prior work [Déjean et al., 2002, Wu and Wang, 2007, Nakov and Ng, 2012, Richardson et al., 2015] in Equation (4.5) to obtain $P(w_i^A|w_k^C)$ and $P(w_k^C|w_i^A)$. Finally, we can obtain a cognate coexistence probability of translation pair candidate $t(w_i^A, w_k^C)$ as $t(w_i^A, w_k^C).H_{coex} = P(w_i^A, w_k^C)$.

$$P(w_i^A, w_k^C) = P(w_k^C|w_i^A)P(w_i^A) \tag{4.1}$$

$$P(w_k^C, w_i^A) = P(w_i^A|w_k^C)P(w_k^C) \tag{4.2}$$

$$P(w_i^A, w_k^C)P(w_k^C, w_i^A) = P(w_i^A|w_k^C)P(w_k^C|w_i^A)P(w_i^A)P(w_k^C) \tag{4.3}$$

$$P(w_i^A, w_k^C) = P(w_i^A|w_k^C)P(w_k^C|w_i^A) \tag{4.4}$$

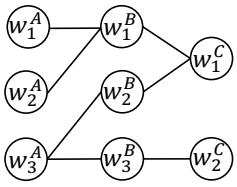


	w_1^B	w_2^B	Total
w_1^A	1/2	1/2	1
Total	1/2	1/2	1

	w_1^C	Total
w_1^B	1/2	1/2
w_2^B	1/2	1/2
Total	1	1

$P(w_1^A, w_1^C) = P(w_1^A|w_1^C)P(w_1^C|w_1^A)$
$= (P(w_1^A|w_1^B)P(w_1^B|w_1^C) + P(w_1^A|w_2^B)P(w_2^B|w_1^C))\ (P(w_1^C|w_1^B)P(w_1^B|w_1^A) + P(w_1^C|w_2^B)P(w_2^B|w_1^A))$
$= (1/2 * 1/2 + 1/2 * 1/2)(1 * 1/2 + 1 * 1/2)$
$= 1$

Figure 4.7: Symmetry Pair Coexistence Probability.



	w_1^B	w_2^B	w_3^B	Total
w_1^A	1/4	0	0	1/4
w_2^A	1/4	0	0	1/4
w_3^A	0	1/4	1/4	2/4
Total	2/4	1/4	1/4	1

	w_1^C	w_2^C	Total
w_1^B	1/3	0	1/3
w_2^B	1/3	0	1/3
w_3^B	0	1/3	1/3
Total	2/3	1/3	1

$P(w_1^A, w_1^C) = P(w_1^A|w_1^C)P(w_1^C|w_1^A)$
$= (P(w_1^A|w_1^B)P(w_1^B|w_1^C))\ (P(w_1^C|w_1^B)P(w_1^B|w_1^A))$
$= (1/2 * 1/2)(1 * 1)$
$= 1/4$

$P(w_3^A, w_1^C) = P(w_3^A|w_1^C)P(w_1^C|w_3^A)$
$= (P(w_3^A|w_1^B)P(w_1^B|w_1^C))\ (P(w_1^C|w_1^B)P(w_1^B|w_3^A))$
$= (1 * 1/2)(1 * 1/2)$
$= 1/4$

Figure 4.8: Equal Treatment of Polysemy in Pivot/Non-Pivot Word.

$$P(w_i^A|w_k^C) = \sum_{j=0} P(w_i^A|w_j^B)P(w_j^B|w_k^C) \tag{4.5}$$

When w_i^A and w_k^C in a transgraph share all of their senses through pivot word(s) from language B and none of the pivot words are ambiguous, the cognate pair coexistence probability equals 1 as shown in Figure 4.7. The algorithm to calculate the probability of the translation pair candidates coexisting as a cognate is shown in Algorithm 2 line number 19. The coexistence probability is very important in differentiating cognates from non cognates, but, it is poor at avoiding polysemy in pivot words. This is because it treats polysemy in the pivot words and polysemy in the non-pivot words equally. In reality, however, polysemy in pivot words negatively impacts the quality

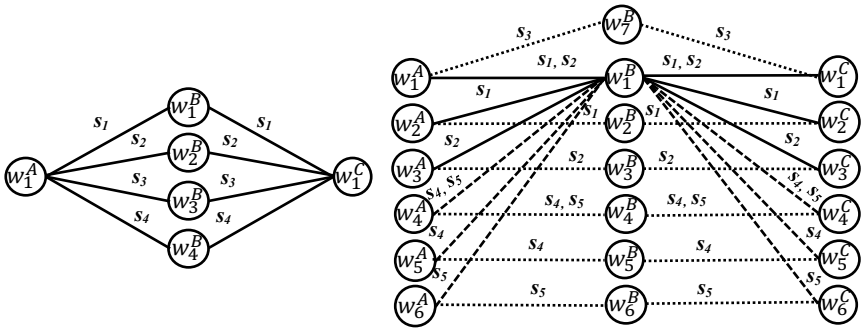
of bilingual dictionary induction rather than polysemy in non-pivot words. A case with high polysemy in pivot words and low polysemy in non-pivot words and a case with low polysemy in pivot words and high polysemy in non-pivot words where the two cases have equal rates of polysemy, will yield same probability as shown in Figure 4.8. Therefore, we introduce a special heuristic to tackle this problem, i.e., polysemy pivot ambiguity rate.

Missing Contribution Rate Toward Cognate Pair Coexistence

Inspired by the Shapley Value [Shapley, 1953], a solution concept in cooperative game theory, we calculate missing contribution rate toward cognate pair coexistence probability by calculating coexistence probability of supposed cognate pair (also considering missing edges as existing) minus the coexistence probability of the pair from existing connectivity only. When w_i^A and w_k^C in a transgraph share all of their senses through pivot word(s) from language B (no missing senses), the missing contribution equals 0. The lower is the missing contribution toward coexistence probability of a translation pair candidate, the more likely is the translation pair candidate of being a cognate. The calculation of missing contribution rate of w_1^A and w_1^C pair, i.e., $t(w_i^A, w_k^C).H_{missCont}$ is shown in Algorithm 2 line number 20.

Polysemy Pivot Ambiguity Rate

To model the effect of polysemy in the pivot language on precision, for the sake of simplicity, we ignore synonym words within the same language. Polysemy in non-pivot languages have no negative effect on precision. In Figure 4.9a, even though the non-pivot words are connected by four pivot



(a) Polysemy in non-pivot language (b) Polysemy in pivot language

Figure 4.9: Polysemy in pivot and non-pivot language.

words representing four senses/meanings, the transgraph only has one translation pair candidate (w_1^A-w_1^C) and so the precision is 100%.

However, polysemy in pivot language negatively impacts precision. Figure 4.9b shows that non-pivot word w_1^A and w_1^C are cognates and share the same meanings (s_1,s_2,s_3), but pivot word w_1^B which has four meanings (s_1,s_2,s_4,s_5) only shares a part of the meanings (s_1,s_2) with the non-pivot words. The solid edges have part or all shared meanings (s_1, s_2) between the non-pivot words (w_1^A, w_1^C) and the pivot word w_1^B. The dashed edges express part or all unshared meanings (s_4, s_5) between the non-pivot words (w_1^A, w_1^C) and the pivot word w_1^B. To investigate the effect of pivot word w_1^B on the overall precision, we extract only translation pair candidates from the connected edges. The precision (38.89%) is affected negatively as there are 22 wrong translations because of the polysemy in pivot language (w_1^B) in the transgraph.

We formalize the effect of polysemy in pivot language on precision with the following formulation where n is the number of shared meanings between

pivot word and non-pivot words and m is the number of pivot meaning(s) that are not shared with non-pivot words. The number of correct translations contributed by the solid edges and the number of correct translations contributed by the dashed edges can be calculated by Equation (4.6). The precision of the translation result is calculated by Equation (4.7).

$$CorrectTrans(n) = 2\sum_{i=1}^{n}\sum_{j=1}^{i}\binom{n}{i}\binom{i}{j} - \sum_{i=1}^{n}\binom{n}{i} \tag{4.6}$$

$$Precision(n,m) = \frac{CorrectTrans(n) + CorrectTrans(m)}{\left[\sum_{i=1}^{n}\binom{n}{i} + \sum_{i=1}^{m}\binom{m}{i}\right]^2} \tag{4.7}$$

We predict the effect of shared meanings between pivot word and non-pivot words by simulating ten sets of transgraphs with n (the number of shared meanings between pivot word and non-pivot words) values ranging from 1 to 10 where, in each set, m (the number of pivot meaning(s) that not shared with non-pivot words) ranges from 0 to n in Figure 4.10. In this experiment, non-pivot languages and pivot language are closely-related languages (w_1^A, w_1^B, and w_1^C are cognates) when there is no pivot meaning that not shared with non-pivot words ($m = 0$). This result shows that the greater the number of shared senses/meanings (represented by n) between pivot and non-pivot words there are, the lower the precision is. Nevertheless, the polysemy in the pivot language has the least negative effect on the precision when the pivot language and non-pivot languages are closely-related where the number of unshared pivot senses (represented by m) equals 0. The negative effect increases as the value of m increases.

Polysemy in pivot words negatively impacts the precision of the translation result, unlike that in non-pivot words. Since we do not have any information about the senses from the input dictionaries, it is difficult to avoid the

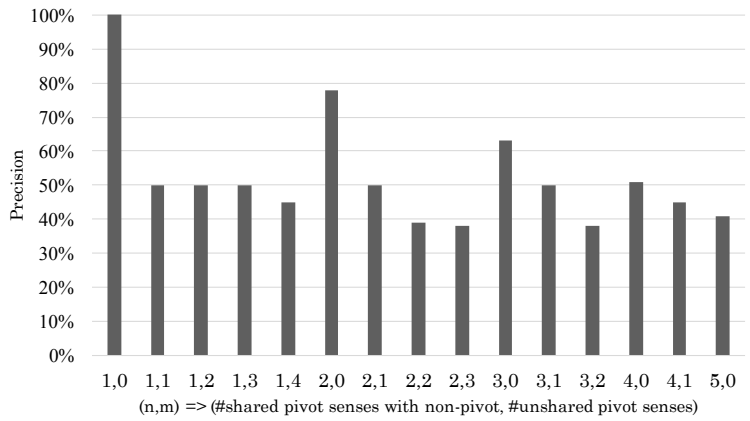


Figure 4.10: Prediction model of precision on polysemy in pivot language.

negative effect of the polysemous pivot word. To predict a probability of w_i^A and w_k^C to be a cognate pair via pivot word w_j^B which share common senses, we assume the worst case scenario where the number of senses belonging to pivot word w_j^B equals the maximum number of connected edges to w_i^A or w_k^C. If the maximum number of indegree or outdegree of the polysemy pivot is n, there are $2^n - 1$ possible combination of shared senses for every paths via pivot word w_j^B in order for the translation pair candidates to be a cognate pair $c(w_i^A, w_k^C)$ out of all $(2^n - 1)^2$ combinations. In Figure 4.4b, the possible combination of shared senses between w_1^A and w_1^C or between w_1^A and w_2^C are: [s_1, s_2, s_1 & s_2]. To calculate the probability of the pair w_i^A and w_k^C being a cognate considering polysemy in the pivot words, we calculate $t(w_i^A, w_k^C).P_{sharedSenses}$, the product of the probabilities of shared senses between the pair for every existing path as shown in Algorithm 2 line number 10. The polysemy pivot ambiguity rate is given by $t(w_i^A, w_k^C).H_{polysemy} = 1 - t(w_i^A, w_k^C).P_{sharedSenses}$ as shown in Equation (4.8)

and Algorithm 2 line number 21.

$$t(w_i^A, w_k^C).H_{polysemy} = 1 - \prod \left((2^n - 1)/(2^n - 1)^2 \right) = 1 - \prod \left(1/(2^n - 1) \right) \quad (4.8)$$

The lower the polysemy pivot ambiguity rate is, the more likely it is that the pair form a cognate and share exact senses. When there is only one path between w_i^A and w_k^C and there is only one indegree and one outdegree of the pivot word w_j^B, the polysemy pivot ambiguity rate equals 0.

Cognate Form Similarity

Because the symmetry assumption can sometimes fail to select a cognate correctly when it gives the same cost for multiple translation pair candidates, the cognate form similarity heuristic will contribute to selecting the cognate. We calculate cognate form similarity using Longest Common Subsequent Ratio (LCSR) ranging from 0 (0% form-similarity) to 1 (100% form-similarity) following [Melamed, 1995] as shown in Equation (4.9) and Algorithm 2 line number 22 where $LCS(w_i^A, w_k^C)$ is the longest common subsequence of w_i^A and w_k^C; $|x|$ is the length of x; and $max(|w_i^A|, |w_k^C|)$ returns the longest length. However, the maximum cost contributed from the form dissimilarity is set at 1/100 of the maximum cost contributed by one symmetry assumption heuristic as shown in Algorithm 3 line number 24 to ensure that the cognate form similarity heuristic will have only a supporting role in helping the main symmetry assumption heuristics.

$$LCSR(w_i^A, w_k^C) = \frac{|LCS(w_i^A, w_k^C)|}{max(|w_i^A|, |w_k^C|)} \quad (4.9)$$

$$t(w_i^A, w_k^C).H_{formSim} = LCSR(w_i^A, w_k^C) \quad (4.10)$$

ALGORITHM 2: Cognate Pair Probability Calculation

Input: Translation pair candidate $t(w_i^A, w_k^C)$;

Output: Translation pair candidate $t(w_i^A, w_k^C)$ with cognate pair probabilites information

$P(w_i^A, w_k^C) = 0$; $P(w_k^C, w_i^A) = 0$; $P_{missing}(w_i^A, w_k^C) = 0$; $P_{missing}(w_k^C, w_i^A) = 0$;
for *each path in* $t(w_i^A, w_k^C).Paths$ **do**
  $P(w_i^A|w_j^B) = 0$; $P(w_j^B|w_k^C) = 0$; $P(w_k^C|w_j^B) = 0$; $P(w_j^B|w_i^A) = 0$;
  `/* Conditional Probability direction: A-C */`
  for *each inEdge in* $w_j^B.inEdges$ *from language A* **do**
    $w_j^B.indegreeFromA \mathrel{+}= 1/inEdge.Prob$;
  for *each inEdge in* $w_k^C.inEdges$ *from language B* **do**
    $w_k^C.indegreeFromB \mathrel{+}= 1/inEdge.Prob$;
  $P(w_i^A|w_j^B) = 1 \,/\, w_j^B.indegreeFromA$; $P(w_j^B|w_k^C) = 1 \,/\, w_k^C.indegreeFromB$;
  `/* Conditional Probability direction: C-A */`
  for *each inEdge in* $w_j^B.inEdges$ *from language C* **do**
    $w_j^B.indegreeFromC \mathrel{+}= 1/inEdge.Prob$;
  for *each inEdge in* $w_i^A.inEdges$ *from language B* **do**
    $w_i^A.indegreeFromB \mathrel{+}= 1/inEdge.Prob$;
  $P(w_k^C|w_j^B) = 1 \,/\, w_j^B.indegreeFromC$; $P(w_j^B|w_i^A) = 1 \,/\, w_i^A.indegreeFromB$;
  $t(w_i^A, w_k^C).P_{sharedSenses} \mathrel{*}= 1/(2^{max(w_j^B.indegreeFromA, w_j^B.indegreeFromC)} - 1)$;
  if *missing edge exist in path* **then**
    $P_{missing}(w_i^A|w_k^C) \mathrel{+}= P(w_i^A|w_j^B)P(w_j^B|w_k^C)$;
    $P_{missing}(w_k^C|w_i^A) \mathrel{+}= P(w_k^C|w_j^B)P(w_j^B|w_i^A)$;
  else
    $P(w_i^A|w_k^C) \mathrel{+}= P(w_i^A|w_j^B)P(w_j^B|w_k^C)$;
    $P(w_k^C|w_i^A) \mathrel{+}= P(w_k^C|w_j^B)P(w_j^B|w_i^A)$;
  end
end
$t(w_i^A, w_k^C).H_{coex} = P(w_i^A|w_k^C)P(w_k^C|w_i^A)$;
$t(w_i^A, w_k^C).H_{missCont} = (P(w_i^A|w_k^C) + P_{missing}(w_i^A|w_k^C))(P(w_k^C|w_i^A) + P_{missing}(w_k^C|w_i^A)) - (P(w_i^A|w_k^C)P(w_k^C|w_i^A))$;
$t(w_i^A, w_k^C).H_{polysemy} = 1 - t(w_i^A, w_k^C).P_{sharedSenses}$;
$t(w_i^A, w_k^C).H_{formSim} = LCSR(w_i^A, w_k^C)$;
return $t(w_i^A, w_k^C)$;

ALGORITHM 3: Cognate and Cognate Synonym Extraction

Input: *transgraphs*, *maxCycle*, *threshold*, *HSelections*;
Output: D_{Co} /* list of cognate pair results */

for *each transgraph in* `calculateEdgeCost(transgraphs)` **do**
  /* Extract the most probable cognate pair and cognate synonym pair with total cost of violating constraints below the threshold iteratively */
  $CNF_{cognate} \leftarrow constructCNF_{cognate}(transgraph.D_C)$; /* following Eq.(11) */
  while $cognatePair \leftarrow SATSolver.solve(CNF_{cognate})$ **do**
    if $cognatePair.totalCost < cognateThreshold$ **then**
      $D_{Co} \leftarrow cognatePair$; $CNF_{cognate}.update()$;
    end
  end
  $CNF_{cognateSynonym} \leftarrow constructCNF_{cognateSynonym}(transgraph.D_C)$; /* following Eq.(12) */
  while $cognateSynonymPair \leftarrow SATSolver.solve(CNF_{cognateSynonym})$ **do**
    if $cognateSynonymPair.totalCost < cognateSynonymThreshold$ **then**
      $D_{Co} \leftarrow cognateSynonymPair$; $CNF_{cognateSynonym}.update()$;
    end
  end
end
return D_{Co};
Function `calculateEdgeCost(transgraphs)`
  for *each transgraph in transgraphs* **do**
    $transgraph.D_C \leftarrow generateCandidates(transgraph)$; /* generate trans. pair cand. */
    for *each* $t(w_i^A, w_k^C)$ *in* $transgraph.D_C$ **do**
      calculateCognatePairProb($t(w_i^A, w_k^C)$); /* using Algorithm 1 */
      /* Cost of adding new edges are calculated from user selected heuristics */
      if *HSelections.coex is TRUE* **then** $t(w_i^A, w_k^C).EdgeCost \mathrel{+}= 1 - t(w_i^A, w_k^C).H_{coex}$;
      if *HSelections.missCont is TRUE* **then** $t(w_i^A, w_k^C).EdgeCost \mathrel{+}= t(w_i^A, w_k^C).H_{missCont}$;
      if *HSelections.polysemy is TRUE* **then** $t(w_i^A, w_k^C).EdgeCost \mathrel{+}= t(w_i^A, w_k^C).H_{polysemy}$;
      if *HSelections.formSim is TRUE* **then** $t(w_i^A, w_k^C).EdgeCost \mathrel{+}= \left(1 - t(w_i^A, w_k^C).H_{formSim}\right)/100$;
      for *each* $w_i^A.outEdges$ **do**
        if $e(w_j^B, w_k^C)$ *is not exist* **then** $t(w_i^A, w_k^C).e(w_j^B, w_k^C).Cost = t(w_i^A, w_k^C).EdgeCost$;
      end
      for *each* $w_i^C.inEdges$ **do**
        if $e(w_i^A, w_j^B)$ *is exist* **then** $t(w_i^A, w_k^C).e(w_i^A, w_j^B).Cost = t(w_i^A, w_k^C).EdgeCost$;
      end
    end
  end
  if *maxCycle is not reached* **then**
    $transgraphs \leftarrow addNewEdges()$; /* add new edges to transgraphs for the next cycle */
    `calculateEdgeCost(transgraphs)`;
  end
  return *transgraphs*

Table 4.1: Constraints for Cognates and Cognate Synonyms Extraction

ID	CNF Formula
	Edge Existence:
φ_1^{∞}	$\Big(\bigwedge_{(w_i^A,w_j^B)\in E_E}\big(e(w_i^A,w_j^B),\infty\big)\Big)\wedge\Big(\bigwedge_{(w_j^B,w_k^C)\in E_E}\big(e(w_j^B,w_k^C),\infty\big)\Big)$
	Edge Non-Existence:
φ_2^{+}	$\Big(\bigwedge_{(w_i^A,w_j^B)\in E_N}\big(\neg e(w_i^A,w_j^B),\omega(w_i^A,w_j^B)\big)\Big)\wedge\Big(\bigwedge_{(w_j^B,w_k^C)\in E_N}\big(\neg e(w_j^B,w_k^C),\omega(w_j^B,w_k^C)\big)\Big)$
	Symmetry:
φ_3^{∞}	$\Big(\bigwedge_{\substack{(w_i^A,w_j^B)\in E_E\cup E_N\\(w_i^A,w_k^C)\in D_C}}\big((\neg c(w_i^A,w_k^C)\vee e(w_i^A,w_j^B)),\infty\big)\Big)\wedge\Big(\bigwedge_{\substack{(w_j^B,w_k^C)\in E_E\cup E_N\\(w_i^A,w_k^C)\in D_C}}\big((\neg c(w_i^A,w_k^C)\vee e(w_j^B,w_k^C)),\infty\big)\Big)$
	Uniqueness:
φ_4^{∞}	$\Big(\bigwedge_{\substack{k\neq n\\(w_i^A,w_k^C)\in D_C\\(w_i^A,w_n^C)\in D_C}}\big((\neg c(w_i^A,w_k^C)\vee\neg c(w_i^A,w_n^C)),\infty\big)\Big)\wedge\Big(\bigwedge_{\substack{i\neq m\\(w_i^A,w_k^C)\in D_C\\(w_i^A,w_n^C)\in D_C}}\big((\neg c(w_i^A,w_k^C)\vee\neg c(w_m^A,w_k^C)),\infty\big)\Big)$
	Extracting at Least One Cognate:
φ_5^{∞}	$\Big(\Big(\bigvee_{(w_i^A,w_k^C)\notin D_R}c(w_i^A,w_k^C)\Big),\infty\Big)$
	Encoding Cognate:
φ_6^{∞}	$\bigwedge_{(w_i^A,w_k^C)\in D_{Co}}\big(c(w_i^A,w_k^C),\infty\big)$
	Encoding Non-Cognate:
φ_7^{∞}	$\bigwedge_{(w_i^A,w_k^C)\in D_{NCo}}\big(\neg c(w_i^A,w_k^C),\infty\big)$
	Cognate Synonym:
φ_8^{∞}	$\Big(\bigwedge_{\substack{k\neq n\\(w_i^A,w_k^C)\in D_{Co}\\(w_i^A,w_n^C)\notin D_R}}\big((\neg s(w_i^A,w_n^C)\vee c(w_i^A,w_k^C)),\infty\big)\wedge\Big(\bigwedge_{w_j^B\in D_{PCo}}\big((\neg s(w_i^A,w_n^C)\vee e(w_j^B,w_n^C)),\infty\big)\Big)\Big)$
	$\wedge\Big(\bigwedge_{\substack{i\neq m\\(w_m^A,w_k^C)\in D_{Co}\\(w_i^A,w_k^C)\notin D_R}}\big((\neg s(w_m^A,w_k^C)\vee c(w_i^A,w_k^C)),\infty\big)\wedge\Big(\bigwedge_{w_j^B\in D_{PCo}}\big((\neg s(w_m^A,w_k^C)\vee e(w_m^A,w_j^B)),\infty\big)\Big)\Big)$
	Extracting at Least One Cognate Synonym:
φ_9^{∞}	$\Big(\Big(\bigvee_{(w_i^A,w_k^C)\notin D_R}s(w_i^A,w_k^C)\Big),\infty\Big)$

4.3.5 Constraints Extension

We extend the one-to-one approach constraints by adding several new constraints to the constraint sets to find cognates and cognate synonyms. All constraints are listed in Table 4.1.

