Computational Linguistics and Chinese Language Processing
Vol. 25, No.2, December 2020, pp. 1-20
© The Association for Computational Linguistics and Chinese Language Processing

Analyzing the Morphological Structures in Seediq Words

Chuan-Jie Lin*, Li-May Sung+, Jing-Sheng You*,

Wei Wang*, Cheng-Hsun Lee*, and Zih-Cyuan Liao*

Abstract

NLP techniques are efficient to build large datasets for low-resource languages. It is helpful for preservation and revitalization of the indigenous languages. This paper proposes approaches to analyze morphological structures in Seediq words automatically as the first step to develop NLP applications such as machine translation. Word inflections in Seediq are plentiful. Sets of morphological rules have been created according to the linguisitic features provided in the Seediq syntax book (Sung, 2018) and based on regular morpho-phonological processing in Seediq, a new idea of "deep root" is also suggested. The rule-based system proposed in this paper can successfully detect the existence of infixes and suffixes in Seediq with a precision of 98.88% and a recall of 89.59%. The structure of a prefix string is predicted by probabilistic models. We conclude that the best system is bigram model with back-off approach and Lidstone smoothing with an accuracy of 82.86%.

Keywords: Seediq, Automatic Analysis of Morphological Structures, Deep Root, Natural Language Processing for Indigenous Languages in Taiwan, Formosan Languages

* Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan
E-mail: {cjlin, 10857039, 00657120, 00657140, 00672042}@email.ntou.edu.tw
+ Graduate Institute of Linguistics, National Taiwan University, Taipei, Taiwan
E-mail: limay@ntu.edu.tw
The authors for correspondence are Chuan-Jie Lin and Li-May Sung.

1. Introduction

1.1 Motivation

Machine learning and deep learning have been the most popular techniques in recent days. Systems built by machine learning or deep learning often achieve good performance, but the scale of the training sets in general should be large enough. Comparing to English, the amount of resources in Mandarin is far more small, not to mention the resources in the Southern Min, Hakka, even the indigenous languages in Taiwan. The United Nations has declared the Year of 2019 as the International Year of Indigenous Languages[1] in order to highlight the preservation issues of these endangered languages and gain more attention from the world. Following the same spirit, in January 2019 Taiwan has also promulgated the National Languages Development Act[2] (國家語言發展法) to speed up the preservation and revitalization of the indigenous languages in Taiwan.

The indigenous languages in Taiwan, well-known as Formosan languages[3] (台灣南島語言) in the Austronesian languages (南島語系) family, include 16 languages of 42 dialects in total. All are endangered to some degree according to the investigation by UNESCO in 2009. So far we have not found many researches on the natural language processing of the Formosan languages. Collaborating with a linguist and an expert in Seediq, one of the authors, this paper aims to provide an innovative first step on Seediq. In addition, we expect that the research results can be applied to linguistically related languages, Atayal (泰雅語) or Truku (太魯閣語) (Li, 1981), without much effort, or even to Amis (阿美語) which has the largest population of speakers and a similar writing system to Seediq.

The morphology in Seediq is quite complicated, including many word inflections to represent verbal focus, aspect and causation etc. For example, the morphological structure of the word "*psetuq*" (break, 斷) is "*p-setuq*", and the structure of the word "*qnyutan*" (bite, 咬) is "*q<n>yuc-an*", where "*p-*" (CAU, causative, 使動), "*<n>*" (PRFTV, perfective aspect, 完成貌), and "*-an*" (LV, locative voice, 處所焦點) are prefix, infix, and suffix, respectively. As we will discuss later, it is not easy to decompose the affixes and the stem in a Seediq word, but they carry important information for NLP tasks such as machine translation. This paper proposes the automatic approaches to analyze the morphological structures in Seediq as the first step of machine translation or other NLP tasks.