Edge Existence

An edge exists in the transgraph between words that share their meaning(s) based on input dictionaries. The existing edges in the transgraph are encoded as TRUE, i.e., $e(w_i^A, w_j^B)$ and $e(w_j^B, w_k^C)$ in the CNF formula which is represented as hard constraint φ_1^∞.

Edge Non-Existence

An edge does not exist in the transgraph between words that do not share their meaning(s) based on input dictionaries. We formalize the non-existence of edge in the transgraph by encoding the negation of the literal edge existence, i.e., $\neg e(w_i^A, w_j^B)$ and $\neg e(w_j^B, w_k^C)$ in the CNF formula which is represented as soft constraint φ_2^+.

Symmetry

Cognate share all of their senses / meanings and symmetrically connected via pivot language B. We convert $c(w_i^A, w_k^C) \rightarrow e(w_i^A, w_1^B) \wedge e(w_i^A, w_2^B) \wedge \ldots \wedge e(w_1^B, w_k^C) \wedge e(w_2^B, w_k^C) \wedge \ldots$ into $(\neg c(w_i^A, w_k^C) \vee e(w_i^A, w_1^B)) \wedge (\neg c(w_i^A, w_k^C) \vee e(w_i^A, w_2^B)) \wedge \ldots . \wedge (\neg c(w_i^A, w_k^C) \vee e(w_1^B, w_k^C)) \wedge (\neg c(w_i^A, w_k^C) \vee e(w_1^B, w_k^C)) \wedge \ldots$ It is encoded as hard constraint φ_3^∞. Unfortunately, a problem arises with low-resource languages where the input dictionaries have no sense information and many senses are expected to be missed due to the small size of the dictionaries. To solve this problem, we add new edges so that cognate pairs share all of the meanings by violating the edge non-existence soft constraint φ_2^+ and paying a cost determined from user-selected heuristics (cognate pair

coexistence probability, missing contribution rate toward the cognate pair coexistence probability, polysemy pivot ambiguity rate, and cognate form similarity). In other words, we assume the edges exist. The higher the cognate pair coexistence probability and the lower the missing contribution rate toward the cognate pair coexistence probability and the lower the polysemy pivot ambiguity rate and the higher the cognate form similarity, the more likely it is that the pair form a cognate, thus, the lower is the cost of adding any new edge to it, i.e., the new edge weight. The new edges in the transgraph is encoded as FALSE (NOT exist), i.e., $\neg e(w_i^A, w_j^B)$ or $\neg e(w_j^B, w_k^C)$ in the CNF formula and depicted as dashed edges in the transgraph. The weight of the new edge from non-pivot word w_i^A to pivot word w_j^B is defined as $\omega(w_i^A, w_j^B)$ and the weight of a new edge from pivot word w_j^B to non-pivot word w_k^C is defined as $\omega(w_j^B, w_k^C)$. Both of $\omega(w_i^A, w_j^B)$ and $\omega(w_j^B, w_k^C)$ values equal $t(w_i^A, w_k^C).H_{coex} + t(w_i^A, w_k^C).H_{missCont} + t(w_i^A, w_k^C).H_{polysemy} + t(w_i^A, w_k^C).H_{formSim}$ as shown in Algorithm 3 line number 21-24.

Uniqueness

The first step of our strategy in obtaining many-to-many translation pair with good quality is to extract a list of cognates in the transgraph. The uniqueness constraint ensures that only one-to-one cognates which share all of their meanings will be considered as translation pairs. In other words, a word in language A can only be a cognate with just one word from language C. This is encoded as hard constraint φ_4^{∞}.

Extracting at Least One Cognate

Since the framework communicates with WPMaxSAT solver iteratively as shown in Algorithm 3 line number 2-7, hard constraint φ_5^∞ ensures that at least one of the $c(w_i^A, w_k^C)$ variables must be evaluated as TRUE. Consequently, each iteration yields one most probable cognate pair and stores it in D_{Co} and also in D_R as a translation pair result. This clause is a disjunction of all $c(w_i^A, w_k^C)$ variables.

Encoding Cognate

We exclude previously selected translation pairs, which are stored in D_{Co} from the following list of cognate pair candidates by encoding them as TRUE, i.e., $c(w_i^A, w_k^C)$ which is encoded as hard constraint φ_6^∞, and excluding them from φ_5^∞.

Encoding Non-Cognate

Once we get list of all cognate pairs, stored in D_{Co}, the remaining translation pair candidates are stored in D_{NCo} and encoded as FALSE, i.e., $\neg c(w_i^A, w_k^C)$ in the CNF formula, which is represented as hard constraint φ_7^∞.

Cognate Synonym

We can further identify cognate synonyms to improve the quantity of the translation results. For each cognate pair $c(w_i^A, w_k^C)$ stored in D_{Co}, we can

find cognate synonym pairs $s(w_i^A, w_n^C)$ and $s(w_m^A, w_k^C)$ by extracting synonyms of w_k^C and w_i^A respectively. We assume that synonymous words are connected to common pivot words. We can add new edges by paying cost of violating soft-constraint φ_2^+ with a weight different from that used when identifying cognate pairs in the first step. In this second step, the weight is calculated based on cognate synonym probability $P_{cognateSyn}$ for both $w_n^C - w_k^C$ and $w_m^A - w_i^A$ based on percentage of shared connectivity with the pivot words. The weight, i.e., $1 - P_{cognateSyn}$ is distributed evenly to each new edges. We convert $s(w_i^A, w_n^C) \rightarrow c(w_i^A, w_k^C) \wedge e(w_1^B, w_n^C) \wedge e(w_2^B, w_n^C) \wedge ...$ into $(\neg s(w_i^A, w_n^C) \vee c(w_i^A, w_k^C)) \wedge (\neg s(w_i^A, w_n^C) \vee e(w_1^B, w_n^C)) \wedge (\neg s(w_i^A, w_n^C) \vee e(w_2^B, w_n^C)) \wedge ....$ With the same rule, we convert $s(w_m^A, w_k^C) \rightarrow c(w_i^A, w_k^C) \wedge e(w_m^A, w_1^B) \wedge e(w_m^A, w_2^B) \wedge ...$ into $(\neg s(w_m^A, w_k^C) \vee c(w_i^A, w_k^C)) \wedge (\neg s(w_m^A, w_k^C) \vee e(w_m^A, w_1^B)) \wedge (\neg s(w_m^A, w_k^C) \vee e(w_m^A, w_2^B)) \wedge ....$ It is encoded as hard constraint φ_8^∞. In Figure 4.3, $s(w_1^A, w_3^C).P_{cognateSyn} = 1$, $s(w_1^A, w_4^C).P_{cognateSyn} = 0.67$, and $s(w_1^A, w_1^C).P_{cognateSyn} = 0.33$. Another example, in Figure 4.4a, if cognate pair $c(w_1^A, w_1^C)$ is identified, we need to identify cognate synonym probability of w_1^A (no candidate exist) and w_1^C (candidate: w_2^C). Based on the rate of shared connectivity with pivot word(s), $s(w_1^A, w_2^C).P_{cognateSyn} = 2/2$ and in Figure 4.4b with the same way we can get $s(w_1^A, w_2^C).P_{cognateSyn} = 1/2$.

Extracting at Least One Cognate Synonym

In the second step, i.e., finding cognate synonyms, the framework also communicates with the WPMaxSAT solver iteratively as shown in Algorithm 3 line number 8-13, and hard constraint φ_9^∞ ensures that at least one of the $s(w_i^A, w_n^C)$ variables or $s(w_m^A, w_k^C)$ variables must be evaluated as TRUE. Consequently, each iteration yields one most probable cognate synonym

Table 4.2: Variation of Constraint-based Bilingual Dictionary Induction

Cycle	$CNF_{cognate}$	$CNF_{cognate} + CNF_{cognateSynonym}$	CNF_{M-M}
1	H1[1], H2, H3, H4, H12, ...	H1, H2, H3, H4, H12, ...	H1[2]
>1	H1, H2, H3, H4, H12, ...	H1, H2, H3, H4, H12, ...	H1[3]

[1] *Identical to one-to-one approach [Wushouer et al., 2015] and* Ω_1 *in our prior work [Nasution et al., 2016]*

[2] *Identical to* Ω_2 *in our prior work [Nasution et al., 2016]*

[3] *For 2-cycle, identical to* Ω_3 *in our prior work [Nasution et al., 2016]*

pair and store it in D_R as a translation pair result. This clause is a disjunction of all $s(w_i^A, w_k^C)$ variables.

4.3.6 Framework Generalization

We define two main CNF formulas; one for recognizing cognate pairs, i.e., $CNF_{cognate}$ as shown in Equation (4.11) and one for recognizing cognate synonym pairs, i.e., $CNF_{cognateSynonym}$ as shown in Equation (4.12). We also define another CNF formula, i.e., CNF_{M-M} as shown in Equation (4.13) which extract many-to-many translation pairs by ignoring uniqueness constraint of the one-to-one approach [Nasution et al., 2016]. Three constraints are shared by the CNF formulas: φ_1^∞, φ_2^+ and φ_6^∞.

$$CNF_{cognate} = \varphi_1^\infty \wedge \varphi_2^+ \wedge \varphi_3^\infty \wedge \varphi_4^\infty \wedge \varphi_5^\infty \wedge \varphi_6^\infty \tag{4.11}$$

$$CNF_{cognateSynonym} = \varphi_1^\infty \wedge \varphi_2^+ \wedge \varphi_6^\infty \wedge \varphi_7^\infty \wedge \varphi_8^\infty \wedge \varphi_9^\infty \tag{4.12}$$

$$CNF_{M-M} = \varphi_1^\infty \wedge \varphi_2^+ \wedge \varphi_3^\infty \wedge \varphi_5^\infty \wedge \varphi_6^\infty \tag{4.13}$$

As shown in Table 4.2, various constraint-based bilingual dictionary induc-

tion methods can be constructed to suit different situations and purposes by using a cognate recognition ($CNF_{cognate}$) or a cognate & cognate synonym recognition ($CNF_{cognate} + CNF_{cognateSynonym}$) methods with a choice of n-cycle symmetry assumption, and with a series of individual and combined heuristics to be chosen. We can also define many-to-many translation pair extraction method in our previous work using CNF_{M-M}. Thus, we define our methods using Backus Normal Form as follow:

$\langle situatedMethod \rangle ::= \langle cycle \rangle \text{“:”} \langle method \rangle \text{“:”} \langle heuristic \rangle$
$\langle cycle \rangle ::= \text{“1”}|\text{“2”}|\text{“3”}|\text{“4”}|\text{“5”}|\text{“6”}|\text{“7”}|\text{“8”}|\text{“9”}$
$\langle method \rangle ::= \text{“}C\text{”}|\text{“}S\text{”}|\text{“}M\text{”}$
$\langle heuristic \rangle ::= \text{“}H1\text{”}|\text{“}H2\text{”}|\text{“}H3\text{”}|\text{“}H4\text{”}|\text{“}H12\text{”}|\text{“}H13\text{”}|\text{“}H14\text{”}|\text{“}H23\text{”}|\text{“}H24\text{”}|$
$\text{”}H123\text{”}|\text{”}H124\text{”}|\text{”}H234\text{”}$

- *cycle*: symmetry assumption cycle where cycle ≥ 1.
- *method*: C as a cognate recognition ($CNF_{cognate}$) or S as a cognate & cognate synonym recognition ($CNF_{cognate} + CNF_{cognateSynonym}$) or M as a many-to-many approach (Ω_2 & Ω_3) in our previous work [Nasution et al., 2016].
- *heuristic*: an individual or combined heuristics where H1234 means a combination of heuristic 1 (cognate pair coexistence probability), heuristic 2 (missing contribution rate toward cognate pair coexistence), heuristic 3 (polysemy pivot ambiguity rate), and heuristic 4 (cognate form similarity).

A combination of cognate only ($CNF_{cognate}$) method with 1-cycle symmetry assumption and heuristic 1 is defined as 1:C:H1, yielding an identical method with one-to-one approach [Wushouer et al., 2015] and Ω_1 in

our prior work [Nasution et al., 2016]. A combination of cognate only (CNF_{M-M}) method with heuristic 1 and 1-cycle symmetry assumption is defined as 1:M:H1, which is identical with Ω_2 and for 2-cycle symmetry assumption is defined as 2:M:H1, which is identical with Ω_3 in our prior work [Nasution et al., 2016].

4.4 Experiment

To evaluate our result, we calculate precision, recall and the harmonic mean of precision and recall using the traditional F-measure or balanced F-score [Rijsbergen, 1979]. In each iteration, WPMaxSAT solver returns the optimal translation pair result with minimum total cost (incurred by violating some soft constraints). Translation pair result with total cost above the threshold are not considered. For the methods equivalent with our prior work [Nasution et al., 2016] which are 1:C:H1, 1:M:H1, and 2:M:H1, we do not set any threshold. We try to analyze the impact of the threshold and the heuristics on the precision, recall and F-score. For this purpose, we need to have a Gold Standard, so that for each experiment, we can iterate threshold from 0 to the highest cost of constraint violation cost with 0.01 interval and try every combination of heuristics as input to Algorithm 3 (as *threshold* & *HSelections*) while observing the resulting precision, recall or F-score after evaluation against the gold standard. In this research, we choose the result with the highest F-score. We want to analyze the algorithm so that our generalized constraint approach can be applied to other datasets for various languages. We conduct experiments with 6 methods constructed from our generalized constraint approach in which 3 of them yielding one-to-

one translation pairs (1-1), i.e., Cognates recognition with all combination of heuristic and 1-cycle symmetry assumption (1:C:$\langle heuristic \rangle$), 2-cycles symmetry assumption (2:C:$\langle heuristic \rangle$), and 3-cycles symmetry assumption (3:C:$\langle heuristic \rangle$), and the rest yielding many-to-many translation pairs (M-M), i.e., Cognate and Cognate Synonyms recognition with all combination of heuristic and 1-cycle symmetry assumption (1:S:$\langle heuristic \rangle$), 2-cycles symmetry assumption (2:S:$\langle heuristic \rangle$), and 3-cycles symmetry assumption (3:S:$\langle heuristic \rangle$). As baselines, we use three methods from our previous work where H1 is the sole heuristic used [Nasution et al., 2016], i.e., one-to-one translation pair extraction (Ω_1) which is defined as 1:C:H1, many-to-many translation pair extraction from connected existing edges (Ω_2) which is defined as 1:M:H1, and many-to-many translation pair extraction from connected existing and new edges (Ω_3) which is defined as 2:M:H1. We also use the inverse consultation method (IC) and translation pairs generated from cartesian product of input dictionaries (CP) as baselines.

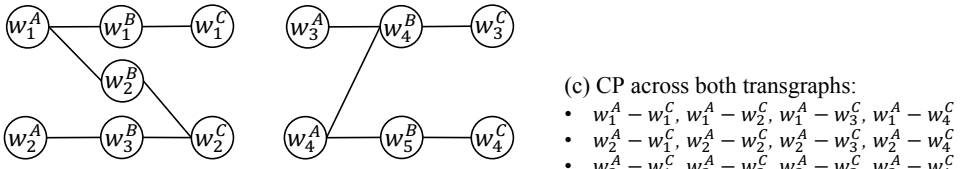
4.4.1 Experimental Settings

We have four case studies; one of the closely related low-resource languages of Austronesian language family and three of high-resource Indo-European languages. The language similarities shown in Table 4.3 were computed using ASJP. We generate translation pairs from cartesian product within and across transgraph to be used in the evaluation as shown in Figure 4.11.

We selected Indonesian ethnic languages Minangkabau (min) and Riau Mainland Malay (zlm) with Indonesian language (ind) as the pivot for our first case study CS1 (min-ind-zlm) since the three languages belong to the

Table 4.3: Language Similarity of Input Dictionaries

Language Pair	Language Similarity
min-ind, zlm-ind, min-zlm	69.14%, 87.70%, 61.66%
deu-eng, nld-eng, deu-nld	31.38%, 39.27% 51.17%
spa-eng, por-eng, spa-por	6.66%, 3.79%, 32.04%
deu-eng, ita-eng, deu-ita	31.38%, 9.75%, 13.64%



(a) CP within transgraph 1:
- $w_1^A - w_1^C$, $w_1^A - w_2^C$
- $w_2^A - w_1^C$, $w_2^A - w_2^C$

(b) CP within transgraph 2:
- $w_3^A - w_3^C$, $w_3^A - w_4^C$
- $w_4^A - w_3^C$, $w_4^A - w_4^C$

(c) CP across both transgraphs:
- $w_1^A - w_1^C$, $w_1^A - w_2^C$, $w_1^A - w_3^C$, $w_1^A - w_4^C$
- $w_2^A - w_1^C$, $w_2^A - w_2^C$, $w_2^A - w_3^C$, $w_2^A - w_4^C$
- $w_3^A - w_1^C$, $w_3^A - w_2^C$, $w_3^A - w_3^C$, $w_3^A - w_4^C$
- $w_4^A - w_1^C$, $w_4^A - w_2^C$, $w_4^A - w_3^C$, $w_4^A - w_4^C$

Figure 4.11: Example of Extracting Translation Pair Candidates from Cartesian Product (CP).

densest cluster in Figure 3.5. Even though Malaysian Malay (zlm) is not part of Indonesian ethnic languages, but it is very similar with Riau Mainland Malay. In fact, Riau Mainland Malay is one of Malaysian Malay dialects [Simons and (eds.), 2017]. Since there is no available machine readable dictionary of Indonesian to Riau Mainland Malay, we used the available machine readable dictionary of Indonesian to Malaysian Malay (zlm) for case study min-ind-zlm. A trilingual Indonesian, Malaysian Malay and Riau Mainland Malay speaker thoroughly cleansed the dictionary by deleting or editing Malaysian Malay words that are not present in the Riau Mainland Malay language. We generate full-matching translation pairs (cartesian product within transgraph from input dictionaries), verified by the Minangkabau-Malay bilingual speaker via crowdsourcing and took them as the gold standard for calculating precision and recall. The Proto-Indo-

Table 4.4: Dictionaries for Evaluation

Source	Number of Translation		
Freedict	deu-nld ∪ nld-deu = 35,962	spa-por = 333	deu-ita ∪ ita-deu = 6,152
Panlex	deu-nld = 405,076	spa-por = 343,665	deu-ita = 475,461
Google Translate*	deu-nld ∪ nld-deu = 1,924	spa-por ∪ por-spa = 1,338	deu-ita ∪ ita-deu = 1,790
TOTAL	deu-nld = 406,370	spa-por = 344,126	deu-ita = 476,172

* *Translating all headwords from CP within the transgraphs.*

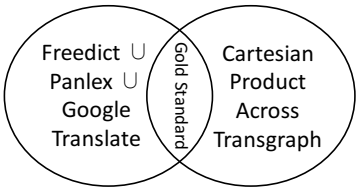


Figure 4.12: Creating Gold Standard for the High-Resource Case Studies.

European language is the common ancestor of the Indo-European language family from which the rest of our case study languages originate. The second case study CS2 (deu-eng-nld) targets high-resource languages of German (deu) and Dutch (nld) with English (eng) as the pivot. The third case study CS3 (spa-eng-por) uses Spanish (spa) and Portugese (por) languages with English (eng) as the pivot. The fourth case study CS4 (deu-eng-ita) uses German (deu) and Italian (ita) languages with English (eng) as the pivot. We utilize Freedict, an open source online bilingual dictionary databases[1] as input dictionaries and combination of Freedict, Panlex - another bilingual dictionary databases[2], and Google Translate[3] as shown in Table 4.4 as dictionaries for evaluation to create a gold standard. We use Google Translate to translate all headwords from cartesian product (CP) within the transgraphs. The gold standard is obtained by intersecting the

[1]http://freedict.org
[2]http://panlex.org
[3]http://translate.google.com

Table 4.5: Structure of Input Dictionaries and Gold Standard

Case Study	min-ind-zlm			deu-eng-nld			spa-eng-por			deu-eng-ita		
Language	min	ind	zlm	deu	eng	nld	spa	eng	por	deu	eng	ita
Headword	520	625	681	968	673	1,183	600	849	986	1,157	1,340	842
CP within transgraph		1,757			5,790			2,526			2,959	
CP across transgraph		354,120			1,145,144			591,600			974,194	
Gold Standard		1,246			1,438			1,069			1,503	

Table 4.6: Translation Relationship of Input Dictionaries

Case Study	Bilingual Dictionary	Translation Relationship							
		1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8
CS1	min-ind	267	210	36	5	1	1	0	0
	zlm-ind	563	115	3	0	0	0	0	0
CS2	deu-eng	785	165	16	2	0	0	0	0
	nld-eng	705	410	49	14	3	1	1	0
CS3	spa-eng	204	289	86	16	2	2	1	0
	por-eng	458	370	116	33	7	2	0	0
CS4	deu-eng	971	154	30	2	0	0	0	0
	ita-eng	256	421	129	25	7	2	1	1

combination of dictionaries for evaluation with CP across transgraph as shown in Figure 4.12. The structure of the input dictionaries and the gold standard for every case studies can be found in Table 4.5. The translation relationship of the input dictionaries varies from one-to-one until one-to-eight as shown in Table 4.6. For the low-resource case study, i.e., min-ind-zlm, the input dictionaries only have few one-to-many translation relations compared to the high-resource case studies. This shows that there are many potential missing senses in the input dictionaries. Consequently, sometimes we can miss some translation pair candidates across the transgraphs. Therefore, in this thesis, we limit our scope to extracting translation pairs within the transgraphs. We do not discriminate both single-word and multi-words expressions in the input dictionaries. After constructing the transgraphs from the input dictionaries, we find one big transgraph for each high-resource

Table 4.7: Size of the Biggest Transgraph

Case Study ($L_1 - P - L_2$)	L_1 Words	P Words	L_2 Words	Edges
CS1: min-ind-zlm	8	14	18	39
CS2: deu-eng-nld	4,669	2,486	6,864	18,548
CS3: spa-eng-por	2,347	2,465	4,460	15,043
CS4: deu-eng-ita	650	822	597	2,242

language case study as shown in Table 4.7. Sometimes, for high-resource languages where the input dictionaries have many shared meanings via the pivot words, a big transgraph can be generated which potentially leads to a computational complexity when we formalize and solve it. Nevertheless, for a low-resource languages where we can expect the input dictionaries only have a few shared meanings via the pivot words, the size of the transgraph is feasible to be formalized and solved. Therefore, for the sake of simplicity, we ignore any big transgraphs in these experiments.

Different users are likely to have different motivation, priority and preference when creating a bilingual dictionary. For high-resource languages, some users tend to priorities precision over recall while for low-resource languages, most users tend to priorities recall to enrich the language resource. In this research, we optimize the hyperparameters (cognate threshold and cognate synonym threshold) with a grid search by incrementing the cognate threshold from 0 to the highest cost of violating the constraints with 0.01 intervals and incrementing the cognate synonym threshold from 0 to 1 with 0.01 intervals in order to find the highest F-score.

4.4.2 Experiment Result

In all experiments and all case studies, all transgraphs are fully symmetrically connected on the third cycle, thus all possible translation pair candidates are reached. To extract many-to-many translation pairs, in the first step, i.e., cognate recognition and the second step, i.e., cognate synonym recognition, the soft-constraint violation threshold is set to reject all translation pairs returned by SATSolver that incurred a higher cost than the cognate threshold and cognate synonym threshold as shown in Algorithm 3 line number 4 and 10, respectively. Even though using the threshold to prioritize precision could yield the highest precision, the recall can be very low. Similarly, even though using the threshold to prioritize recall could yield the highest recall, the precision can also be harmed. Blindly prioritizing the precision over the recall or recall over the precision might not be a good strategy when implementing the framework.

Threshold Yielding The Highest F-score

To obtain a good strategy when we want to implement the framework, a balance between precision and recall is crucial. We calculate a harmonic mean of precision and recall using the traditional F1-measure or balanced F1-score by weighting the precision and recall equally. Based on user preference and priority, F0.5-score can be used when precision is considered more important, and F2-score can be used when recall is preferred. The results of all four case studies that targeted the threshold yielding the highest F-score are shown in Table 4.8. For the case study min-ind-zlm, our best yielding M-M result method (2:S:H14) yields 0.4% higher F-score than our

Table 4.8: Threshold Yielding The Highest F-score

Case Study	Method	Cognate Threshold	Cognate Synonym Threshold	Precision	Recall	F-score
	3:S:H14 (M-M)	4.79	1	0.656	0.998	0.792
	2:S:H14 (M-M)	4.79	0.74	0.735	0.923	0.818
	1:S:H14 (M-M)	4.17	1	0.836	0.713	0.770
	3:C:H14 (1-1)	4.79		0.884	0.331	0.481
	2:C:H14 (1-1)	4.79		0.884	0.331	0.481
CS1	1:C:H14 (1-1)	4.17		0.878	0.328	0.478
	Baseline: 2:M:H1 (M-M)			0.713	0.953	0.815
	Baseline: 1:M:H1 (M-M)			0.836	0.713	0.770
	Baseline: 1:C:H1 (1-1)			0.873	0.327	0.475
	Baseline: CP (M-M)			0.654	0.998	0.791
	Baseline: IC (M-M)			0.950	0.031	0.059
	3:S:H14 (M-M)	1.97	1	0.230	0.926	0.368
	2:S:H14 (M-M)	1.97	0.49	0.323	0.707	0.443
	1:S:H124 (M-M)	4.10	0.99	0.400	0.820	0.537
	3:C:H14 (1-1)	1.97		0.474	0.250	0.328
	2:C:H14 (1-1)	1.97		0.474	0.250	0.328
CS2	1:C:H124 (1-1)	4.10		0.472	0.249	0.327
	Baseline: 2:M:H1 (M-M)			0.257	0.919	0.402
	Baseline: 1:M:H1 (M-M)			0.397	0.821	0.536
	Baseline: 1:C:H1 (1-1)			0.447	0.238	0.311
	Baseline: CP (M-M)			0.230	0.926	0.368
	Baseline: IC (M-M)			0.612	0.078	0.138
	3:S:H34 (M-M)	3.01	1	0.368	0.870	0.517
	2:S:H34 (M-M)	3.01	0.49	0.467	0.751	0.576
	1:S:H14 (M-M)	3.21	0.66	0.569	0.765	0.653
	3:C:H34 (1-1)	3.01		0.716	0.367	0.486
	2:C:H34 (1-1)	3.01		0.716	0.367	0.486
CS3	1:C:H14 (1-1)	3.21		0.717	0.367	0.486
	Baseline: 2:M:H1 (M-M)			0.389	0.870	0.537
	Baseline: 1:M:H1 (M-M)			0.538	0.818	0.649
	Baseline: 1:C:H1 (1-1)			0.695	0.356	0.471
	Baseline: CP (M-M)			0.368	0.870	0.517
	Baseline: IC (M-M)			0.708	0.402	0.513
	3:S:H134 (M-M)	6.14	1	0.320	0.630	0.425
	2:S:H134 (M-M)	6.14	0.85	0.477	0.534	0.504
	1:S:H14 (M-M)	6.14	0.85	0.544	0.564	0.554
	3:C:H134 (1-1)	6.14		0.621	0.310	0.413
	2:C:H134 (1-1)	6.14		0.621	0.310	0.413
CS4	1:C:H14 (1-1)	6.14		0.626	0.310	0.415
	Baseline: 2:M:H1 (M-M)			0.377	0.627	0.471
	Baseline: 1:M:H1 (M-M)			0.542	0.565	0.553
	Baseline: 1:C:H1 (1-1)			0.600	0.298	0.398
	Baseline: CP (M-M)			0.320	0.630	0.424
	Baseline: IC (M-M)			0.930	0.071	0.131

previous best yielding M-M result method (2:M:H1), 3.4% higher F-score than CP, and 12.9 times higher F-score than IC, while our best yielding 1-

1 result method (3:C:H14) yields 1.3% higher precision than our previous method (1:C:H1). The high F-score of the CP in the case study min-ind-zlm indicates how very closely-related the input languages are. For the case study deu-eng-nld, our best yielding M-M result method (1:S:H124) yields 0.2% higher F-score than our previous best yielding M-M result method (1:M:H1), 46% higher F-score than CP, and 2.9 times higher F-score than IC, while our best yielding 1-1 result method (3:C:H14) yields 5.5% higher precision than our previous method (1:C:H1). For the case study spa-eng-por, our best yielding M-M result method (1:S:H14) yields 0.6% higher F-score than our previous best yielding M-M result method (1:M:H1), 26.3% higher F-score than CP, and 27.3% higher F-score than IC, while our best yielding 1-1 result method (3:C:H34) yields 3.6% higher precision than our previous method (1:C:H1). For the case study deu-eng-ita, our best yielding M-M result method (1:S:H14) yields 0.2% higher F-score than our previous best yielding M-M result method (1:M:H1), 30.7% higher F-score than CP, and 3.2 times higher F-score than IC, while our best yielding 1-1 result method (3:C:H134) yields 3.6% higher precision than our previous method (1:C:H1).

To enrich the bilingual dictionary result for low-resource languages, cognates and cognate synonyms recognition with higher cycles is the best approach. The exact number of cycles can be customized based on the priority and preference as regards the precision-recall trade-off. The cognates and cognate synonyms recognition with one-cycle is recommended for attaining the highest F-score result, since for almost all case studies in our experiments except min-ind-zlm, it always realized the highest F-score.

For the case study deu-eng-nld, the best one-to-one cognate (3:C:H14)

method precision is unexpectedly low, 0.474 while the lower language similarity case studies (spa-eng-por and deu-eng-ita) with the same cycle have higher precision (0.716 and 0.621 respectively). The case study deu-eng-nld always yielded lower F-scores than case studies deu-eng-ita and spa-eng-por when the methods that generate many-to-many results were applied. We believe that inadequacy of the gold standard was the cause of this counter-intuitive result. For the case study deu-eng-nld, if we look at the ratio of the size of the cartesian product across transgraph in Table 4.5 and the size of the combined dictionaries for evaluation in Table 4.4, relative to the ratio of the gold standard and the cartesian product within the transgraph, it is obvious that the ratio is inadequate compared to the other case study languages.

Statistical Significant Test

To show that our generalized methods are statistically significant compared to our previous methods [Nasution et al., 2016], as listed in Table 4.9-Table 4.12, for each case study, firstly, we split the dataset into several datapoints (transgraphs), then we compare the potentially best methods yielding the most many-to-many translation pairs (M-M), i.e., the 2:S:H14 to our previous method that potentially yielding the most many-to-many translation pairs (M-M), i.e., 2:M:H1. We also compare the potentially best methods yielding the most one-to-one translation pairs (1-1), i.e., the 2:C:H14 to our previous method that yielding one-to-one translation pairs (1-1), i.e., 1:C:H1. Student's paired t-test is a good statistical procedure used in Information Retrieval research to determine whether the mean difference between two sets of observations is zero [Smucker et al., 2007]. It is very

Table 4.9: Comparison of The Generalized Methods and The Previous Method: Case Study min-ind-zlm

	Transgraph	Previous Method			Generalized Method					
		Precision	Recall	F-score	Precision	Diff.	Recall	Diff.	F-score	Diff.
1-1*	0-24	0.920	0.548	0.687	0.920	0	0.548	0	0.687	0
	25-40	0.813	0.542	0.65	0.813	0	0.542	0	0.650	0
	41-56	0.813	0.520	0.634	0.875	+0.063	0.560	+0.040	0.683	+0.049
	57-72	1	0.516	0.681	1	0	0.516	0	0.681	0
	73-88	0.900	0.621	0.735	0.900	0	0.621	0	0.735	0
	89-104	0.889	0.471	0.615	0.889	0	0.471	0	0.615	0
	105-120	0.630	0.447	0.523	0.667	+0.037	0.474	+0.026	0.554	+0.031
	121-136	0.552	0.533	0.542	0.552	0	0.533	0	0.542	0
	137-152	0.828	0.500	0.623	0.862	+0.034	0.521	+0.021	0.649	+0.026
	153-168	0.966	0.346	0.509	1	+0.034	0.358	+0.012	0.527	+0.018
	169-184	1	0.352	0.520	1	0	0.352	0	0.520	0
	185-200	1	0.340	0.508	1	0	0.340	0	0.508	0
	201-216	0.975	0.312	0.473	0.975	0	0.312	0	0.473	0
	217-232	0.889	0.294	0.442	0.889	0	0.294	0	0.442	0
	233-248	0.866	0.179	0.296	0.878	+0.012	0.181	+0.003	0.301	+0.004
M-M**	0-24	0.913	1	0.955	0.913	0	1	0	0.955	0
	25-40	0.750	1	0.857	0.750	0	1	0	0.857	0
	41-56	0.781	1	0.877	0.781	0	1	0	0.877	0
	57-72	0.969	1	0.984	0.969	0	1	0	0.984	0
	73-88	0.725	1	0.841	0.725	0	1	0	0.841	0
	89-104	0.850	1	0.919	0.864	+0.014	1	0	0.927	+0.008
	105-120	0.644	1	0.784	0.644	0	1	0	0.784	0
	121-136	0.492	1	0.659	0.492	0	1	0	0.659	0
	137-152	0.774	1	0.873	0.774	0	1	0	0.873	0
	153-168	0.920	1	0.959	0.920	0	1	0	0.959	0
	169-184	0.938	1	0.968	0.938	0	1	0	0.968	0
	185-200	0.906	0.990	0.946	0.906	0	0.990	0	0.946	0
	201-216	0.886	0.992	0.936	0.886	0	0.992	0	0.936	0
	217-232	0.744	0.985	0.848	0.772	+0.028	0.949	-0.037	0.851	+0.003
	233-248	0.544	0.864	0.667	0.544	0	0.864	0	0.667	0

* Comparison between the previous method (1:C:H1 / Ω_1) [Nasution et al., 2016] and the generalized method (2:C:H14) which yield one-to-one translation pair results.

** Comparison between the previous method (2:M:H1 / Ω_3) [Nasution et al., 2016] and the generalized method (2:S:H14) which yield many-to-many translation pair results.

useful to show that our generalized methods are truly better than our previous methods rather than performed better by chance. In a student's paired t-test, each subject or entity is measured twice, resulting in pairs of observations. In this research, we use the same set of datapoints and conduct the student's paired t-test with precision and F-score as measures. Since we expect that our generalized methods have improvement compared to our

Table 4.10: Comparison of The Generalized Methods and The Previous Method: Case Study deu-eng-nld

	Transgraph	Previous Method			Generalized Method					
		Precision	Recall	F-score	Precision	Diff.	Recall	Diff.	F-score	Diff.
1-1*	0-16	0.529	0.900	0.667	0.588	+0.059	1	+0.100	0.741	+0.074
	17-34	0.478	0.478	0.478	0.522	+0.043	0.522	+0.043	0.522	+0.043
	35-52	0.594	0.463	0.521	0.594	0	0.463	0	0.521	0
	53-70	0.286	0.250	0.267	0.286	0	0.250	0	0.267	0
	71-88	0.406	0.271	0.325	0.500	+0.094	0.333	+0.063	0.400	+0.075
	89-106	0.447	0.333	0.382	0.447	0	0.333	0	0.382	0
	107-124	0.641	0.321	0.427	0.667	+0.026	0.333	+0.013	0.444	+0.017
	125-142	0.455	0.235	0.310	0.455	0	0.235	0	0.310	0
	143-160	0.439	0.220	0.293	0.512	+0.073	0.256	+0.037	0.341	+0.049
	161-178	0.333	0.237	0.277	0.426	+0.093	0.303	+0.066	0.354	+0.077
	179-196	0.526	0.265	0.353	0.517	-0.009	0.265	0	0.351	-0.002
	197-214	0.435	0.195	0.269	0.435	0	0.195	0	0.269	0
	215-232	0.408	0.228	0.293	0.380	-0.028	0.213	-0.016	0.273	-0.020
	233-250	0.410	0.211	0.279	0.457	+0.047	0.230	+0.019	0.306	+0.027
	251-268	0.446	0.135	0.208	0.485	+0.039	0.140	+0.004	0.217	+0.009
M-M**	0-16	0.417	1	0.588	0.417	0	1	0	0.588	0
	17-34	0.435	0.870	0.580	0.455	+0.020	0.870	0	0.597	+0.017
	35-52	0.559	0.927	0.697	0.587	+0.028	0.902	-0.024	0.712	+0.014
	53-70	0.329	0.844	0.474	0.329	0	0.844	0	0.474	0
	71-88	0.392	0.833	0.533	0.392	0	0.833	0	0.533	0
	89-106	0.349	0.882	0.500	0.366	+0.017	0.804	-0.078	0.503	+0.003
	107-124	0.531	0.987	0.691	0.531	0	0.987	0	0.691	0
	125-142	0.363	0.729	0.484	0.389	+0.026	0.659	-0.071	0.489	+0.005
	143-160	0.371	0.915	0.528	0.371	0	0.915	0	0.528	0
	161-178	0.274	0.961	0.427	0.304	+0.029	0.763	-0.197	0.434	+0.008
	179-196	0.330	0.947	0.490	0.330	0	0.947	0	0.490	0
	197-214	0.254	0.675	0.369	0.287	+0.033	0.643	-0.032	0.397	+0.027
	215-232	0.224	0.898	0.358	0.271	+0.047	0.646	-0.252	0.381	+0.023
	233-250	0.197	0.870	0.322	0.254	+0.056	0.671	-0.199	0.368	+0.046
	251-268	0.199	0.849	0.323	0.301	+0.102	0.561	-0.288	0.392	+0.069

* Comparison between the previous method (1:C:H1 / Ω_1) [Nasution et al., 2016] and the generalized method (2:C:H14) which yield one-to-one translation pair results.

** Comparison between the previous method (2:M:H1 / Ω_3) [Nasution et al., 2016] and the generalized method (2:S:H14) which yield many-to-many translation pair results.

previous methods, we choose a one-tailed t-test. There are two sets of null hypotheses (precision null hypotheses and F-score null hypotheses), which are that the true precision or F-score means difference between the generalized methods and our previous methods are equal to zero. We decide 0.05 cutoff value for determining statistical significance which corresponds to a 5% (or less) chance of obtaining a result like the one that was observed if

Table 4.11: Comparison of The Generalized Methods and The Previous Method: Case Study spa-eng-por

	Transgraph	Previous Method			Generalized Method					
		Precision	Recall	F-score	Precision	Diff.	Recall	Diff.	F-score	Diff.
1-1*	0-24	1	0.833	0.909	1	0	0.833	0	0.909	0
	25-45	0.714	0.750	0.732	0.714	0	0.750	0	0.732	0
	46-66	0.810	0.680	0.739	0.810	0	0.680	0	0.739	0
	67-87	0.762	0.421	0.542	0.762	0	0.421	0	0.542	0
	88-108	0.667	0.467	0.549	0.714	+0.048	0.500	+0.033	0.588	+0.039
	109-129	0.762	0.471	0.582	0.810	+0.048	0.500	+0.029	0.618	+0.036
	130-150	0.640	0.364	0.464	0.680	+0.040	0.386	+0.023	0.493	+0.029
	151-171	0.724	0.382	0.500	0.690	-0.034	0.364	-0.018	0.476	-0.024
	172-192	0.900	0.351	0.505	0.900	0	0.351	0	0.505	0
	193-213	0.600	0.296	0.397	0.600	0	0.296	0	0.397	0
	214-234	0.610	0.212	0.314	0.585	-0.024	0.203	-0.008	0.302	-0.013
	235-255	0.587	0.287	0.386	0.630	+0.043	0.309	+0.021	0.414	+0.029
	256-276	0.577	0.288	0.385	0.615	+0.038	0.308	+0.019	0.410	+0.026
	277-297	0.678	0.276	0.392	0.712	+0.034	0.290	+0.014	0.412	+0.020
	298-318	0.708	0.221	0.337	0.740	+0.031	0.231	+0.010	0.352	+0.015
M-M**	0-24	1	0.833	0.909	1	0	0.833	0	0.909	0
	25-45	0.714	0.750	0.732	0.714	0	0.750	0	0.732	0
	46-66	0.750	0.840	0.792	0.750	0	0.840	0	0.792	0
	67-87	0.667	0.737	0.700	0.667	0	0.737	0	0.700	0
	88-108	0.585	0.800	0.676	0.585	0	0.800	0	0.676	0
	109-129	0.596	0.824	0.691	0.596	0	0.824	0	0.691	0
	130-150	0.667	0.909	0.769	0.678	+0.011	0.909	0	0.777	+0.007
	151-171	0.632	0.782	0.699	0.646	+0.014	0.764	-0.018	0.700	+0.001
	172-192	0.663	0.766	0.711	0.663	0	0.766	0	0.711	0
	193-213	0.460	0.704	0.556	0.460	0	0.704	0	0.556	0
	214-234	0.438	0.534	0.481	0.458	+0.021	0.508	-0.025	0.482	+0.001
	235-255	0.433	0.830	0.569	0.433	0	0.830	0	0.569	0
	256-276	0.359	0.817	0.499	0.359	0	0.817	0	0.499	0
	277-297	0.360	0.862	0.508	0.433	+0.073	0.697	-0.166	0.534	+0.026
	298-318	0.255	0.779	0.384	0.359	+0.104	0.590	-0.189	0.446	+0.062

* Comparison between the previous method (1:C:H1 / Ω_1) [Nasution et al., 2016] and the generalized method (2:C:H14) which yield one-to-one translation pair results.