There is no large corpus in Seediq available so far. The experimental data in this paper came from the book "賽德克語語法概論" (A Sketch Grammar of Seediq) (Sung, 2018)

[1] https://en.iyil2019.org/

[2] https://law.moj.gov.tw/LawClass/LawAll.aspx?pcode=H0170143

[3] https://zh.wikipedia.org/wiki/台灣南島語言

(referred to as ***the Seediq syntax book*** hereafter). This book provides many sentences with morphological information as illustrations. We used these data to construct the training set. Dr. Li-May Sung, the author of the Seediq syntax book and one of the authors of this paper, provided another batch of sentences tagged with morphological information as well. We used them to construct the test set. There are 394 and 322 affixed Seediq words in these two datasets, far less than the necessary amount to train a classifier by machine learning or deep learning. One additional Seediq resource is an online Seediq dictionary "賽德克語德固達雅方言" (Tgdaya Seediq, referred to as ***the CIP Seediq dictionary*** hereafter) (Sung, 2011), compiled by Dr. Sung for the Council of Indigenous Peoples (原住民族委員會). There are about 5,600 words in this dictionary but with no morphological analysis. In the future we will apply the techniques developed in this paper to analyze these dictionary words in order to build up a larger dataset.

1.2 Related Work

To our best knowledge, there are not many researches on natural language processing of the Formosan languages. The most related studies are the ones done by Dr. Meng-Chien Yang in Tao (達悟語, aka. Yami 雅美語), including the construction of a wordnet and a lexicon in Yami (Yang & Rau, 2011; Yang *et al*., 2011; Rau *et al*., 2015), and machine translation between Yami and Mandarin under a small bilingual corpus (Yang & Rau, 2015).

There are many NLP studies for other local languages in Taiwan though, including machine translation for Taiwanese (Lin & Chen, 1999), speech recognition and synthesis in Taiwanese (Iunn *et al.*, 2007; Yu & Lin, 2012), and prosodic models in Hakka (Gu *et al*., 2007; Chiang, 2018). As we know that Taiwanese Southern Min, Hakka, and Mandarin belong to the Sinitic languages (漢語群), and they do not share similar language structure with the Formosan languages. Thus the research results cannot be applied directly to the Formosan languages.

In addition, there are limited electronic resources in Seediq available in the Internet. The CIP Seediq dictionary contains 5,595 words and 6,019 sentences with Mandarin translations. It is the largest dataset we can find so far. There are also textbooks for the elementary, junior-high and high schools available in "原住民族電子書城[4]" (Taiwanese Indigenous ebooks) and "族語 E 樂園[5]" (Formosan Languages E-Land), but their amounts are still comparatively small with no morphological analysis. Only sentences in the Seediq syntax book are tagged with morphological information.

A Seediq ontology was built by Dr. Shu-Kai Hsieh and Dr. Chu-Ren Huang (Hsieh *et al.*,

[4] https://alilin.apc.gov.tw/tw/ebooks

[5] http://web.klokah.tw/

2007). It contains 270 Seediq words mapping to the senses of WordNet in English in order to study the hyponymy relationships between Seediq words. As the ontology only covers a small set of Seediq words and provides mainly semantic information, we will not use it in this paper.

The development of a machine translation system usually requires a large size bilingual corpus in order to train a good-quality MT system by machine learning or deep learning (Bahdanau *et al*., 2015; Luong *et al*., 2015). It is important to create a large corpus efficiently by the help of NLP techniques, and this is the main goal we plan to do on Seediq in this paper.

2. Introduction to Seediq

2.1 Seediq Writing System

Seediq as one of the Formosan languages, the Seediq people mainly live in Nantou County and Hualien County. Linguistically belonging to the Atayalic subgroup (Li, 1981), Seediq is closely related to Atayal (泰雅語) and Truku (太魯閣語).

It has three dialects, including Tgdaya (德固達雅), Toda (都達), and Truku (德路固). Our experimental data came from the Seediq syntax book "賽德克語語法概論" (Sung, 2018) focusing on the Tgdaya dialect. Most morphological information about Seediq provided in this section also came from this syntax book.