** Comparison between the previous method (2:M:H1 / Ω_3) [Nasution et al., 2016] and the generalized method (2:S:H14) which yield many-to-many translation pair results.

the null hypotheses were true. For all case studies min-ind-zlm, deu-eng-nld, spa-eng-por, and deu-eng-ita, we reject the precision null hypotheses since the p-value of the tests are 0.00732, 0.00007, 0.00398, 0.00464, respectively, which are all smaller than 0.05. For all case studies min-ind-zlm, deu-eng-nld, spa-eng-por, and deu-eng-ita, we also reject the F-score null hypotheses since the p-value of the tests are 0.01673, 0.00034, 0.00652,

Table 4.12: Comparison of The Generalized Methods and The Previous Method: Case Study deu-eng-ita

	Transgraph	Previous Method			Generalized Method					
		Precision	Recall	F-score	Precision	Diff.	Recall	Diff.	F-score	Diff.
1-1*	0-34	0.943	0.367	0.528	0.943	0	0.367	0	0.528	0
	35-64	0.633	0.235	0.342	0.700	+0.067	0.259	+0.025	0.378	+0.036
	65-94	0.700	0.300	0.420	0.667	-0.033	0.286	-0.014	0.400	-0.020
	95-124	0.533	0.246	0.337	0.667	+0.133	0.308	+0.062	0.421	+0.084
	125-154	0.667	0.274	0.388	0.667	0	0.274	0	0.388	0
	155-184	0.500	0.167	0.250	0.533	+0.033	0.178	+0.011	0.267	+0.017
	185-214	0.567	0.230	0.327	0.633	+0.067	0.257	+0.027	0.365	+0.038
	215-244	0.646	0.316	0.425	0.667	+0.021	0.327	+0.010	0.438	+0.014
	245-274	0.694	0.256	0.374	0.673	-0.020	0.248	-0.008	0.363	-0.011
	275-304	0.689	0.341	0.456	0.711	+0.022	0.352	+0.011	0.471	+0.015
	305-334	0.556	0.197	0.291	0.587	+0.031	0.213	+0.016	0.312	+0.021
	335-364	0.561	0.182	0.275	0.542	-0.019	0.182	0	0.272	-0.002
	365-394	0.540	0.177	0.267	0.556	+0.016	0.182	+0.005	0.275	+0.008
	395-424	0.519	0.169	0.256	0.532	+0.013	0.174	+0.004	0.262	+0.006
	425-454	0.544	0.184	0.275	0.562	+0.017	0.189	+0.005	0.283	+0.007
M-M**	0-34	0.946	0.389	0.551	0.946	0	0.389	0	0.551	0
	35-64	0.672	0.481	0.561	0.672	0	0.481	0	0.561	0
	65-94	0.627	0.529	0.574	0.627	0	0.529	0	0.574	0
	95-124	0.593	0.538	0.565	0.593	0	0.538	0	0.565	0
	125-154	0.610	0.493	0.545	0.610	0	0.493	0	0.545	0
	155-184	0.583	0.389	0.467	0.583	0	0.389	0	0.467	0
	185-214	0.633	0.514	0.567	0.633	0	0.514	0	0.567	0
	215-244	0.515	0.520	0.518	0.515	0	0.520	0	0.518	0
	245-274	0.535	0.406	0.462	0.535	0	0.406	0	0.462	0
	275-304	0.455	0.549	0.498	0.455	0	0.549	0	0.498	0
	305-334	0.444	0.441	0.443	0.444	0	0.441	0	0.443	0
	335-364	0.407	0.409	0.408	0.419	+0.012	0.398	-0.011	0.408	0
	365-394	0.367	0.438	0.399	0.401	+0.034	0.422	-0.016	0.411	+0.012
	395-424	0.331	0.462	0.386	0.379	+0.048	0.419	-0.042	0.398	+0.013
	425-454	0.226	0.488	0.309	0.339	+0.112	0.336	-0.152	0.338	+0.028

* Comparison between the previous method (1:C:H1 / Ω_1) [Nasution et al., 2016] and the generalized method (2:C:H14) which yield one-to-one translation pair results.

** Comparison between the previous method (2:M:H1 / Ω_3) [Nasution et al., 2016] and the generalized method (2:S:H14) which yield many-to-many translation pair results.

0.00783, respectively, which are all smaller than 0.05. Thus, our generalized methods have statistically significant improvement of precision and F-score compared to our previous methods.

Hyperparameter Optimization

We have shown that our generalized methods outperformed the baselines in the previous sections. Nevertheless, before implementing our model in a big scale, we need to validate how good our model perform in practice with unknown data. Since there is not enough data available to partition it into separate training and test sets without losing significant modelling or testing capability, a good way to properly estimate model prediction performance is to use cross-validation as a powerful general technique. Due to the computational complexity of our model, we conduct 3-folds cross validation to predict the optimal hyperparameters (cognate threshold and cognate synonym threshold) to gain the highest F-score as shown in Table 4.13. We optimize the hyperparameters with a grid search by incrementing the cognate threshold from 0 to the highest cost of violating the constraints with 0.01 intervals and incrementing the cognate synonym threshold from 0 to 1 with 0.01 intervals in order to find the highest F-score. We choose the same methods as in Table 4.9-Table 4.12, the potentially best methods yielding the most one-to-one translation pairs (1-1), i.e., the 2:C:H14 and the potentially best methods yielding the most many-to-many translation pairs (M-M), i.e., the 2:S:H14. For all case studies, the mean F-score approaches the mean F-score of the over-fitting model in Table 4.9-Table 4.12.

4.5 Conclusion

Our strategy to create high quality many-to-many translation pairs between closely-related languages consists of two steps. We first recognize cognates

Table 4.13: Cognate Threshold and Cognate Synonym Threshold Optimization

Case Study	Method	Validation Set	Optimal T.*		Testing on Unknown Data				
			C	S	Test Set	Precision	Recall	F-score	Mean F-score
CS1	2CH14	0-82, 83-165	1.35	-	166-248	0.933	0.257	0.403	0.559
		0-82, 166-248	4.79	-	83-165	0.786	0.471	0.589	
		83-165, 166-248	4.79	-	0-82	0.916	0.547	0.685	
	2SH14	0-82, 83-165	1.99	1	166-248	0.688	0.933	0.792	0.853
		0-82, 166-248	4.79	0.26	83-165	0.729	1	0.843	
		83-165, 166-248	4.79	0.26	0-82	0.858	1	0.924	
CS2	2CH14	0-90, 91-179	1.85	-	180-268	0.467	0.185	0.265	0.359
		0-90, 180-268	1.97	-	91-179	0.493	0.285	0.361	
		91-179, 180-268	1.97	-	0-90	0.485	0.423	0.452	
	2SH14	0-90, 91-179	1.85	1	180-268	0.219	0.86	0.35	0.474
		0-90, 180-268	1.97	0.51	91-179	0.361	0.893	0.514	
		91-179, 180-268	1.97	0.51	0-90	0.41	0.878	0.559	
CS3	2CH14	0-106, 107-212	2.96	-	213-318	0.676	0.268	0.384	0.525
		0-106, 213-318	3.21	-	107-212	0.724	0.366	0.486	
		107-212, 213-318	3.21	-	0-106	0.804	0.628	0.705	
	2SH14	0-106, 107-212	2.96	0.51	213-318	0.394	0.636	0.487	0.639
		0-106, 213-318	3.21	0.51	107-212	0.603	0.756	0.671	
		107-212, 213-318	3.21	0.51	0-106	0.719	0.803	0.759	
CS4	2CH14	0-150, 151-302	1.5	-	303-454	0.557	0.202	0.297	0.371
		0-150, 303-454	6.14	-	151-302	0.652	0.279	0.391	
		151-302, 303-454	6.14	-	0-150	0.735	0.3	0.426	
	2SH14	0-150, 151-302	1.5	0.01	303-454	0.341	0.414	0.374	0.479
		0-150, 303-454	6.14	0.56	151-302	0.531	0.481	0.505	
		151-302, 303-454	6.14	0.56	0-150	0.67	0.478	0.558	

* Optimal Threshold; C: Cognate Threshold, S: Cognate Synonym Threshold.

from direct and indirect connectivity via pivot word(s) by iterating multiple symmetry assumption cycles to reach more cognates in the transgraph. Once we obtain a list of cognates, the next step identifies synonyms of those cognates.

The result of case studies showed that our method offers good performance on weakly related high-resource languages. Thus, our method has the potential to complement other bilingual dictionary creation methods like word alignment models using parallel corpora. Our method shows particularly high performance on the closely related low-resource language case study. Our generalized methods have statistically significant improvement of pre-

cision and F-score compared to our previous methods in spite of sacrificing the recall a little bit.

Our key research contribution is a generalized constraint-based bilingual lexicon induction framework for closely related low-resource languages. This generalization makes our method applicable for a wider range of language groups than the one-to-one approach. Our customizable approach allows the user to conduct cross validation to predict the optimal hyperparameters (cognate threshold and cognate synonym threshold) with various combination of heuristics and number of symmetry assumption cycles to gain the highest F-score. To the best of our knowledge, our study is the first attempt to recognize both cognates and cognate synonyms in bilingual lexicon induction.

Chapter 5

Plan Optimization to Bilingual Dictionary Induction

5.1 Introduction

Despite the high potential of our approach in enriching low-resource languages, when actually implementing our constraint-based bilingual lexicon induction, we need to consider the inclusion of a more traditional method like manually creating the bilingual dictionaries by bilingual native speakers. In spite of the high cost, this will be unavoidable if no machine-readable dictionaries are available. Given the various methods and costs that may need to be considered, we recently introduced a plan optimizer to find the feasible optimal plan of creating multiple bilingual dictionaries with the least total cost [Nasution et al., 2017b]. The plan optimizer should decides which bilingual dictionary to be invested first or induced right from the start

in order to obtain all possible combination of bilingual dictionaries with a satisfying size from the language set with the minimum total cost to be paid.

5.2 Motivating Scenario

In order to illustrate the needs of optimal plan for creating multiple bilingual dictionaries with the least total cost we present an example motivating scenario. Consider a stakeholder has a motivation to obtain all 10 combination of bilingual dictionaries from 5 languages with a minimum size of 2,000 translation pairs each. Currently, the stakeholder already has a bilingual dictionary of language 1 and 3 ($d_{(1,3)}$) with 2,100 translation pairs and two bilingual dictionaries ($d_{(1,2)}$ and $d_{(2,3)}$) with a number of translation pairs below 2,000. Obviously, the stakeholder can just hire native speakers to create and evaluate the bilingual dictionaries following the traditional investment plan to reach his goal with a total cost of C. However, he can save cost of bilingual dictionary creation by utilizing our constraint-based bilingual lexicon induction with a zero creation cost. Even though the resulting bilingual dictionary still needs to be evaluated by native speakers, by following the optimal plan, the stakeholder can cut about half of the total cost.

At this point, the reader might wonder that even before executing the optimal plan, how can we know that utilizing the constraint-based bilingual lexicon induction to enrich $d_{(2,3)}$ resulting a satisfying size bilingual dictionary above 2,000 translation pairs or below 2,000 translation pairs that need to be invested more by native speakers to fill in the gap? To answer this

question, the constraint-based bilingual lexicon induction precision need to be estimated in order to calculate the resulting size bilingual dictionary. This uncertainty is the research challenge that we want to address in the following sections by modeling beta distribution of constraint-based bilingual lexicon induction precision and further utilize it in formalizing plan optimization in creating bilingual dictionaries using Markov Decision Process (MDP), since MDP can handle planning under uncertainty. If one try to utilize both our constraint-based bilingual lexicon induction and manual creation by native speakers and try to create the plan (order of dictionary creation task to take) manually without our MDP approach, the total cost might be higher than following our MDP plan. Since the created bilingual dictionary can be used as input for inducing the other unsatisfying size dictionary, the order of dictionary creation task to take is crucial.

5.3 Modeling Constraint-based Bilingual Lexicon Induction Precision Distribution

The constraint-based bilingual lexicon induction has characteristics where it work better on closely-related languages and a higher polysemy pivot rate will hurt the precision. Having these positive and negative parameters, a beta distribution is the best distribution to model the constraint-based bilingual lexicon induction precision. A beta distribution is a family of continuous probability distributions defined on the interval [0, 1] parametrized by two positive shape parameters, denoted by α which positively affecting the probability (x-axis) and β which negatively affecting the probability

(x-axis). The two parameters control the shape of the distribution. Beta distribution is usually used in Bayesian statistics as prior distribution for either a proportion, or the probability of occurrence of an event, or the value of any random variable [0, 1] such as the reliability of a component [Gupta and Nadarajah, 2004]. The constraint-based bilingual lexicon induction precision is useful to estimate the resulting bilingual dictionary size. However, before actually implementing our constraint-based bilingual lexicon induction, it is difficult to precisely know the precision beforehand. We can treat the precision as a random variable [0, 1] that can be modeled with a beta distribution. When sample observations are not available, a beta distribution can be defined by using subjective information [Fente et al., 1999]. A precision of the constraint-based bilingual lexicon induction for closely-related low-resource languages is likely to fall in the middle area between 0 and 1, and the likelihood is getting slimmer as the precision close to 0 or 1, therefore, the precision is better modeled with a bell-shaped beta distribution with $\alpha \geq 2$ and $\beta \geq 2$.

After determining the shape of beta distribution, we further model the α and β parameters for the prior beta distribution. Since α has a positive contribution to the precision, a language similarity of the target dictionary is a best fit because our constraint-based bilingual lexicon induction works better on a closely-related languages [Nasution et al., 2017a]. The language similarity between each language pair can be calculated with ASJP as discussed in Chapter 3. On the other hand, polysemy of the pivot word could cause a mistranslation when we induce a translation pair candidate from the connected edge in the transgraph as shown in Figure 4.9. However, considering low-resource languages have limited resources, our constraint-based

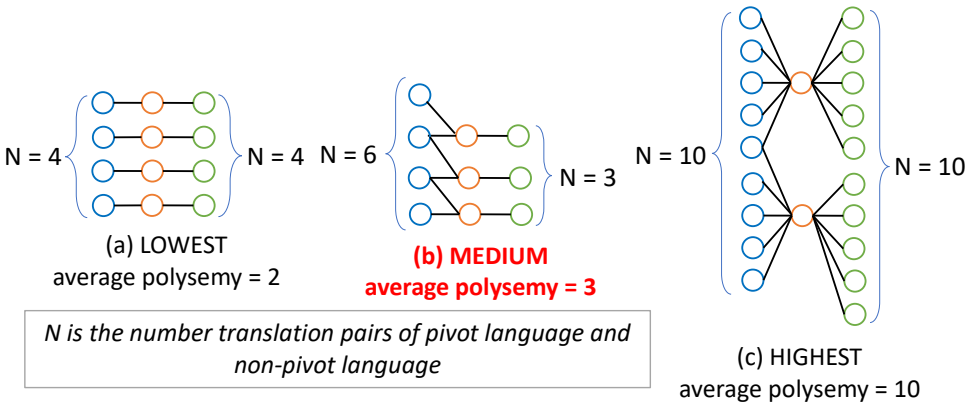


Figure 5.1: Average polysemy of the topology.

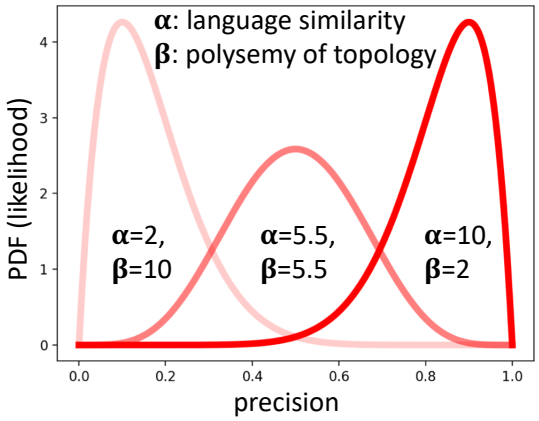


Figure 5.2: Variety of beta distribution bell-shaped depends on α and β.

bilingual lexicon induction only consider input bilingual dictionary as list of translation pairs without any additional information like part-of-speech or sense information. Therefore, we assume that an edge in a transgraph represents distinct sense/meaning. We define a polysemy of the topology as an average number of connected edges to pivot word in all transgraphs. When a one-to-one topology rate is 1 which means that every pivot word is only connected to one word from each of the non-pivot language as shown

in Figure 5.1a, the polysemy of the topology is the lowest = 2. When each pivot word is connected to five word from each of the non-pivot language as shown in Figure 5.1c, the polysemy of the topology is 10. We assume that the highest polysemy of topology is 10. The higher the polysemy of the topology, the more likely it is polysemous, hence negatively affect the constraint-based bilingual lexicon induction precision. So, we define β as the polysemy of the topology ranging from 2 to 10. The language similarity is normalized into $\alpha \in [2,10]$ to balance it with β. The beta distribution of constraint-based bilingual lexicon induction precision will have different bell-shaped depends on the α and β parameters as shown in Figure 5.2. The probability density function (PDF) is calculated by the following equation:

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}; 0 < x < 1; \alpha,\beta \geq 2 \tag{5.1}$$

After executing the constraint-based bilingual lexicon induction and further let native speaker to evaluate the result, the precision can be calculated. The likelihood's α parameter is calculated by normalizing precision to a range of [0, 10] and the β parameter is $10-\alpha$. A posterior beta distribution can be constructed using Bayes' theorem as shown in Equation (5.2). The posterior beta distribution α and β parameters are calculated by adding the prior beta distribution α and β parameters with the likelihood α and β parameters. The likelihood will contribute to adding believe toward the posterior beta distribution while not overwhelming the prior beta distribution because the likelihood's parameters are normalized close to the range of the prior beta distribution parameters.

$$posterior \propto prior \times likelihood \tag{5.2}$$

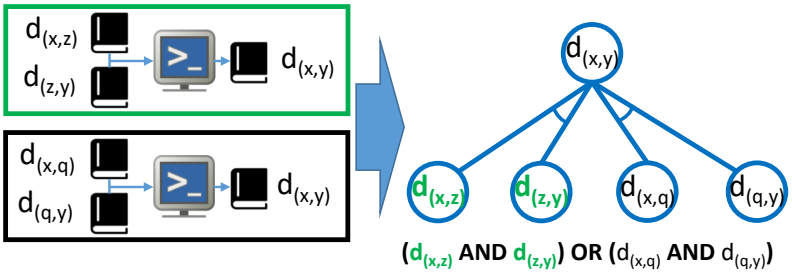


Figure 5.3: Modeling Bilingual Dictionary Induction Dependency.

5.4 Modeling Dictionary Dependency

Our constraint-based bilingual lexicon induction requires two bilingual dictionaries that share the same pivot language. We can induce bilingual $d_{(x,y)}$ from $d_{(x,z)}$ and $d_{(z,y)}$ as input (language z is the pivot). Nevertheless, we can also induce $d_{(x,y)}$ with different input bilingual dictionaries using language q as the pivot language for instance. We use an AND/OR graph to model the dependency: bilingual $d_{(x,y)}$ can be induced from $d_{(x,z)}$ and $d_{(z,y)}$ OR from $d_{(x,q)}$ and $d_{(q,y)}$ as shown in Figure 5.3.

If two sets of input dictionaries can be used to induce $d_{(x,y)}$, if we have to choose between the two sets, we need to prioritize input dictionaries that can induce $d_{(x,y)}$ with more correct translation pairs. But, the number of correct translation pairs that can be induced depends on the constraint-based bilingual lexicon induction precision and the size of translation pair candidates generated from the transgraph.

5.5 Formalizing Plan Optimization

The plan optimization to bilingual dictionary induction involves discovering the order of bilingual dictionary creation task from a set of possible tasks including constraint-based bilingual lexicon induction and manual creation by native speakers to minimize the total cost. We assume that the number of existing translation pairs for existing bilingual dictionaries and the minimum number of translation pairs the output bilingual dictionary should have, $size(d^m_{(x,y)})$, are both known. Multiple candidate plans exist to finally obtain all bilingual dictionaries. One criteria for selecting a plan is to establish a model of optimality and select the plan that is most optimal. We formulate the plan optimization in the context of creating multiple bilingual dictionaries from a set of language of interest as a constraint optimization problem (CSP). Formally, a constraint satisfaction problem is defined as a triple $\langle X, D, C \rangle$, where $X = \{X_1, \ldots, X_n\}$ is a set of variables, $D = \{D_1, \ldots, D_n\}$ is a set of the respective domains of values, and $C = \{C_1, \ldots, C_m\}$ is a set of constraints [Russell and Norvig, 2016].

5.5.1 Variable

If n is a number of target languages specified, the total number of all possible combinations of target bilingual dictionaries is $h = \binom{n}{2}$. For example, if we have 4 languages (L_1, L_2, L_3, L_4), there will be $h = \binom{4}{2} = 6$ target bilingual dictionaries: $d_{(1,2)}$, $d_{(1,3)}$, $d_{(1,4)}$, $d_{(2,3)}$, $d_{(2,4)}$, and $d_{(3,4)}$. A state S_i stores h bilingual dictionaries, each $d_{(x,y)}$ has four possible status types: not existing $d_{(x,y):n}$, existing but number of translation pairs is below minimum

dictionary size requested by user: $d_{(x,y):eu}$, induced with constraint-based bilingual induction with z as pivot language but the number of translation pairs is below minimum dictionary size requested by user: $d_{(x,y):pu(z)}$, and existing or induced constraint-based bilingual induction where the number of translation pairs equals or exceeds minimum dictionary size requested by user: $d_{(x,y):s}$, hence, the maximum number of state is $4^h = 4^6 = 4,096$. Based on the status, we further categorize the bilingual dictionary as either $SATDict$ ($d_{(x,y):s}$) or $UnSATDict$ ($d_{(x,y):n}$, $d_{(x,y):eu}$, or $d_{(x,y):pu(z)}$). A variable X_i is a possible bilingual dictionary creation method applied to enrich the size hence changing the status of bilingual dictionaries inside state S_i. The number of state increases exponentially with the number of target languages. So as to cast formulation complexity into a graph theory problem, we initially create only one start state S_1 along with variable X_1 where each bilingual dictionary status is labeled based on the size of existing bilingual dictionaries given by user. The following states S_2, S_3, ..., S_m and the respective variables X_2, X_3, ..., X_m are created as each value in domain D_i is defined.

5.5.2 Domain

Some bilingual dictionary creation methods such as the inverse consultation method, the one-to-one constraint-based approach, and our constraint-based bilingual lexicon induction require only bilingual dictionaries as input. However, since our method outperformed both previous methods, we model our method as one of value that can be assigned to variable X_i and call it pivot action $a^p_{(x,z,y)}$ to create dictionary $d_{(x,y)}$ where z is the pivot lan-

guage. For low-resource languages, adequate machine-readable bilingual dictionaries are often unavailable, so, we define another value, manual bilingual dictionary creation by a native speaker as investment action $a^{i}_{(x,y)}$. The purposes of assigning the two values, the pivot action and investment action, are to enrich the size and change the category of the bilingual dictionaries stored in each state S_i from $UnSATDict$ to $SATDict$.

5.5.3 Constraints for Domain Reduction

The following constraints are used to reduce the domain of a variable X_i.

Adequate Dictionary Size Constraint (C_1)

A dictionary $d_{(x,y)}$ inside a state S_i cannot be created or enriched if the dictionary status is $d_{(x,y):s}$ where the number of translation pairs equals or exceeds minimum dictionary size requested by user, $size(d^{m}_{(x,y)})$. In other word, neither $a^{i}_{(x,y)}$ nor $a^{p}_{(x,z,y)}$; for any pivot language z can be assigned to the variable X_i. If all dictionaries in a state S_i have a status of $d_{(x,y):s}$, there are no available value to be assigned to variable X_i in the domain D_i.

Initial Dictionary Status Constraint (C_2)

Initially, user provides information about the size of machine readable bilingual dictionaries if exist. The dictionary size information is mapped to a dictionary status of either $d_{(x,y):n}$, $d_{(x,y):eu}$, or $d_{(x,y):s}$. An $UnSATDict$ with status of $d_{(x,y):n}$ or $d_{(x,y):eu}$ inside a variable X_i can be enriched by both in-

vestment action $a^{i}_{(x,y)}$ and pivot action $a^{p}_{(x,z,y)}$. Both values can be assigned to the variable X_i.

One-Time Induction Constraint (C_3)

For an *UnSATDict* with status $d_{(x,y):pu(z)}$ inside a variable X_i, however, the next action is limited to investment action $a^{i}_{(x,y)}$ only, because pivot action $a^{p}_{(x,q,y)}$ was already executed exactly one step prior with q as pivot language. Thus, investment action $a^{i}_{(x,y)}$ is the only possible value to be assigned to the variable X_i.

Dictionary Induction Dependency Constraint (C_4)

A pivot action can be taken with a pair of input dictionary $d_{(x,z)}$ and $d_{(z,y)}$ as input when both of dictionaries have a status of s, where the number of translation pairs equals or exceeds minimum dictionary size requested by user, *eu*, which exists but the number of translation pairs is below minimum dictionary size requested by user, or $pu(z)$ induced with constraint-based bilingual induction with z as pivot language but the number of translation pairs is below minimum dictionary size requested by user. However, allowing dictionary with a status of $pu(z)$ as input can cause inconsistency of the translation pair result size. We consider the worst case scenario and choose the minimum translation pair result size. The bilingual dictionary induction dependency is shown in Figure 5.4.

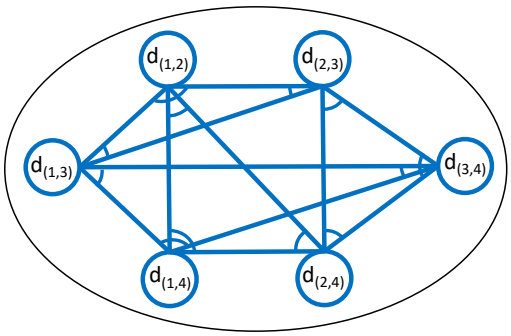


Figure 5.4: Bilingual Dictionary Induction Dependency Model.

5.5.4 Objective Function

In order to create or enrich bilingual dictionaries inside a state S_i, a constraint-based bilingual lexicon induction as pivot action $a^p_{(x,z,y)}$ or a manual bilingual dictionary creation by a native speaker as investment action $a^i_{(x,y)}$ can be assigned to a variable X_i. When we take an investment action, we are actually asking a bilingual native speaker to manually create and evaluate a bilingual dictionary and we need to pay for the time and effort incurred. On the other hand, for taking pivot action, i.e., using the constraint-based bilingual lexicon induction, when we already have the input dictionaries, we can generate the output dictionary in a short time. Thus, we assume that there is no cost for creating the bilingual dictionary, however, we still need to pay the bilingual native speaker to evaluate it.

Let W be the set of candidate plans. Let $C(w, X_i, a)$ be the cost function associated with assigning a value a in a corresponding domain D_i for variable X_i in some plan, w. The objective function is to minimize the expected total cost of assigning values in the corresponding domain to all variables while

satisfying all four constraints. A plan optimization is a way to find an optimal plan, w^*, that results in the minimal expected total cost of assignment. Formally,

$$\begin{aligned} w^* = \underset{w \in W}{\operatorname{argmin}} \quad & E\Big(\sum_{a \in D_i} C(w, X_i, a) \Big) \\ \text{subject to} \quad & \text{satisfying all constraints } C_1, C_2, C_3, C_4 \end{aligned} \tag{5.3}$$

The expectation operator, $E(.)$, in the above equation is necessary due to the stochastic nature of constraint-based bilingual lexicon induction. Based on the constraint-based bilingual lexicon induction precision, the resulting bilingual dictionary size can be above or below the minimum dictionary size requested by user, $size(d^m_{(x,y)})$. That is why we can only estimate the total cost before actually executing the task. A stochastic nature of the constraint-based bilingual lexicon induction is best handled by a Markov Decision Process (MDP), a well-known technique to solve problems containing uncertainty. Therefore, we model the plan application to bilingual dictionaries creation as a directed acyclic graph with MDP. A MDP has been used to model workflow composition and optimization [Doshi et al., 2004, Yu et al., 2005].

5.5.5 Markov Decision Process (MDP)

A MDP is the tuple $(S, A, T(s,a,s'), C(s,a,s'))$, where S is a set of states, A is a set of actions, $T(s,a,s')$ is a transition probability distribution over the state space when action a is taken in state s, and $C(s,a,s')$ is the negative reward or cost for taking action a in state s. The formalization to MDP is

described in Algorithm 4.

State

We model a MDP state similar with the way we define CSP variable. If n is a number of target languages specified, the total number of all possible combinations of bilingual dictionaries in the state is $h = \binom{n}{2}$ as shown in Algorithm 4 line number 1. Each state stores h bilingual dictionaries, each $d_{(x,y)}$ with four possible status types: not existing $d_{(x,y):n}$, existing but number of translation pairs is below minimum dictionary size requested by user: $d_{(x,y):eu}$, induced from pivot action with z as pivot language but the number of translation pairs is below minimum dictionary size requested by user: $d_{(x,y):pu(z)}$, and existing or induced from pivot action where the number of translation pairs equals or exceeds minimum dictionary size requested by user: $d_{(x,y):s}$, hence, the maximum number of MDP states is also $4^h = 4^6 = 4,096$. Based on the status, we further categorize the bilingual dictionary as either $SATDict$ ($d_{(x,y):s}$) or $UnSATDict$ ($d_{(x,y):n}$, $d_{(x,y):eu}$, or $d_{(x,y):pu(z)}$). Each state also stores information about the dependency between its dictionaries as shown in Figure 5.4. After an agent takes an action in state s to enrich an $UnSATDict$ of language x and y, if the size of the output dictionary satisfies minimum dictionary size requested by user, the agent will transit to the next one step ahead state, s'_{sat}, which has a $SATDict$ of the same languages, x and y, while the other bilingual dictionaries in s'_{sat} are unchanged from the previous state, s. On the other hand, if the size of the output dictionary below user request, the agent will transit to the next one step ahead state, s'_{unsat}, which has an $UnSATDict$ of the same languages, x and y, while the other bilingual dictionaries in s'_{unsat} are unchanged from

ALGORITHM 4: State Transition Graph Generation

Input: targetLanguages, targetLanguageInfo, existingDictionaries
/* 5 targetLanguages: [Indonesia "*ind*", Malay "*zlm*", Minangkabau "*min*", Javanese "*jav*", Sundanese "*sun*] */
/* targetLanguageInfo is a list of pair of language similarities and $size(d^{m}_{(x,y)}) = 2,000$ */
/* existingDictionaries=[$size(d_{(ind,zlm)}) = 711, size(d_{(ind,min)}) = 2,590, size(d_{(zlm,min)}) = 1,246$] */
Output: S, A, TS, T, C, dictionaryList /* Abbr: States, Actions, Target States, State Transition Probabilities, Costs */
/* Generate all $\binom{5}{2} = 10$ combinations. Initialize the size to 0 and status to not existing (n) */
dictionaryList ← generateDictionaryList(targetLanguages);
for *each* $d_{(x,y)}$ *in existingDictionaries* **do**
    dictionaryList.updateSizeAndStatus($d_{(x,y)}$);
end
S[0] ← createStartState(dictionaryList); /* In this example S[0] = [$d_{(ind,zlm):eu}$, $d_{(ind,min):s}$, $d_{(ind,jav):n}$, $d_{(ind,sun):n}$, $d_{(zlm,min):eu}$, $d_{(zlm,jav):n}$, $d_{(zlm,sun):n}$, $d_{(min,jav):n}$, $d_{(min,sun):n}$, $d_{(jav,sun):n}$] */
unvisitedStates.add(S[0]);
while *unvisitedStates is not empty* **do**
    state ← getStateWithLowestId(unvisitedStates);
    A[state] ← createPossibleActions(state); /* Adhere to all constraints in Section 5.5.3 */
    for *each action in A[state]* **do**
        TS[state, action] ← createTargetStates(state, action);
        for *each targetState in TS[state, action]* **do**
            T[state, action, targetState] ← calculateTransitionProb(state, action, targetState, targetLanguageInfo); /* Section (5.9), (5.10) */
            C[state, action, targetState] ← calculateCost(state, action, targetState, targetLanguageInfo); /* Section (5.11), (5.12) */
            unvisitedStates.add(targetState);
        end
    end
    unvisitedStates.remove(state);
end
return S, A, TS, T, C, dictionaryList;

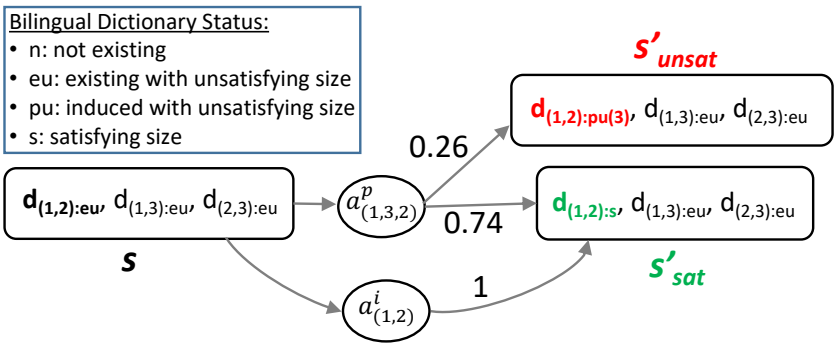


Figure 5.5: Example of State Transition.

the previous state, s. This is shown in Figure 5.5.

The number of states increases exponentially with the number of languages. So as to cast formulation complexity into a graph theory problem, we initially create only one start state where each bilingual dictionary status is calculated based on the input bilingual dictionaries size given by user as shown in Algorithm 4 line number 5. A list of unvisited states, *unvisitedStates* is initialized with the start state as shown in line number 6. For each possible action of each state, target states are generated. Each target state which is not in *unvisitedStates* list will be registered. After assigning all possible actions to the current state, it will be unregistered from the *unvisitedStates* list as shown in line number 18. The iteration is stopped when the *unvisitedStates* list is empty and the final state is reached where all m bilingual dictionaries from n languages are available and the number of translation pairs equals or exceeds user requested number of translation pairs.

Action

We also model a MDP action similar with the way we define CSP value in a domain. We apply our method as one of MDP action and call it pivot action $a^{p}_{(x,z,y)}$ to create dictionary $d_{(x,y)}$ where z is the pivot language. We also define manual bilingual dictionary creation by a native speaker as investment action $a^{i}_{(x,y)}$. The purposes of the pivot action and investment action are to enrich and change the category of the bilingual dictionaries stored in each state from UnSATDict to SATDict. Adhering to CSP constraints, we assign all possible actions to a state based on the state's situation as shown in Algorithm 4 line number 9. An UnSATDict with status $d_{(x,y):n}$ or $d_{(x,y):eu}$ can be enriched by both investment action and pivot action. For an UnSATDict with status $d_{(x,y):pu(z)}$, we limit the next action to investment action only because pivot action $a^{p}_{(x,z,y)}$ was already tried exactly one step prior. A pivot action can be taken from input dictionaries with status $d_{(x,y):s}$ and $d_{(x,y):eu}$.

State Transition Probability

The state transition probability from a state s to a target state s' after taking an action is calculated as shown in Algorithm 4 line number 13. The size of dictionaries in the current state affects performance of the pivot action taken in the current state, and thus the number of induced translation pairs in the next state. When the bilingual dictionary output by the pivot action $a^{p}_{(x,z,y)}$ in the current state s equals or exceeds minimum dictionary size requested by user, the agent will transit to the next state, s'_{sat} in which the bilingual dictionary status of languages x and y is $d_{(x,y):s}$ or else transit to the next state, s'_{unsat} in which the bilingual dictionary status of languages x and y

is $d_{(x,y):pu(z)}$ and the remaining bilingual dictionaries in the next state are unchanged from the previous state s as shown in Figure 5.5. In practice, we predict that the topology in Figure 5.1b is more likely to be generated, so, we estimate the number of translation pair candidates, $size(d^c_{(x,y)})$, twice the minimum size of the two input dictionaries. Formally,

$$size(d^c_{(x,y)}) = 2 \times \min\left\{size(d_{(x,z)}), size(d_{(y,z)})\right\} \tag{5.4}$$

The number of induced translation pairs is calculated by multiplying the pivot action precision with the number of translation pair candidates. Formally,

$$size(d_{(x,y)}) = precision(a^p_{(x,z,y)}) \times size(d^c_{(x,y)}) \tag{5.5}$$

To calculate the required number of translation pairs to be induced or invested, for dictionary with the following status: $d_{(x,y):eu}$ or $d_{(x,y):pu(z)}$, it can be obtained by subtracting the minimum dictionary size requested by user to the dictionary size $size(d_{(x,y):eu})$ or $size(d_{(x,y):pu(z)})$. Formally,

$$size(d^r_{(x,y)}) = size(d^m_{(x,y)}) - size(d_{((x,y))}) \tag{5.6}$$

However, for empty dictionary with no existing translation pairs: $d_{(x,y):n}$, the required number of translation pairs to be induced or invested equals the minimum dictionary size requested by user. Formally,

$$size(d^r_{(x,y)}) = size(d^m_{(x,y)}) \tag{5.7}$$

In order for $d_{(x,y)}$ to satisfy the required number of translation pairs, $size(d^r_{(x,y)})$, the pivot action precision should be at least equals to,

$$k = \frac{size(d^r_{(x,y)})}{size(d^c_{(x,y)})} \tag{5.8}$$

The state transition probability for taking a pivot action depends on the size of output bilingual dictionary which also depends on the precision of the constraint-based bilingual lexicon induction. If the precision is 1, then all translation pair candidates are taken as translation pairs. We model the state transition probability for taking a pivot action from the current state s and fail to satisfy the minimum dictionary size requested by user, $size(d^r_{(x,y)})$ and going to s'_{unsat} using beta distribution cumulative distribution function (CDF) ranging from 0 to k. Formally,

$$T(s,a,s'_{unsat}) = F(k;\alpha,\beta) = \int_0^k f(x;\alpha,\beta)dx \tag{5.9}$$

In the case of successfully satisfying the minimum dictionary size requested by user, $size(d^r_{(x,y)})$ and going to s'_{sat}, we use survival function. Formally,

$$T(s,a,s'_{sat}) = 1 - F(k;\alpha,\beta) = 1 - \int_0^k f(x;\alpha,\beta)dx \tag{5.10}$$

For instance, when we want to enrich UnSATDict $d_{(1,2):eu}$ from an existing dictionary size of $4,000$ to a minimum dictionary size requested by user, $size(d^m_{(x,y)}) = 10,000$, we can calculate the required number of translation pairs to be induced with Equation (5.6), $size(d^r_{(x,y)}) = 10,000 - 4,000 = 6,000$. If we enrich $d_{(1,2):eu}$ with pivot action $a^p_{(1,3,2)}$ from existing UnSATDict $d_{(1,3):eu}$ with input dictionary size equals $5,000$ and $d_{(2,3):eu}$ with input

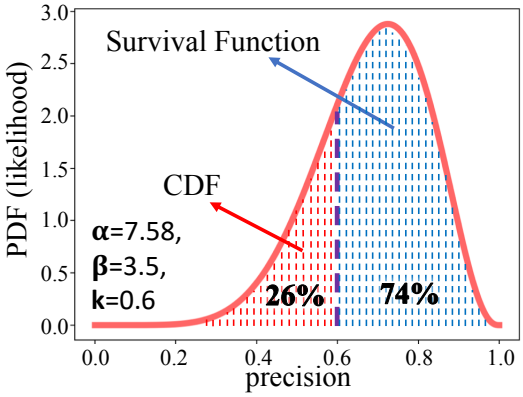


Figure 5.6: Cumulative distribution function (CDF) and survival function.

dictionary size equals 6,500, using Equation (5.4) we can get the number of translation pair candidates $size(d^{c}_{(1,2)}) = 2 \times 5{,}000 = 10{,}000$. Using Equation (5.8), we can calculate the minimum constraint-based bilingual lexicon induction precision, $k = 6{,}000/10{,}000 = 0.6$. If the beta distribution parameters are known, $\alpha = 7.58, \beta = 3.5$, using Equation (5.9), we can calculate the $T(s,a,s'_{unsat}) = 0.259$, and using Equation (5.10), we can calculate the $T(s,a,s'_{sat}) = 0.741$. As shown in Figure 5.6, there is 74% probability of getting precision above the minimum constraint-based bilingual lexicon induction precision to satisfy the required number of translation pairs to be induced, thus agent will transit to s'_{sat} and there is 26% probability of getting precision below the minimum constraint-based bilingual lexicon induction precision to satisfy the required number of translation pairs to be induced, thus agent will transit to s'_{unsat} as shown in Figure 5.5.