The Seediq writing system follows the definition of "原住民族語言書寫系統" (writing systems of Formosan languages) published by the Ministry of Education and the Council of Indigenous Peoples on December 15th, 2005. It is a Romanization system. There are 18 consonants (including 2 half-vowels) and 5 vowels in Seediq. An example of a Seediq sentence is as follows.

[Seediq] Teta su kmkelun psetuq qnyutan su!
[Chinese] 看你咬得斷咬不斷！
(English: See if you can bite this off!)

2.2 Morphology in Seediq

The Seediq syntax book (Sung, 2018) provides detailed morphological information in each exemplar sentence to help the reader understand Seediq more efficiently. Words, especially the verbs, in the sentence are affixed to indicate actor voice (AV), patient voice (PV), locative voice (LV), beneficiary/instrumental/referential voices (BV/IV/RV), *etc*., and aspects such as perfective aspect (PFV). Affixation is overwhelmingly prevailing in Seediq. Such information is very useful in our study. One example of the morphological information is as follows.

[Morpho Info] teta=su kmekul-un p-setuq q<n>iyuc-an=su
[Explanation] 看看=你.屬格 能夠-受事焦點 使動-斷 <完成貌>咬-處所焦點=你.屬格
(English: See=you.GEN able-PV CAUS-break <PFV>bite-LV = you.GEN)

In this example, the root of the word "*qnyutan*" (bitten by, 被..咬) is "*qiyuc*". This word is affixed with a suffix "*-an*" (LV, locative voice, 處所焦點) and an infix "*<n>*" (PFV, perfective aspect, 完成貌), and becomes "*q<n>iyuc-an*". Similarly, the root of the word "*psetuq*" (broken, 使斷) is "*setuq*". This word is affixed with a prefix "*p-*" (CAUS, causative, 使動) and becomes "*p-setuq*". (GEN means genitive case. The symbol '=' represents the attachment of pronouns and other cases. It will not be discussed in this paper.) Several examples of word inflections are provided in Table 1.

Table 1. Examples of Seediq Word Inflections

Seediq	**Root**	**Morphological Structure**	**Meaning (Seediq / Root)**
mpkbeyax	beyax	m-p-k-beyax	hard-working, 努力 / do with force, 用力
cmnebu	cebu	c<m><n>ebu	shot successfully, 打中了 / shoot, 擊射
qyaanun	qeya	qeya-an-un	hang, 掛 / hang, 掛
pndsanan	adis	p<n>adis-an-an	bring back, 帶回 / bring, 帶

Notes: "m-": AV, agent voice 主事焦點; "p-": FUT, future 未來 or CAUS, causative 使動; "k-": STAT, stative 靜態; "<m>": AV, actor voice 主事焦點; "<n>": PFV, perfective aspect 完成貌; "-an": LV, locative voice 處所焦點; "-un": PV, patient voice 受事焦點

Another type of prefixes is reduplication (RED, 重疊) which repeats some part of the word. It is used for plurality, intensification, and *etc*. For example, the word "*sseediq*" ("*s-seediq*") (RED-person, 重疊-人) means "many people", and the word "*mkrkere*" ("*m-kr-kere*") (AV-RED-strong, 主事焦點-重疊-強壯) means something is very strong. Even prefixes can be repeated, such as in the word "*pposa*" ("*p-p-osa*") (RED-CAUS-go, 重疊-使動-去) which means "forced to go to somewhere". The reduplication usually does not change the meaning of a word but its amount or intensity, which could also be an issue in machine translation.

When a Seediq word is affixed, the final writing form can be different from its original combination, as we can see in the examples in Table 1. This is the reason why the morphological structure of a Seediq word cannot be generated directly from its surface form.

We discuss only three variation cases here (Sung, 2018; Yang, 1976; Li, 1977; Li, 1991).

The first case is related to vowel neutralization (元音中性化) and vowel reduction (元音脫落). In Seediq, vowels other than the last two syllables are weakened (neutralized) and omitted when writing. It usually happens in the suffixation process in Seediq. Take examples from Table 1. In the word "*qyaanun*" ("*qeya-an-un*"), the first vowel "*e*" of its root "*qeya*" is omitted when affixed. And in the word "*pndsanan*" ("*p<n>adis-an-an*"), both vowels of its root "*adis*" are omitted.