Cost

In the MDP model, the agent expects to get a reward after taking some actions. The reward will guide the agent to reach the final state and obtain the best path or in this case the best plan. Because for creating a bilingual dictionary we need to pay some cost instead of getting some rewards afterward, here we cast the reward as a cost. The terms of reward and cost are interchangeable in many previous MDP studies [White, 1993]. The cost of taking an action a from a state s to a target state s' is calculated as shown in Algorithm 4 line number 14. When we take an investment action, we are actually asking a native speaker to manually create and evaluate a bilingual dictionary and we need to pay for the time and effort incurred, however, in the MDP model, we define the cost as duration/time taken to do the task. To calculate the cost of taking investment action $a \in A^i$ from state s to state s', the required number of translation pairs is multiplied by both $creationCost$ and $evaluationCost$. By estimating 0.8 human accuracy for manual dictionary creation, the cost of investment action is as follow,

$$C(s,a,s') = \frac{size(d^r_{(x,y)})}{0.8} \times (creationCost + evaluationCost); a \in A^i \quad (5.11)$$

On the other hand, for taking pivot action, i.e., using the constraint-based bilingual lexicon induction, when we already have the input dictionaries, we can generate the output dictionary in a short time. Thus, we assume that there is no cost for creating the bilingual dictionary, in other word, the $creationCost = 0$, however, we still need to pay native speaker to evaluate it. To calculate the cost of taking pivot action $a \in A^p$ from state s to state s',

the number of translation pair candidates is multiplied by $evaluationCost$.

$$C(s,a,s') = size(d^{c}_{(x,y)}) \times evaluationCost; a \in A^{p} \tag{5.12}$$

Since the action cost, $C(s,a,s')$, for pivot action, depends on the number of translation pair candidates, $size(d^{c}_{(x,y)})$, and for investment action, depends on the required number of translation pairs, $size(d^{r}_{(x,y)})$, which are both calculated based on the size of the input dictionaries, which are unknown except for the existing dictionaries, we need to estimate the size of each dictionary in every state beforehand. This involves estimating the size of output dictionary in state s' after taking investment action and pivot action in state s. Based on Equation (5.11), estimating 0.8 human accuracy, we can easily predict the output dictionary by dividing the required number of translation pairs with 0.8, $size(d^{r}_{(x,y)})/0.8$. However, for pivot action, we need to estimate the precision of the constraint-based bilingual lexicon induction when the agent transit to s'_{sat} and s'_{unsat}. To calculate the expected value (mean) of a beta distribution, we can use the following Equation:

$$E(X) = \int_0^1 x f(x;\alpha,\beta)dx = \frac{\alpha}{\alpha+\beta} \tag{5.13}$$

However, the above equation consider the whole beta distribution, while we need to calculate upper mean and lower mean to estimate the precision of the constraint-based bilingual lexicon induction when the agent transit to s'_{sat} and s'_{unsat}, respectively. To do this, firstly, we need to truncate the beta distribution of constraint-based bilingual lexicon induction precision by k, the minimum precision to satisfy minimum dictionary size requested by user, $size(d^{m}_{(x,y)})$, and further calculate the upper mean and lower mean

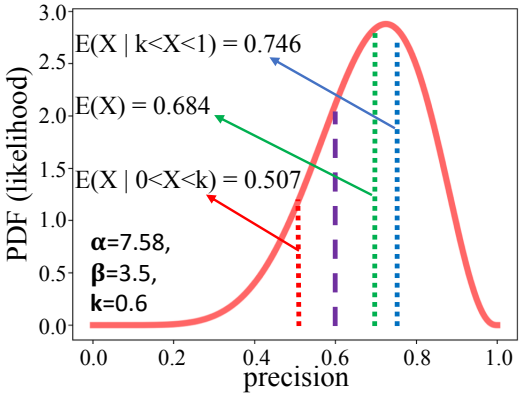


Figure 5.7: Mean of truncated beta distribution.

of the truncated beta distribution. This mean of a truncated distribution is pretty straightforward with a beta. For a positive random variable we have

$$E(X|X<k) = \frac{\int_0^k xf(x;\alpha,\beta)dx}{\int_0^k f(x;\alpha,\beta)dx} \tag{5.14}$$

Moving from Equation (5.1), we have

$$xf(x;\alpha,\beta) = \frac{B(\alpha+1,\beta)}{B(\alpha,\beta)}f(x;\alpha+1,\beta) = \frac{\alpha}{\alpha+\beta}f(x;\alpha+1,\beta) \tag{5.15}$$

Substituting Equation (5.15) to Equation (5.14), the mean of the truncated beta distribution is simplified as Equation (5.16) to calculate the lower mean of the truncated beta distribution to estimate the precision of the constraint-based bilingual lexicon induction when the agent transit to s'_{unsat}. Now the two integrals are just beta CDFs which are easily computed.

$$E(X|0<X<k) = \frac{\alpha}{\alpha+\beta}\frac{\int_0^k f(x;\alpha+1,\beta)dx}{\int_0^k f(x;\alpha,\beta)dx} \tag{5.16}$$

Following Equation (5.16), we can calculate the upper mean of the truncated beta distribution to estimate the precision of the constraint-based bilingual lexicon induction when the agent transit to s'_{sat} as

$$E(X|k<X<1) = \frac{\alpha}{\alpha+\beta}\frac{1-\int_0^k f(x;\alpha+1,\beta)dx}{1-\int_0^k f(x;\alpha,\beta)dx} \tag{5.17}$$

Using the same example, using Equation (5.13), Equation (5.16), and Equation (5.17), the beta distribution overall mean equals 0.684, lower mean equals 0.507, and upper mean equals 0.746 as shown in Figure 5.7. Now we can estimate the size of the SATDict, $size(d_{(x,y):s})$ and the UnSATDict, $size(d_{(x,y):pu(z)})$ after taking pivot action $a^p_{(x,z,y)}$ with Equation (5.18) and Equation (5.19), respectively.

$$size(d_{(x,y):s}) = E(X|k<X<1) \times size(d^c_{(x,y)}) \tag{5.18}$$

$$size(d_{(x,y):pu(z)}) = E(X|0<X<k) \times size(d^c_{(x,y)}) \tag{5.19}$$

Value Iteration

We use value iteration algorithm [Howard, 1960] to calculate utility (optimal policy) of each state by summing the cost for starting at state s and acting according to policies thereafter. Bellman [Bellman, 2013], via his Principle of Optimality, showed that the stochastic dynamic programming

equation given below is guaranteed to find the optimal policy for the MDP.

$$V_i(s) = \begin{cases} \min_{a \in A(s)} \sum_{s'} T(s,a,s') \big(C(s,a,s') + V_{i-1}(s') \big) & i > 0 \\ 0 & i = 0 \end{cases} \tag{5.20}$$

The above function, V_i, quantifies the long-term negative value, or cost, of reaching each state with i actions remaining to be performed. Every state will have a policy of best action in order to minimize cumulative costs. Once we know the cost associated with each state of the plan, the optimal action for each state is the one which results in the minimum expected cost. In Equation (5.21) below, π^* is the optimal policy which is simply a mapping from states to actions. Following the policy, we will obtain the optimal plan with the minimum cumulative costs.

$$\pi^*(s) = \operatorname*{argmin}_{a \in A(s)} \sum_{s'} T(s,a,s') \big(C(s,a,s') + V_{i-1}(s') \big) \tag{5.21}$$

5.6 Conclusion

Our constraint-based bilingual lexicon induction has the potential to enrich low-resource languages with the only input being machine readable bilingual dictionaries. Our MDP model can calculate the cumulative cost considering manual investment by bilingual native speakers while predicting and considering the probability of constraint-based bilingual lexicon induction to yield a satisfying output bilingual dictionary as utility for every state to better predict the most feasible optimal plan.

Chapter 6

A Collaborative Process to Create Bilingual Dictionaries of Indonesian Ethnic Languages

6.1 Introduction

To evaluate our MDP plan optimizer, we provide a sample experiment in Indonesia. Indonesia has 707 low-resource ethnic languages [Lewis et al., 2015] that require our attention. There are two factors we consider in selecting the target languages: language similarity and number of speakers. In order to ensure that the created bilingual dictionaries will be useful for many users, we listed the top 10 Indonesian ethnic languages ranked by the number of speakers. Since our constraint-based approach works best on closely related languages, we further generated the language similarity matrix by

Table 6.1: Similarity Matrix of Top 10 Indonesian Ethnic Languages Ranked by Number of Speakers

Language	**Indonesian**	Malang	Yogyakarta	**Javanese**	**Sundanese**	**Malay**	Palembang	Madurese	**Minangkabau**
Malang	23.46%								
Yogyakarta	27.29%	87.36%							
Javanese	**24.09%**	47.50%	52.18%						
Sundanese	**39.43%**	18.55%	22.43%	**21.82%**					
Malay	**85.10%**	20.53%	24.35%	**21.36%**	**41.12%**				
Palembang	68.24%	33.97%	37.97%	31.85%	38.90%	73.23%			
Madurese	34.45%	17.63%	14.15%	15.18%	19.86%	34.16%	34.32%		
Minangkabau	**61.59%**	26.59%	29.63%	**25.01%**	**30.81%**	**61.66%**	63.60%	34.32%	
Buginese	31.21%	12.76%	16.85%	18.33%	24.80%	32.04%	31.00%	17.94%	32.00%

utilizing ASJP as shown in Table 6.1 following the approach discussed in Chapter 3. Based on number of speaker, we select Javanese and Sundanese. To find and coordinate native speakers of those languages, we collaborated with Telkom University. Based on relatedness with Indonesian, we select Malay and Minangkabau. To find and coordinate native speakers of those language, we collaborated with Islamic University of Riau. Hence, we target 5 languages, i.e., Indonesian (ind), Malay (zlm), Minangkabau (min), Javanese (jav), and Sundanese (sun). We want to enrich/create the following dictionaries: $d_{(ind,zlm)}$, $d_{(ind,min)}$, $d_{(ind,jav)}$, $d_{(ind,sun)}$, $d_{(zlm,min)}$, $d_{(zlm,jav)}$, $d_{(zlm,sun)}$, $d_{(min,jav)}$, $d_{(min,sun)}$, and $d_{(jav,sun)}$ with at least 2,000 translation pairs each, $size(d^{m}_{(x,y)}) = 2,000$.

We model the *creationCost* and *evaluationCost* based on the availability of the native speakers. We provide example of modeling task for native speaker with Indonesian language families as target languages following our previous work [Nasution et al., 2018]. The detailed process of bilingual dictionaries generation process is explained in Algorithm 5.

ALGORITHM 5: Bilingual Dictionaries Generation

Input: S, A, TS, T, C, dictionaryList /* output of Algorithm 4 in Chapter 5 */
Output: dictionaryList /* all combination of bilingual dictionaries */

policy ← valueIteration(S, A, TS, T, C); /* Calculating policy, a mapping from State to Action using Equation (5.21) */
state ← S[0]; /* Start State */
while *state is not a finalState* **do**
    action ← policy.getAction(state);
    if *action.getType() = investment* **then**
        /* CT1(L_{ind}, L_x): Indonesia-Ethnic Dictionary Creation & Eval. */
        if L_x *or* L_y *is Indonesian language* L_{ind} **then**
            $d_{(x,y)}$ ← invest($s_{(x,y)}$); /* by a bilingual speaker */
            dictionaryList.updateSizeAndStatus($d_{(x,y)}$);
        end
        /* CT2(L_x, L_y): Ethnic-Ethnic Dictionary Creation & Eval. */
        else
            if *native bilingual speaker* $s_{(x,y)}$ *is available* **then**
                $d_{(x,y)}$ ← invest($s_{(x,y)}$); /* by a bilingual speaker */
                dictionaryList.updateSizeAndStatus($d_{(x,y)}$);
            end
            else
                $t_{(x,ind,y)}$ ← invest($s_{(ind,x)}$, $s_{(ind,y)}$); /* by two bilingual speakers */
                $d_{(x,y)}$ ← induce($t_{(x,ind,y)}$);
                dictionaryList.updateSizeAndStatus($d_{(x,y)}$);
            end
        end
    end
    else if *action.getType() = pivot* **then**
        $t_{(x,z,y)}$ ← pivot($d_{(x,z)}$, $d_{(z,y)}$); /* use constraint-based induction */
        /* T4 (L_x, L_z, L_y) */
        if *native bilingual speaker* $s_{(x,y)}$ *is available* **then**
            $t_{(x,z,y)}$ ← evaluate($t_{(x,z,y)}$, $s_{(x,y)}$); /* by a bilingual speaker */
            $d_{(x,y)}$ ← induce($t_{(x,z,y)}$);
            dictionaryList.updateSizeAndStatus($d_{(x,y)}$);
        end
        else
            $t_{(x,z,y)}$ ← evaluate($t_{(x,z,y)}$, $s_{(x,z)}$, $s_{(z,y)}$); /* by two bilingual speakers */
            induce $d_{(x,y)}$ from $t_{(x,z,y)}$;
            dictionaryList.updateSizeAndStatus($d_{(x,y)}$);
        end
    end
    state ← TS[state, action]; /* get the target state */
end
return dictionaryList;

6.2 Modeling Task for Native Speaker

Indonesian, a national language of Indonesia, is commonly used in both formal and informal settings, thus, almost everyone can speak Indonesian well. However, to create bilingual dictionary $d_{(x,y)}$ between ethnic language L_x and ethnic language L_y, there is a difficulty in finding a bilingual native speaker of the two ethnic languages. To overcome this limitation, we can firstly create triple $t_{(x,ind,y)}$ using the common language, Indonesian as pivot language L_{ind} where $s_{(ind,x)}$, a bilingual native speaker of Indonesian language L_{ind} - ethnic language L_x and $s_{(ind,y)}$, a bilingual native speaker of Indonesian language L_{ind} - ethnic language L_y collaborate by explaining the senses with Indonesian language. Then, the bilingual dictionary $d_{(x,y)}$ can be induced from the triple $t_{(x,ind,y)}$.

We measure the cost of creation/evaluation for each translation with a unit time which is calculated from the estimated time taken for doing the task and average daily wages of student part-time worker in Indonesia. This unit time simply shows that the creation cost of bilingual dictionary $d_{(ind,x)}$ is three times it's evaluation cost as shown in Figure 6.1 and Figure 6.2. When actually implementing our constraint-based bilingual lexicon induction, we need native speakers for manual creation of bilingual dictionaries or evaluation of the output dictionaries. We define several rules of which native speaker can create/evaluate which dictionary. A bilingual dictionary between ethnic language L_x and ethnic language L_y, $d_{(x,y)}$ can be induced from a triple $t_{(x,ind,y)}$, while a triple $t_{(x,ind,y)}$ can be induced from a bilingual dictionary $d_{(ind,x)}$ and a bilingual dictionary $d_{(ind,y)}$. A bilingual dictionary between Indonesian language L_{ind} and ethnic language L_x, $d_{(ind,x)}$

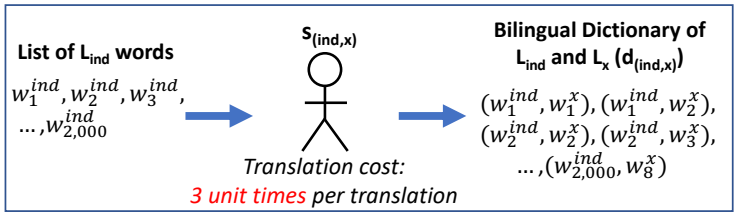


Figure 6.1: $T1(L_{ind}, L_x)$: Creation of Bilingual Dictionary $d_{(ind,x)}$.

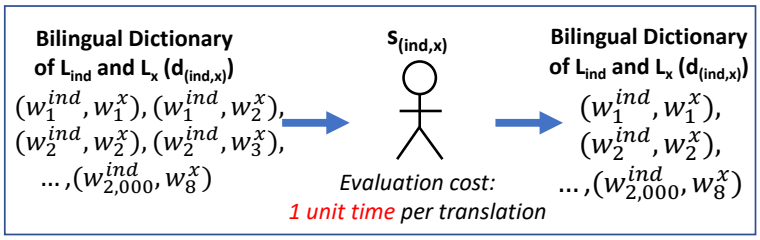


Figure 6.2: $T2(L_{ind}, L_x)$: Evaluation of Bilingual Dictionary $d_{(ind,x)}$.

can be manually created or evaluated by a native bilingual speaker $s_{(ind,x)}$ as shown in Algorithm 5 line number 6-9. A bilingual dictionary $d_{(x,y)}$ can be manually created or evaluated by a native bilingual speaker $s_{(ind,x)}$ and a native bilingual speaker $s_{(ind,y)}$ collaboratively as shown in Algorithm 5 line number 15-19 or by a native bilingual speaker $s_{(x,y)}$ alone as shown in Algorithm 5 line number 11-14. The incorrect triples $t_{(x,z,y)}$ output by the constraint-based bilingual lexicon induction are pruned by a native bilingual speaker $s_{(x,y)}$ individually as shown in Algorithm 5 line number 24-28 or by a native bilingual speaker $s_{(x,z)}$ and a native bilingual speaker $s_{(z,y)}$ collaboratively as shown in Algorithm 5 line number 29-33. There are some bilingual dictionaries between Indonesian and Indonesian ethnic languages exist in a printed format. We may be able to digitalized the printed Indonesian - ethnic language bilingual dictionaries to a machine readable format. Nevertheless, when we connect the digitalized bilingual dictionary $d_{(ind,x)}$

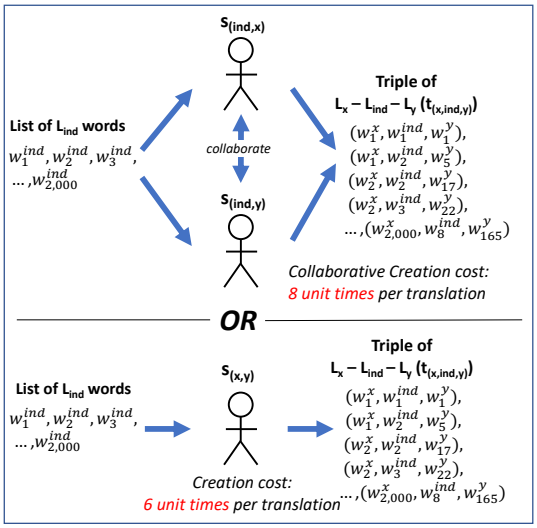


Figure 6.3: $T3(L_x, L_{ind}, L_y)$: (Individual/Collaborative) Creation of Triple $t_{(x,ind,y)}$ to induce Bilingual Dictionary $d_{(x,y)}$.

and a bilingual dictionary $d_{(ind,y)}$ via Indonesian language L_{ind} as a pivot, and further induced $d_{(x,y)}$ with our constraint-based approach, we expect that there will be many unreachable translation pair candidates since some Indonesian words in one bilingual dictionary may not exist in the other bilingual dictionary. In order to maximize the use of our pivot-based approach, we prepare a list of $2,000$ most commonly used Indonesian noun words to be translated to ethnic language L_x to create a bilingual dictionary $d_{(ind,x)}$ by a bilingual native speaker $s_{(ind,x)}$ as shown in Figure 6.1. Due to budget limitation, we only allow the native speaker to translate an Indonesian word to up to five words of ethnic language L_x.

To ensure the quality of the manually created bilingual dictionary $d_{(ind,x)}$, another bilingual native speaker $s_{(ind,x)}$ will evaluate the translation pairs as shown in Figure 6.2. We only pay correct translation pairs to the bilingual

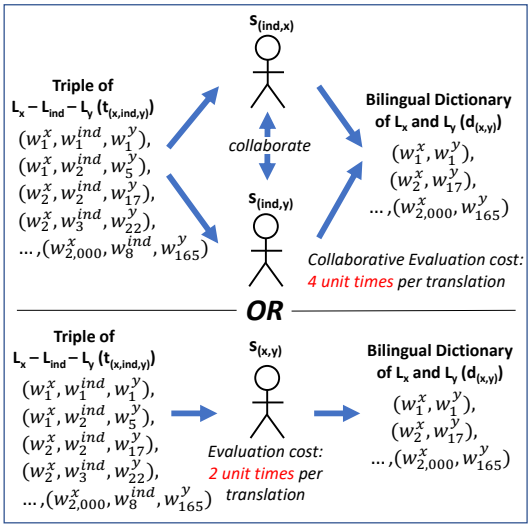
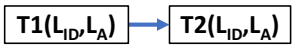


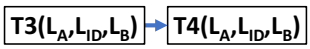
Figure 6.4: $T4(L_x, L_{ind}, L_y)$: (Individual/Collaborative) Evaluation of Triple $t_{(x,ind,y)}$ to induce Bilingual Dictionary $d_{(x,y)}$.

native speaker who does the creation task in order to motivate them to do the task carefully. To overcome the limitation in finding bilingual native speakers of two ethnic languages for creation and evaluation of bilingual dictionary $d_{(x,y)}$, two bilingual native speakers $s_{(ind,x)}$ and $s_{(ind,y)}$ can collaborate as shown in Figure 6.3 and Figure 6.4 respectively. Finally, there are two composite tasks, which are $CT1(L_{ind}, L_x)$, a manual creation followed by evaluation of bilingual dictionary $d_{(ind,x)}$ as shown in Figure 6.5a and $CT2(L_x, L_{ind}, L_y)$, a manual creation followed by evaluation of bilingual dictionary $d_{(x,y)}$ as shown in Figure 6.5b.

Finally, we integrate our constraint-based bilingual lexicon induction and plan optimizer with an online collaborative dictionary generation as a tool to bridge the spacial gap between native speakers [Nasution et al., 2018].



(a) $CT1(L_{ind}, L_x)$: Composite Task Creation and Evaluation of Bilingual Dictionary $d_{(ind,x)}$.



(b) $CT2(L_x, L_{ind}, L_y)$: Composite Task Creation and Evaluation of Bilingual Dictionary $d_{(x,y)}$.

Figure 6.5: Composite Tasks.

6.3 Online Collaborative Dictionary Generation

The online collaborative dictionary generation has 6 modules: individual creation of Indonesia-Ethnic bilingual dictionary, individual evaluation of Indonesia-ethnic bilingual dictionary, individual creation of ethnic-ethnic bilingual dictionary, individual evaluation of ethnic-ethnic bilingual dictionary, collaborative creation of ethnic-ethnic bilingual dictionary, and collaborative evaluation of ethnic-ethnic bilingual dictionary. Each native speakers get his/her own user account. They can login to the system, read the user manual, update their profile, check their assigned task, do their assigned task, view the created dictionaries, view the overall progress and finally view their salary. For the individual task, the native speakers can do the task anywhere before the deadline as shown in Figure 6.6. However, for the collaborative task, a pair of native speakers need to login to the system at the same time in order to collaborate. The live chat is used to ease communication and discussion during the collaborative creation/evaluation session as shown in Figure 6.7. The created and evaluated dictionaries can be viewed at dictionary interface along with statistic such as the worker's accuracy and evaluation result for each translation as shown in Figure 6.8

Figure 6.6: Individual Creation of Indonesia-Ethnic Bilingual Dictionary.

6.4 Plan Estimation

To show effectiveness of our method, we use, as a baseline, all investment plan where the only bilingual dictionary creation method used is the manual creation by native speakers without using our constraint-based bilingual lexicon induction as shown in Table 6.2. We further construct an estimated MDP optimal plan utilizing prior beta distributions of constraint-based bilingual lexicon induction precision for all 10 language pairs. We model α parameter from the language similarities shown in Table 6.1, which is a subset of language similarity matrix obtained in Chapter 3. Since in practice, we predict that the topology in Figure 5.1b is more likely to be

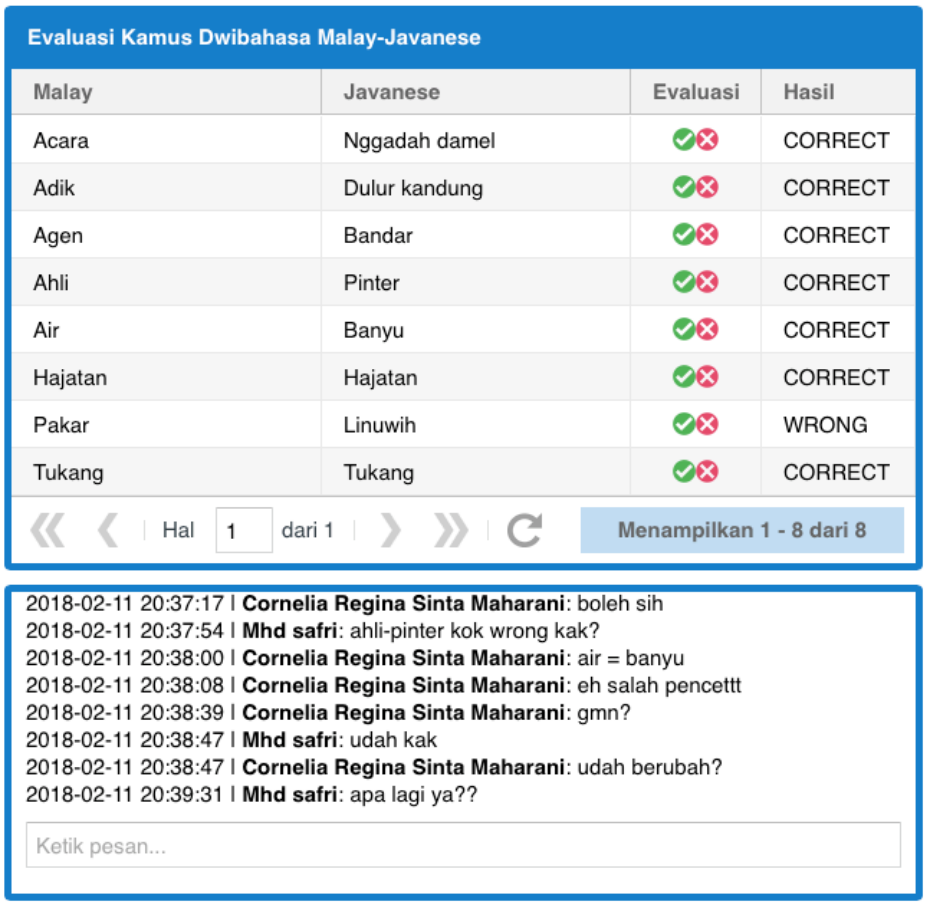


Figure 6.7: Collaborative Evaluation of Ethnic-Ethnic Bilingual Dictionary.

generated, so, we model β parameter by assuming all topology polysemy equals 3. We obtain the prior beta distribution for all language pairs as shown in Figure 6.9a-Figure 6.9j which will be used to calculate the MDP state transition probability and cost function. The estimated MDP plan is presented in Table 6.3.