Consider another example. The word "*dngei*" ("*dengu-i*") consists of a root "*dengu*" (sun-dry; 曬乾) and a suffix "*-i*" (IMP, imperative, 祈使). We suggest that the root word "*dengu*" may originally be "*denge*": that is, the second vowel '*e*' is neutralized as '*u*' when it appears at the end of a word. When "*denge*" is suffixed with "*-i*", the accent falls on the second vowel '*e*' (hence not neutralized any more) and makes it remain as "*e*"; meanwhile, the first vowel "*e*" of "*denge*" is neutralized and omitted, resulting in "*dngei*". That is, the word "*dngei*" comes from the original structure of "*denge-i*". We refer to such original form of a root as its "deep root" and will discuss it in details in Section 3.1.

The second case is about vowel harmony (元音和諧變化). When a root word starts with a vowel, the preceding prefix usually ends with the same vowel. For example, if the prefix "*s-*" (RV, referential voice, 參考焦點) attaches to the root "*osa*" (go, 去), the prefix becomes "*so-*" and the final writing form is "*soosa*" ("*so-osa*").

The third case is about word-final consonant mutation (詞尾輔音變化). Some word-final consonants will be changed if there is no suffix attached. When such a word is suffixed, its final consonant changes back to the original one. Take the word "*qnyutan*" (bite, 咬) as an example. Its root "*qiyuc*" is in fact the result of word-final consonant mutation from its original form (deep root) "*qiyut*". When "*qiyuc*" is attached with a suffix "*-an*" (LV, locative voice, 處所焦點), the final consonant '*c*' changes back to '*t*' and the affixed word is in fact "*q<n>iyut-an*" and the final writing form is "*qnyutan*" (note that the first vowel '*i*' of the root is omitted).

Word inflections in Seediq are overwhelmingly plentiful. In the CIP Seediq dictionary, for example, there are 39 words relating to the same root "*adis*" (bring, 帶走): *desan, dese, desi, deso, desun, dnsanan, dsanan, dsane, dsani, dsanun, dsdesan, dsdesi, dsdesun, knddesi*, *maadis, madis, mdaadis, mkdesun, mkmadis, mnadis, nadis, paadis, pdaadis, pdesan, pdese, pdesi, pdeso, pdesun, pdsanan, pdsane, pdsani, pdsanun, pnaadis, pnadis, pndesan, pndsanan*, *ppaadis, saadis*, and *spaadis*.

3. Seediq Morphological Structure Analysis

The main issue focused in this paper is: when we have a Seediq word and its root word, we want to know its morphological structure, i.e. the combination of prefixes, infixes, and suffixes in that word. In the CIP Seediq dictionary, words and their roots are available. By the techniques developed in this paper, we can generate those words' morphological structures automatically and efficiently.