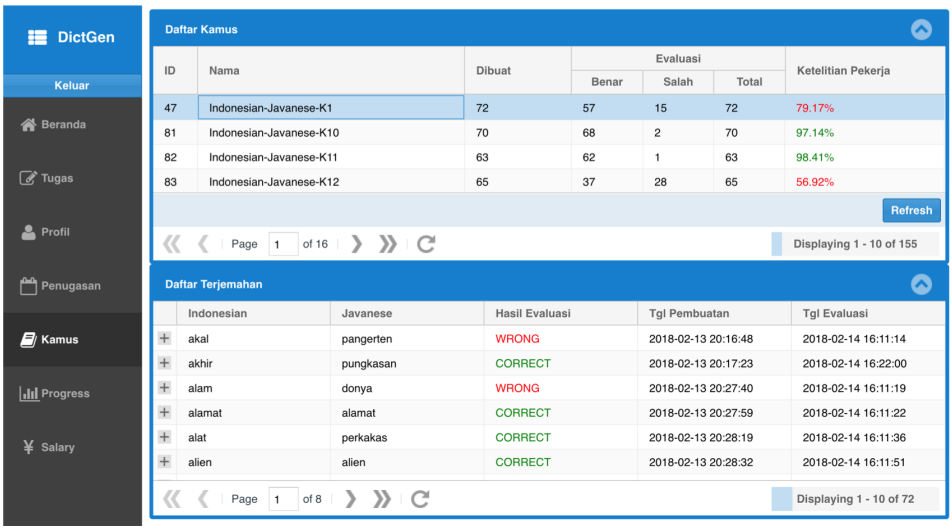


Figure 6.8: Dictionary Interface.

Table 6.2: Estimated Cost of Actions following All Investment Plan

Task Following Plan	#Translation Ordered[1]	#Translation Paid[2]	Total Cost (unit time)
CT1(ind, zlm) - 711 exist	1611.25	2900.25	5478
CT1(ind, jav)	2500	4500	8500
CT1(ind, sun)	2500	4500	8500
CT2(zlm, min) - 1246 exist	943	1697	9802
CT2(jav, sun)	2500	4500	26000
CT2(zlm, jav)	2500	4500	26000
CT2(min, sun)	2500	4500	26000
CT2(zlm, sun)	2500	4500	26000
CT2(min, jav)	2500	4500	26000
TOTAL			**162280**

[1] *Estimating 0.8 human accuracy.*

[2] *We only pay correct translation pairs to the bilingual native speaker who do the creation task to motivate them to do the task carefully.*

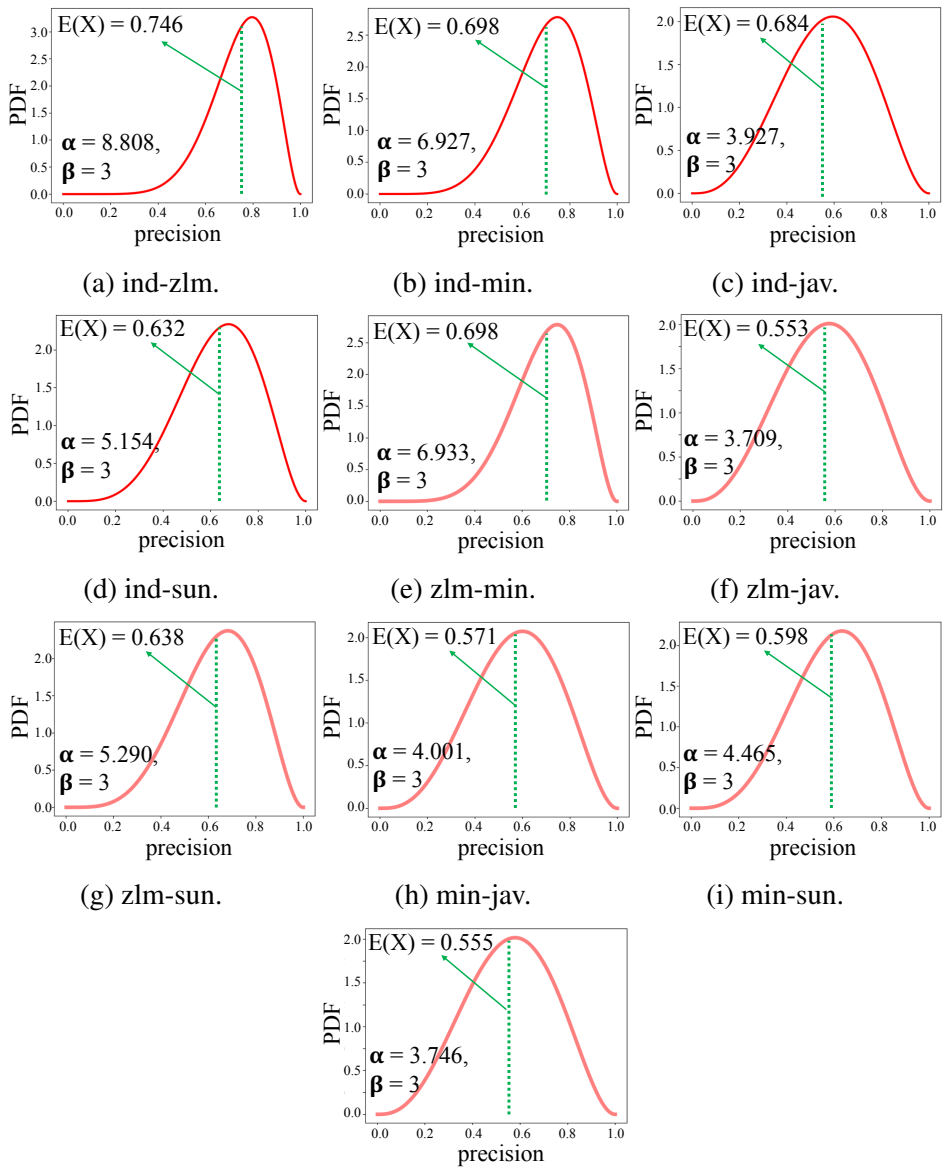


Figure 6.9: Prior Beta Distribution for All Language Pairs.

Table 6.3: Estimated Cost of Actions following MDP Optimal Plan

Task following Plan	#Translation Candidate[1]	Induction Precision[2]	#Translation Induced	Human Accuracy[3]	#Translation Paid[4]	Total Cost (unit time)
CT1(ind, zlm) - 711 exist				0.8	2900	5478
CT1(ind, jav)				0.8	4500	8500
CT1(ind, sun)				0.8	4500	8500
P(zlm, ind, min) - 1246 exist	4000	0.6981	2792			0
T4(zlm, ind, min)				1	2792	11170
P(jav, ind, sun)	5378	0.6108	3285			0
T4(jav, ind, sun)				1	32854	13140
P(zlm, ind, jav)	5388	0.6094	3283			0
T4(zlm, ind, jav)				1	3283	13134
P(min, ind, sun)	4000	0.6817	2727			0
T4(min, ind, sun)				1	2727	10907
P(zlm, ind, sun)	5553	0.6563	3644			0
T4(zlm, ind, sun)				1	3644	14578
P(min, zlm, jav)	4000	0.6735	2694			0
T4(min, zlm, jav)				1	2694	10776
TOTAL						**96182**

[1] *Estimated with Equation* (5.4).

[2] *Estimated from beta distribution based on language similarity as* α *and topology polysemy = 3 as* β.

[3] *Human accuracy for creation task is estimated as 0.8 and 1 for evaluation task.*

[4] *Only correct translation pairs are paid to motivate the bilingual native speaker to do the creation task carefully.*

6.5 Experiment Result

The result depicted in Table 6.4 shows that our MDP optimal plan outperformed the all investment plan as regards of total cost with 42% of cost reduction. The estimated total cost of actions following the MDP optimal plan shown in Table 6.3 is 97% close to the total cost in the real experiment with 3% of cost reduction. The average human accuracy shown in Table 6.4 is 0.837, close to our estimated human accuracy, 0.8. The average topology polysemy is 2.958, also close to our estimation, which is 3. The number of translation pair candidates of pivot actions: $a^{p}_{(jav,ind,sun)}$, $a^{p}_{(zlm,ind,jav)}$, and $a^{p}_{(min,zlm,jav)}$ are lower than our estimation, because the one-to-one relation rate of the topology like in Figure 5.1a are quite high, 0.61, 0.57, and 0.48, respectively.

From the experiment result, we can obtain the constraint-based bilingual lexicon induction precision. The likelihood's α parameter is calculated by normalizing precision to a range of [0, 10] and the β parameter is $10-\alpha$. As shown in Table 6.5, the posterior beta distribution α and β parameters are calculated by adding the prior beta distribution α and β parameters with the likelihood α and β parameters. Since the likelihood's α and β parameters are normalized to a range of [0, 10], close to the range of the prior beta distribution parameters [2, 10], the likelihood will contribute to adding believe toward the posterior beta distribution while not overwhelming the prior beta distribution. The posterior beta distribution of the six language pairs shown in Figure 6.10a-Figure 6.10f can be used for the future experiment when the same language pairs are used.

Table 6.4: Real Cost of Actions following MDP Optimal Plan

Task following Plan	#Translation Candidate	Polysemy (β)[1]	Induction Precision[2]	#Translation Induced	Human Accuracy[1]	#Translation Paid	Total Cost (unit time)
CT1(ind, zlm) - 711 exist					0.868	3338	6440
CT1(ind, jav)					0.790	4573	8610
CT1(ind, sun)					0.830	4517	8615
P(zlm, ind, min) - 1246 exist	5108	3.355	0.885	1940			0
T4(zlm, ind, min)					1	1940	7760
P(jav, ind, sun)	2181	2.498	0.824	2071			0
T4(jav, ind, sun)					1	2071	8284
CT2(jav, sun)					0.838	715	4164
P(zlm, ind, jav)	2170	2.583	0.801	2018			0
T4(zlm, ind, jav)					1	2018	8072
CT2(zlm, jav)					0.843	892	5200
P(min, ind, sun)	4155	3.300	0.802	2239			0
T4(min, ind, sun)					1	2239	8956
CT2(min, sun)					0.732	435	2557
P(zlm, ind, sun)	3691	2.824	0.833	2029			0
T4(zlm, ind, sun)					1	2029	8116
CT2(zlm, sun)					0.840	665	3896
P(min, zlm, jav)	2828	3.192	0.739	2069			0
T4(min, zlm, jav)					1	2069	8276
CT2(min, jav)					0.957	678	4760
TOTAL							**93707**

[1] *Closed to our estimation in Table 6.3.*

[2] *All pivot action precisions are higher than our estimation in Table 6.3.*

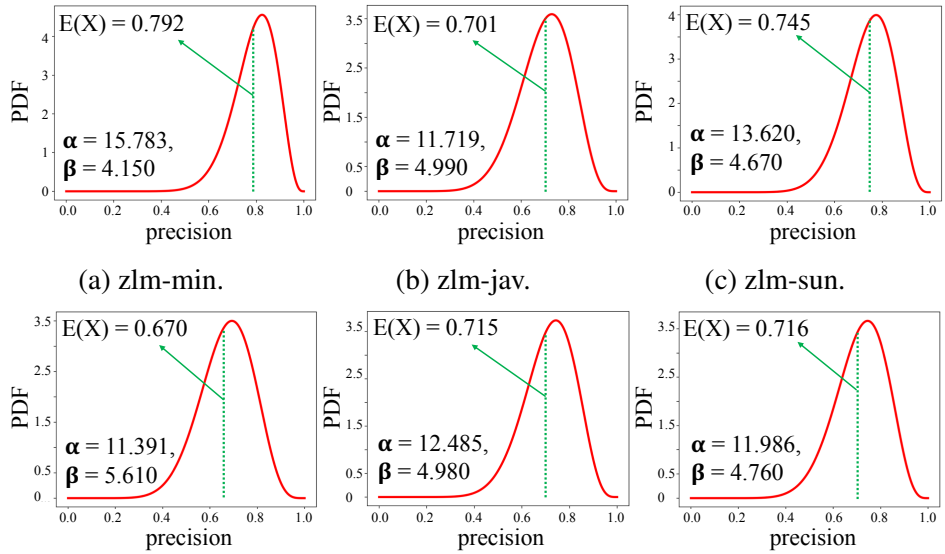
Table 6.5: Prior and Posterior Beta Distribution of Pivot Action Precision

Language Pair	Language Similarity	Prior[1]			Induction Precision	Likelihood[2]			Posterior[3]		
		α	β	E(X)		α	β	E(X)	α	β	E(X)
zlm-min	0.617	6.933	3.000	0.698	0.885	8.850	1.150	0.885	15.783	4.150	0.792
zlm-jav	0.214	3.709	3.000	0.553	0.801	8.010	1.990	0.801	11.719	4.990	0.701
zlm-sun	0.411	5.290	3.000	0.638	0.833	8.330	1.670	0.833	13.620	4.670	0.745
min-jav	0.250	4.001	3.000	0.571	0.739	7.390	2.610	0.739	11.391	5.610	0.670
min-sun	0.308	4.465	3.000	0.598	0.802	8.020	1.980	0.802	12.485	4.980	0.715
jav-sun	0.218	3.746	3.000	0.555	0.824	8.240	1.760	0.824	11.986	4.760	0.716

[1] *β parameter is an initial believe because we predict that the topology in Figure 5.1b is more likely to be generated, and α parameter is normalized to a range of [2, 10] to balance with the β parameter.*

[2] *The likelihood's α parameter is calculated by normalizing precision to a range of [0, 10] and the β parameter is $10 - \alpha$.*

[3] *The posterior beta distribution α and β parameters are calculated by adding the prior beta distribution α and β parameters with the likelihood α and β parameters.*



(a) zlm-min. (b) zlm-jav. (c) zlm-sun.

(d) min-jav. (e) min-sun. (f) jav-sun.

Figure 6.10: Posterior Beta Distribution for 6 Language Pairs.

6.6 Conclusion

Our constraint-based bilingual lexicon induction has the potential to enrich low-resource languages with the only input being machine readable bilingual dictionaries. Unfortunately, the scarcity of such dictionaries for low-resource languages makes it difficult to create an optimal plan to obtain all possible combination of bilingual dictionaries from the language set with the least total cost to be paid. Our MDP model can calculate the cumulative cost while predicting and considering the probability of the pivot action to yield a satisfying output bilingual dictionary as utility for every state to better predict the most feasible optimal plan.

Our key research contribution is twofold. For the earliest implementation of

our approach, a prior beta distribution of constraint-based bilingual lexicon induction precision is modeled with language similarity and topology polysemy as α and β parameters, respectively. After one episode of experiment, a posterior beta distribution can be constructed by utilizing the constraint-based bilingual lexicon induction precision as an added believe to the prior beta distribution while not overwhelming the prior beta distribution. The second key research contribution is the MDP optimal plan formalization itself. Following our MPD optimal plan, the actual total cost was 97% close to the estimated total cost, which shows that our plan optimizer is reliable. Our formalization allows user to get estimation of the feasible optimal plan with the least total cost before implementing the constraint-based bilingual lexicon induction in a big scale.

Chapter 7

Conclusion

7.1 Contributions

This thesis presented four contributions toward bilingual lexicon induction framework for closely related low-resource languages. The first is the language similarity cluster generation method of Indonesian ethnic languages with semi-supervised clustering. The resulting lexicostatistic and language similarity clusters are useful to determine the target languages. The second is the generalized constraint approach to bilingual dictionary induction for low-resource language families. This approach minimizes human touch on the bilingual dictionary creation. The third is the plan optimization to bilingual dictionaries induction for low-resource language families. With the optimal plan, user can get an idea of how our constraint-based approach perform in the actual implementation. The last is the design of a collaborative process to create bilingual dictionaries of Indonesian ethnic languages.

This includes implementation of both constraint-based approach and plan optimizer for creating 10 bilingual dictionaries between 5 Indonesian languages. We, in this section, review these contributions.

1. Introduce a language similarity cluster generation method of Indonesian ethnic languages.

   An approach of creating language similarity clusters is formulated by generating the language similarity matrix, followed by the hierarchical clusters. Two stable clusters with high language similarities are extracted from the hierarchical clusters. An extended k-means clustering semi-supervised learning is proposed to evaluate the stability level of the hierarchical stable clusters being grouped together regardless the change of the number of cluster. The two hierarchical stable clusters in the generated k-clusters are more likely to be distinctly found on a higher number of the trial. We take the 5 clusters as the best clusters of Indonesian ethnic languages since for all five experiments, the stability level of the two hierarchical stable clusters is the highest.

2. Propose a generalized constraint approach to bilingual dictionary induction.

   Our proposal extends constraints from the recent pivot-based induction technique and further enabling multiple symmetry assumption cycle to reach many more cognates in the transgraph. Cognate synonyms are identified to obtain many-to-many translation pairs. There are three baselines defined: three constraint-based methods from our previous work, the Inverse Consultation method and translation pairs generated from cartesian product of input dictionaries. We evaluate

our result using the metrics of precision, recall and F-score. Our approach is customizable and a cross validation can be conducted to predict the optimal hyperparameters (cognate threshold and cognate synonym threshold) with various combination of heuristics and number of symmetry assumption cycles to gain the highest F-score. Our proposed methods have statistically significant improvement of precision and F-score compared to our previous constraint-based methods. The experiment results on one Austronesian low-resource language and three Indo-European high-resource languages show that our method demonstrates the potential to complement other bilingual dictionary creation methods like word alignment models using parallel corpora for high-resource languages while well handling low-resource languages.

3. Propose a plan optimization to bilingual dictionary induction.
   We provide a MDP optimal plan formalization that allows user to get estimation of the feasible optimal plan with the least total cost before implementing the constraint-based bilingual lexicon induction in a big scale. For the earliest implementation of our approach, a prior beta distribution of constraint-based bilingual lexicon induction precision is modeled with language similarity and topology polysemy as α and β parameters, respectively. After one episode of experiment, a posterior beta distribution can be constructed by utilizing the constraint-based bilingual lexicon induction precision as an added believe to the prior beta distribution while not overwhelming the prior beta distribution.

4. Design a collaborative process to create bilingual dictionaries of In-

donesian ethnic languages.

To show applicability of the whole framework, we implement both the constraint-based approach and the plan optimizer for creating 10 bilingual dictionaries of Indonesian ethnic languages from every combination of 5 languages, i.e., Indonesian, Malay, Minangkabau, Javanese, and Sundanese. To overcome the difficulty of finding bilingual native speaker of two ethnic languages, we design a collaborative process where bilingual native speakers can create and evaluate a bilingual dictionary together. We provide an online collaborative dictionary generation tool to bridge spatial gap between native speakers. To evaluate our optimal plan with total cost as an evaluation metric, a heuristic plan that only utilizes manual investment by the native speaker is defined. We have shown that the optimal plan outperformed the heuristic plan with a 42% cost reduction. Following our MPD optimal plan, the actual total cost was 97% close to the estimated total cost. This show that our constraint-based approach and plan optimizer are applicable and reliable in enriching low-resource languages.

7.2 Future Direction

In this section, we will describe few areas for the future research. The proposed solutions open the possibility to pursue future directions as follows:

1. An online language similarity cluster generation system.

   An online language similarity cluster generation system with an ease of use features like exporting lexicostatistic to a CSV and plotting

similarity cluster to an interactive chart and a geographical map will be very useful not only to a computational linguistics researchers, but also to various other potential users like linguists, journalists, politicians, and government.

2. Incorporating more language resources for sense disambiguation of the pivot word.

   To improve the precision of the constraint-based bilingual lexicon induction, when available, WordNet and comparable corpora can be incorporated for tackling the polysemy ambiguity of the pivot word. A part-of-speech information of the input dictionary can also be utilized when available. However, when some language resources are only available for some languages and not for others, a mechanism to enable a fair treatment is crucial.