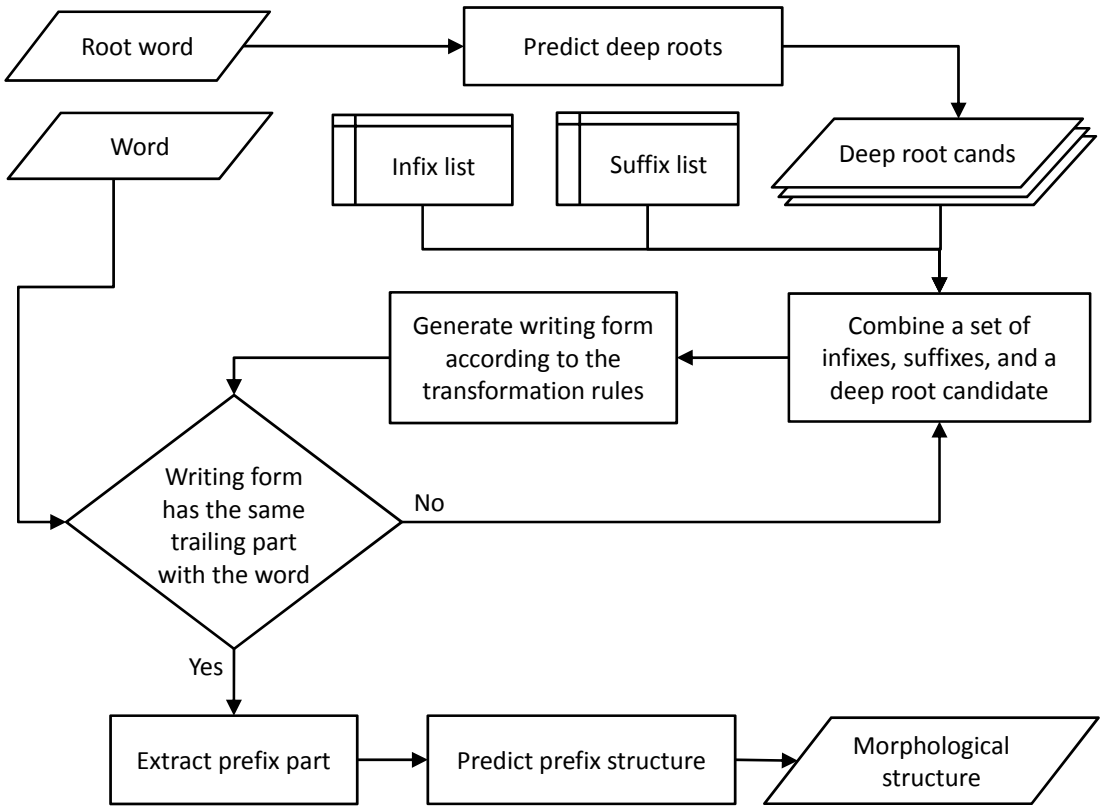

Figure 1. Flowchart of Automatic Seediq Morphological Structure Analysis

Figure 1 demonstrates our proposed flowchart to analyze Seediq morphological structure automatically. Take the word "*pnsltudan*" (whose root word is "*lutuc*") as an example to explain the flowchart. First, a list of deep root candidates {"*lutud*", "*lutuc*"…} of the root word "*lutuc*" is prepared by the method introduced in Section 3.1. (The definition of *deep root* is also given in Section 3.1.) Each deep root candidate is combined with a set (or none) of known infixes and suffixes to form a partial morphological structure (cf. Section 3.2). For example, by selecting a deep root "*lutud*", no infix, and a suffix "*-an*", we will have a partial morphological structure "*lutud-an*". Transformation rules (described in Section 3.3) are then

applied to the partial structure and its writing form "*ltudan*" will be generated (note that the first vowel '*u*' and all the structural symbols are omitted). Since that "*ltudan*" is exactly the trailing substring of the given word "*pnsltudan*", the leading substring "*pns*" is extracted as the prefix part, and its structure "*p<n>s*" is decided by prefix structure analysis methods (as discussed in Section 3.4). Finally, the overall predicted morphological structure of the word "*pnsltudan*" is "*p<n>s-lutuc-an*". Note that we still use root words in the morphological structures, not the deep roots.

3.1 Deep Root Prediction

As discussed briefly above, some root words (原形詞) when suffixed in Seediq will change back to their original forms before vowel neutralization or word-final consonant mutation (cf. Section 2.2). We refer to such original form of a root word as its ***deep root*** (深層原形). For example, when generating writing form of the word "*p-adis-o*", the root word "*adis*" should be replaced with its deep root "*ades*", so that, by omitting neutralized vowels, "*p-ades-o*" becomes the correct writing form "*pdeso*" (bring, 帶).

Table 2 provides more examples of deep roots. All four root words in Table 2 have the same trailing substring "*uk*". However, when they are attached with the suffix "*-i*", the result words (in the third column) do not all end with "*uki*" but change into different trailing substrings. That is because the deep root of "*aduk*" is "*adup*", the deep root of "*ciyuk*" is the same as the root word, the deep root of "*dehuk*" is "*dehek*", and the deep root of "*eluk*" is "*eleb*".