3. Dynamic plan optimization.

   The current plan optimization algorithm is static/offline as the policy is only calculated once in Algorithm 5 line number 1. After executing some actions from the static optimal plan, the previously optimal plan can be sub-optimal. For example, in our estimated MDP optimal plan shown in Table 6.3, all pivot action successfully induced bilingual dictionaries with a satisfying size, however, after following the MDP optimal plan, despite of the higher constraint-based bilingual lexicon induction precision compared to the estimation, only one out of six pivot actions successfully induced bilingual dictionaries with a satisfying size. This phenomena is due to the error in estimating the size of translation pair candidates. We estimated that all average polysemy of the topology will be medium as shown in Figure 5.1b while in real-

ity, we can find a lot of transgraph with a one-to-one relation with the lowest average polysemy of the topology as shown in Figure 5.1a. To make a dynamic/online plan optimization, we can update Algorithm 5 by adding a recursive procedure to re-formalize the problem with Algorithm 4 with updated information of the environment (size of translation pair candidates and dictionary status) every time after executing an action based on the current policy and further re-execute the new policy. This will make the planOptimizer adaptable to the changing of the environment. With a dynamic plan optimization, we can get a better estimation as well as reducing the computational complexity of the problem since the variable and the corresponding domain will be greatly reduced as more episodes of planOptimizer has been executed. There is also a possibility to relax One-Time Induction Constraint (C_3) into a soft-constraint. However, this could lead to an overlapped result when more than one constraint-based bilingual lexicon induction taken with different pivot languages. A discount parameter can be introduce to estimate the degree of overlapping result.

4. Remodeling task for native speakers.

   At the end of the experiment, through a short survey, we found out that the task modeling is too optimistic. In reality, the time taken for doing each task is longer than our estimation. For future experiment, a new estimated cost of action should be considered. Moreover, the online collaborative dictionary generation is also open for an upgrade and new useful features like a voice call to ease the discussion. New fields like example of word usage and part-of-speech information can be added to enrich the resulting bilingual dictionaries.

Publications

Journal

Arbi Haza Nasution, Yohei Murakami, and Toru Ishida. Generating Similarity Cluster of Indonesian Ethnic Languages with Semi-Supervised Clustering. *International Journal of Electrical and Computer Engineering (IJECE)*. 2018. (*in press*)

Arbi Haza Nasution, Yohei Murakami, and Toru Ishida. Plan Optimization to Bilingual Dictionary Induction for Low-Resource Language Families. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*. 2018. (*under review*)

Arbi Haza Nasution, Yohei Murakami, and Toru Ishida. A Generalized Constraint Approach to Bilingual Dictionary Induction for Low-Resource Language Families. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*. vol.17, no.2, article 9, pp.1-29, 2017.

Conferences

Arbi Haza Nasution, Yohei Murakami, and Toru Ishida. Designing a Collaborative Process to Create Bilingual Dictionaries of Indonesian Ethnic Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May, 2018.

Arbi Haza Nasution, Yohei Murakami, and Toru Ishida. Similarity Cluster of Indonesian Ethnic Languages. In *Proceedings of International Conference on Science, Engineering, and Technology (ICoSET 2017)*, Riau, Indonesia, pp 12-27, November, 2017.

Arbi Haza Nasution, Yohei Murakami, and Toru Ishida. Plan Optimization for Creating Bilingual Dictionaries of Low-Resource Languages. In *International Conference on Culture and Computing (Culture Computing)*, pp 35-41, 2017.

Arbi Haza Nasution, Yohei Murakami, and Toru Ishida. Constraint-Based Bilingual Lexicon Induction for Closely Related Languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3291–3298, Paris, France, May, 2016.

Other Publications

Arbi Haza Nasution. Pivot-based Hybrid Machine Translation to Support Multilingual Communication for Closely Related Languages. *World Transactions on Engineering and Technology Education*, 16, 2, 2018.

Arbi Haza Nasution, Nesi Syafitri, Panji Rahmat Setiawan, and Des Suryani. Pivot-Based Hybrid Machine Translation to Support Multilingual Communication. In *International Conference on Culture and Computing (Culture Computing)*, pp 147-148, 2017.

Bibliography

[Alfonseca et al., 2009] Alfonseca, E., Ciaramita, M., and Hall, K. (2009). Gazpacho and summer rash: Lexical relationships from temporal patterns of web search queries. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1046–1055, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Ansótegui et al., 2009] Ansótegui, C., Bonet, M. L., and Levy, J. (2009). Solving (weighted) partial maxsat through satisfiability testing. In *Theory and Applications of Satisfiability Testing-SAT 2009*, pages 427–440. Springer.

[Balcan et al., 2014] Balcan, M.-F., Liang, Y., and Gupta, P. (2014). Robust hierarchical clustering. *The Journal of Machine Learning Research*, 15(1):3831–3871.

[Bellman, 2013] Bellman, R. (2013). *Dynamic programming*. Courier Corporation.

[Beloucif et al., 2016] Beloucif, M., Saers, M., and Wu, D. (2016). Improving word alignment for low resource languages using english mono-

lingual srl. In *The 26th International Conference on Computational Linguistics (COLING 2016)*, page 51.

[Berg-Kirkpatrick and Klein, 2011] Berg-Kirkpatrick, T. and Klein, D. (2011). Simple effective decipherment via combinatorial optimization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 313–321, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Bergsma and Kondrak, 2007] Bergsma, S. and Kondrak, G. (2007). Alignment-based discriminative string similarity. In *Annual meeting-Association for Computational Linguistics*, volume 45, page 656.

[Biere et al., 2009] Biere, A., Heule, M., and van Maaren, H. (2009). *Handbook of satisfiability*, volume 185. IOS press.

[Bond and Ogura, 2008] Bond, F. and Ogura, K. (2008). Combining linguistic resources to create a machine-tractable japanese-malay dictionary. *Language Resources and Evaluation*, 42(2):127–136.

[Bond et al., 2001] Bond, F., Yamazaki, T., Sulong, R. B., and Okura, K. (2001). Design and construction of a machine–tractable japanese-malay lexicon. In *Annual Meeting of The Association for Natural Language Processing*, volume 7, page 1.

[Brown et al., 1990] Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

[Campbell, 2013] Campbell, L. (2013). *Historical linguistics*. Edinburgh University Press.

[Campbell and Poser, 2008] Campbell, L. and Poser, W. J. (2008). Language classification. *History and method. Cambridge*.

[Church and Gale, 1999] Church, K. and Gale, W. (1999). *Inverse Document Frequency (IDF): A Measure of Deviations from Poisson*, pages 283–295. Springer Netherlands, Dordrecht.

[Church and Gale, 1995] Church, K. W. and Gale, W. A. (1995). Poisson mixtures. *Natural Language Engineering*, 1(2):163–190.

[Danielsson and Muehlenbock, 2000] Danielsson, P. and Muehlenbock, K. (2000). Small but efficient: the misconception of high-frequency words in scandinavian translation. In *Envisioning Machine Translation in the Information Future*, pages 158–168. Springer.

[Déjean et al., 2002] Déjean, H., Gaussier, E., and Sadat, F. (2002). An approach based on multilingual thesauri and model combination for bilingual lexicon extraction. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Doshi et al., 2004] Doshi, P., Goodwin, R., Akkiraju, R., and Verma, K. (2004). Dynamic workflow composition using markov decision processes. In *Proceedings. IEEE International Conference on Web Services, 2004.*, pages 576–582.

[Echizen-ya et al., 2005] Echizen-ya, H., Araki, K., and Momouchi, Y. (2005). Automatic acquisition of bilingual rules for extraction of bilin-

gual word pairs from parallel corpora. In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, DeepLA '05, pages 87–96, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Fente et al., 1999] Fente, J., Knutson, K., and Schexnayder, C. (1999). Defining a beta distribution function for construction simulation. In *Proceedings of the 31st Conference on Winter Simulation: Simulation—a Bridge to the Future - Volume 2*, WSC '99, pages 1010–1015, New York, NY, USA. ACM.

[Fung, 1995] Fung, P. (1995). Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In *Proceedings of the 3rd Workshop on Very Large Corpora*, volume 78.

[Fung, 1998] Fung, P. (1998). A statistical view on bilingual lexicon extraction: from parallel corpora to non-parallel corpora. In *Machine Translation and the Information Soup*, pages 1–17. Springer.

[Fung, 2000] Fung, P. (2000). A statistical view on bilingual lexicon extraction. In *Parallel Text Processing*, pages 219–236. Springer.

[Fung and Yee, 1998] Fung, P. and Yee, L. Y. (1998). An ir approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 414–420, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Gooskens, 2006] Gooskens, C. (2006). Linguistic and extra-linguistic predictors of inter-scandinavian intelligibility. *Linguistics in the Nether-*

lands, 23(1):101–113.

[Guha et al., 1999] Guha, S., Rastogi, R., and Shim, K. (1999). Rock: A robust clustering algorithm for categorical attributes. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 512–521. IEEE.

[Gupta and Nadarajah, 2004] Gupta, A. K. and Nadarajah, S. (2004). *Handbook of beta distribution and its applications*. CRC press.

[Haghighi et al., 2008] Haghighi, A., Liang, P., Berg-Kirkpatrick, T., and Klein, D. (2008). Learning bilingual lexicons from monolingual corpora. *Proceedings of ACL-08: HLT*, pages 771–779.

[Hassine et al., 2006] Hassine, A. B., Matsubara, S., and Ishida, T. (2006). A constraint-based approach to horizontal web service composition. In *International semantic Web conference*, pages 130–143. Springer.

[Heeringa, 2004] Heeringa, W. J. (2004). *Measuring dialect pronunciation differences using Levenshtein distance*. PhD thesis.

[Holman et al., 2011] Holman, E. W., Brown, C. H., Wichmann, S., Müller, A., Velupillai, V., Hammarström, H., Sauppe, S., Jung, H., Bakker, D., Brown, P., et al. (2011). Automated dating of the world's language families based on lexical similarity. *Current Anthropology*, 52(6):841–875.

[Holman et al., 2008] Holman, E. W., Wichmann, S., Brown, C. H., Velupillai, V., Müller, A., and Bakker, D. (2008). Explorations in automated language classification. *Folia Linguistica*, 42(3-4):331–354.

[Howard, 1960] Howard, R. A. (1960). *Dynamic Programming and Markov Processes.* The M.I.T. Press.

[Hull and Grefenstette, 1996] Hull, D. A. and Grefenstette, G. (1996). Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 49–57, New York, NY, USA. ACM.

[Inkpen et al., 2005] Inkpen, D., Frunza, O., and Kondrak, G. (2005). Automatic identification of cognates and false friends in french and english. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 251–257.

[Irvine and Callison-Burch, 2017] Irvine, A. and Callison-Burch, C. (2017). A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

[Irvine et al., 2010] Irvine, A., Callison-Burch, C., and Klementiev, A. (2010). Transliterating from all languages. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 100–110.

[Ishida, 2006] Ishida, T. (2006). Language grid: An infrastructure for intercultural collaboration. In *Applications and the Internet, 2006. SAINT 2006. International Symposium on*, pages 5–pp. IEEE.

[Ishida, 2011] Ishida, T. (2011). *The Language Grid: Service-Oriented Collective Intelligence for Language Resource Interoperability*. Springer Publishing Company, Incorporated.

[Ishida, 2016] Ishida, T. (2016). Intercultural collaboration and support systems: A brief history. In *International Conference on Principles and*

Practice of Multi-Agent Systems (PRIMA 2016), pages 3–19. Springer.

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

[Kessler, 1995] Kessler, B. (1995). Computational dialectology in irish gaelic. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, pages 60–66. Morgan Kaufmann Publishers Inc.

[Klementiev et al., 2012] Klementiev, A., Irvine, A., Callison-Burch, C., and Yarowsky, D. (2012). Toward statistical machine translation without parallel corpora. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 130–140, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Klementiev and Roth, 2006] Klementiev, A. and Roth, D. (2006). Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 817–824, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Koehn and Knight, 2000] Koehn, P. and Knight, K. (2000). Estimating word translation probabilities from unrelated monolingual corpora using the em algorithm. In *AAAI/IAAI*, pages 711–715.

[Lam et al., 2015] Lam, K. N., Al Tarouti, F., and Kalita, J. K. (2015). Automatically creating a large number of new bilingual dictionaries. In *AAAI*, pages 2174–2180.

[Langfelder and Horvath, 2012] Langfelder, P. and Horvath, S. (2012). Fast r functions for robust correlations and hierarchical clustering. *Journal of statistical software*, 46(11).

[Lehmann, 2013] Lehmann, W. P. (2013). *Historical linguistics: an introduction*. Routledge.

[Levenshtein, 1966] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

[Lewis et al., 2015] Lewis, M. P., Simons, G. F., and Fennig, C. D., editors (2015). *Ethnologue: Languages of the World*. SIL International, Dallas, Texas, 18th edition.

[Li et al., 2009] Li, H., Kumaran, A., Pervouchine, V., and Zhang, M. (2009). Report of news 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, NEWS '09, pages 1–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Mann and Yarowsky, 2001] Mann, G. S. and Yarowsky, D. (2001). Multipath translation lexicon induction via bridge languages. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

[Matsuno and Ishida, 2011] Matsuno, J. and Ishida, T. (2011). Constraint optimization approach to context based word selection. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1846–1851. AAAI Press.

[Mausam et al., 2009] Mausam, Soderland, S., Etzioni, O., Weld, D. S., Skinner, M., and Bilmes, J. (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 - Volume 1*, ACL '09, pages 262–270, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Melamed, 1995] Melamed, I. D. (1995). Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 184–198. Boston, MA.

[Nagy, 1968] Nagy, G. (1968). State of the art in pattern recognition. *Proceedings of the IEEE*, 56(5):836–863.

[Nakov and Ng, 2012] Nakov, P. and Ng, H. T. (2012). Improving statistical machine translation for a resource-poor language using related resource-rich languages. *Journal of Artificial Intelligence Research*, 44:179–222.

[Nakov and Tiedemann, 2012] Nakov, P. and Tiedemann, J. (2012). Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual*

Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pages 301–305. Association for Computational Linguistics.

[Narasimhan et al., 2006] Narasimhan, M., Jojic, N., and Bilmes, J. A. (2006). Q-clustering. In *Advances in Neural Information Processing Systems*, pages 979–986.

[Nasution, 2018] Nasution, A. H. (2018). Pivot-based hybrid machine translation to support multilingual communication for closely related languages. *World Transactions on Engineering and Technology Education*, 16(2):12–17.

[Nasution et al., 2016] Nasution, A. H., Murakami, Y., and Ishida, T. (2016). Constraint-based bilingual lexicon induction for closely related languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3291–3298, Paris, France.

[Nasution et al., 2017a] Nasution, A. H., Murakami, Y., and Ishida, T. (2017a). A generalized constraint approach to bilingual dictionary induction for low-resource language families. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 17(2):9:1–9:29.

[Nasution et al., 2017b] Nasution, A. H., Murakami, Y., and Ishida, T. (2017b). Plan optimization for creating bilingual dictionaries of low-resource languages. In *2017 International Conference on Culture and Computing (Culture and Computing)*, pages 35–41.

[Nasution et al., 2017c] Nasution, A. H., Murakami, Y., and Ishida, T. (2017c). Similarity cluster of indonesian ethnic languages. In *Proceed-*

ings of the First International Conference on Science Engineering and Technology (ICoSET 2017), pages 12–27, Pekanbaru, Indonesia.

[Nasution et al., 2018] Nasution, A. H., Murakami, Y., and Ishida, T. (2018). Designing a collaborative process to create bilingual dictionaries of indonesian ethnic languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3397–3404, Paris, France. European Language Resources Association (ELRA).

[Nasution et al., 2017d] Nasution, A. H., Syafitri, N., Setiawan, P. R., and Suryani, D. (2017d). Pivot-based hybrid machine translation to support multilingual communication. In *2017 International Conference on Culture and Computing (Culture and Computing)*, pages 147–148.

[Petroni and Serva, 2008] Petroni, F. and Serva, M. (2008). Language distance and tree reconstruction. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(08):P08012.

[Pierrehumbert, 2012] Pierrehumbert, J. B. (2012). *Burstiness of Verbs and Derived Nouns*, pages 99–115. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Rapp, 1995] Rapp, R. (1995). Identifying word translations in non-parallel texts. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 320–322. Association for Computational Linguistics.

[Rapp, 1999] Rapp, R. (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, ACL '99, pages 519–526, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Richardson et al., 2015] Richardson, J., Nakazawa, T., and Kurohashi, S. (2015). Pivot-based topic models for low-resource lexicon extraction. In *PACLIC*.

[Rijsbergen, 1979] Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition.

[Russell and Norvig, 2016] Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

[Saers and Wu, 2009] Saers, M. and Wu, D. (2009). Improving phrase-based translation via word alignments from stochastic inversion transduction grammars. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, SSST '09, pages 28–36, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Saralegi et al., 2011] Saralegi, X., Manterola, I., and Vicente, I. S. (2011). Analyzing methods for improving precision of pivot based bilingual dictionaries. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 846–856. Association for Computational Linguistics.

[Scannell, 2006] Scannell, K. P. (2006). Machine translation for closely related language pairs. In *Proceedings of the Workshop Strategies for developing machine translation for minority languages*, pages 103–109.

[Schafer and Yarowsky, 2002] Schafer, C. and Yarowsky, D. (2002). Inducing translation lexicons via diverse similarity measures and bridge languages. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Shapley, 1953] Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

[Shezaf and Rappoport, 2010] Shezaf, D. and Rappoport, A. (2010). Bilingual lexicon generation using non-aligned signatures. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 98–107. Association for Computational Linguistics.

[Simons and (eds.), 2017] Simons, G. F. and (eds.), C. D. F. (2017). Ethnologue: Languages of the world, twentieth edition.

[Sjöbergh, 2005] Sjöbergh, J. (2005). Creating a free digital japanese-swedish lexicon. In *Proceedings of PACLING*, pages 296–300.

[Smucker et al., 2007] Smucker, M. D., Allan, J., and Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 623–632, New York, NY, USA. ACM.

[Snyder et al., 2010] Snyder, B., Barzilay, R., and Knight, K. (2010). A statistical model for lost language decipherment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL

'10, pages 1048–1057, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Soderland et al., 2009] Soderland, S., Etzioni, O., Weld, D. S., Skinner, M., Bilmes, J., et al. (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-Volume 1*, pages 262–270. Association for Computational Linguistics.

[Swadesh, 1950] Swadesh, M. (1950). Salish internal relationships. *International Journal of American Linguistics*, 16(4):157–167.

[Swadesh, 1952] Swadesh, M. (1952). Lexico-statistic dating of prehistoric ethnic contacts: with special reference to north american indians and eskimos. *Proceedings of the American philosophical society*, 96(4):452–463.

[Swadesh, 1955] Swadesh, M. (1955). Towards greater accuracy in lexicostatistic dating. *International journal of American linguistics*, 21(2):121–137.

[Swadesh, 2017] Swadesh, M. (2017). *The origin and diversification of language*. Routledge.

[Tanaka and Umemura, 1994] Tanaka, K. and Umemura, K. (1994). Construction of a bilingual dictionary intermediated by a third language. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, pages 297–303, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Tanaka et al., 2009] Tanaka, R., Murakami, Y., and Ishida, T. (2009). Context-based approach for pivot translation services. In *IJCAI*, volume 2009, pages 1555–1561.

[Tang and Van Heuven, 2015] Tang, C. and Van Heuven, V. J. (2015). Predicting mutual intelligibility of chinese dialects from multiple objective linguistic distance measures. *Linguistics*, 53(2):285–312.

[Tao et al., 2006] Tao, T., Yoon, S.-Y., Fister, A., Sproat, R., and Zhai, C. (2006). Unsupervised named entity transliteration using temporal and phonetic correlation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 250–257, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Tiedemann, 2009] Tiedemann, J. (2009). Character-based psmt for closely related languages. In *Proceedings of the 13th Conference of the European Association for Machine Translation (EAMT 2009)*, pages 12–19.

[Tsunakawa et al., 2008] Tsunakawa, T., Okazaki, N., and Tsujii, J. (2008). Building bilingual lexicons using lexical translation probabilities via pivot languages. LREC.

[Turney and Pantel, 2010] Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

[Van Bezooijen and Gooskens, 2005] Van Bezooijen, R. and Gooskens, C. (2005). How easy is it for speakers of dutch to understand frisian and afrikaans, and why? *Linguistics in the Netherlands*, 22(1):13–24.

[Virga and Khudanpur, 2003] Virga, P. and Khudanpur, S. (2003). Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition - Volume 15*, MultiNER '03, pages 57–64, Stroudsburg, PA, USA. Association for Computational Linguistics.

[White, 1993] White, D. J. (1993). A survey of applications of markov decision processes. *Journal of the Operational Research Society*, 44(11):1073–1096.

[Wichmann et al., 2010] Wichmann, S., Holman, E. W., Bakker, D., and Brown, C. H. (2010). Evaluating linguistic distance measures. *Physica A: Statistical Mechanics and its Applications*, 389(17):3632–3639.

[Wu and Wang, 2007] Wu, H. and Wang, H. (2007). Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, 21(3):165–181.

[Wushouer et al., 2014] Wushouer, M., Lin, D., Ishida, T., and Hirayama, K. (2014). *Pivot-Based Bilingual Dictionary Extraction from Multiple Dictionary Resources*, pages 221–234. Springer International Publishing, Cham.

[Wushouer et al., 2015] Wushouer, M., Lin, D., Ishida, T., and Hirayama, K. (2015). A constraint approach to pivot-based bilingual dictionary induction. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 15(1):4:1–4:26.

[Yamada and Knight, 1999] Yamada, K. and Knight, K. (1999). A compu-

tational approach to deciphering unknown scripts. In *Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing*, pages 37–44.

[Yoon et al., 2007] Yoon, S.-Y., Kim, K.-Y., and Sproat, R. (2007). Multilingual transliteration using feature based phonetic method. In *Annual Meeting-Association for Computational Linguistics*, volume 45, page 112.

[Yu et al., 2005] Yu, J., Buyya, R., and Tham, C. K. (2005). Cost-based scheduling of scientific workflow applications on utility grids. In *First International Conference on e-Science and Grid Computing (e-Science'05)*, pages 8 pp.–147.