Table 2. Examples of Deep Roots

Root Word	Suffixed Structure	Word	Struct w. Deep Root
aduk (repel, 趕走)	aduk-i	dupi	adup-i
ciyuk (reply, 回覆)	ciyuk-i	ciyuki	ciyuk-i
dehuk (arrive, 到達)	dehuk-i	dheki	dehek-i
eluk (close door or window, 關門窗)	eluk-i	lebi	eleb-i

Predicting deep roots is not an easy task. Neither dictionaries nor syntax books provide information of deep roots. Vowel neutralization or word-final consonant mutation could also be many-to-one mapping. In the following we discuss how we gain a list of deep roots.

Method 1. Inductive Deep Root Prediction

Our first proposed method is based on inductive method. From the CIP Seediq dictionary, we can collect a set of suffixed words referencing to the same root word. The most frequent common trailing substrings among the suffixed words is extracted as its deep root. Note that it

may be identical to the root word, but we are only interested in the transformed deep roots. Details of steps to predict deep roots as well as some examples are given as follows.

Step 1. Collect words referencing to the same root word. In the CIP Seediq dictionary, "參考條目" (cross reference) often provides the root word information. For example, as shown in Section 1.2, 39 words including *desan*, *dese*, *desi*, *etc*., all refer to the same root word "*adis*". Words referencing to the same root word should have the same deep root.

Step 2. Select words with suffixes. Only suffixed words will reveal its deep root, so we need to decide if a word is suffixed or not. Note that the CIP Seediq dictionary does not provide detailed morphological structural information.

If a word ends with its root word, it is not suffixed. For example, both "*ppaadis*" and "*maadis*" end with their root word "*adis*", so there is no suffix in these two words.

If a word ends with its root word after removing possible infixes, it is not suffixed. For example, the word "*cmnebu*" does not ends with its root word "*cebu*". By removing infixes "<*m*><*n*>" after the first consonant '*c*', this word appears exactly the same as its root word and hence not suffixed.

Words other than the two cases above and ending with known suffixes are considered as suffixed words.

Step 3. Predict deep roots by induction. When there is only one vowel in a suffix, the last vowel of the suffixed deep root will not be omitted. We can check if these words end with the same trailing substring and decide the deep root. For example, the structures of the words "*desan*", "*dese*", and "*desi*" are "*ades-an*", "*ades-e*", and "*ades-i*", respectively. After removing suffix parts, they all end with "*es*". Moreover, the preceding consonant '*d*' appears in the root word "*adis*". By replacing the trailing substring of the root word with the most common substring induced from these suffixed words, we can obtain the deep root "*ades*".

Unfortunately, some root words do not have enough related words to induce their deep roots. Moreover, in some rare cases, we found two different deep roots related to the same root word. In order to increase the coverage of deep root prediction and morphological analysis, below we further propose a mapping table for the deep root prediction.

Method 2. Deep Root Mapping Table

The deep root mapping table lists the mapping of trailing substrings between root words and their deep roots. This table is constructed from the <root_word, deep_root> pairs collected by Method 1. For example, the pairs in Table 2 of Section 3.1 tell us that a root word ending with

"*uk*" may have a deep root ending with "*up*", "*ek*", or "*eb*". These mappings are saved in the deep root mapping table. The real data show that "*uk*" maps to "*ek*" for 4 times, "*up*" for twice, and "*eb*" for once.

Since deep roots are closely related to the processes of vowel neutralization and word-final consonant mutation, we only need to consider trailing substrings consist of the last vowel and the word-final consonant. For example, the extracted trailing substring of the word "*aduk*" is "*uk*" and trailing substring of the word "*beebu*" is "*u*".

Figure 2 illustrates the steps of building the deep root mapping table. First, by applying Method 1 to the words in the CIP Seediq dictionary, a set of predicted <root_word, deep_root> pairs are collected. Words in the pairs are then replaced with their trailing substrings. Finally, by counting the mapping pairs, a mapping table is constructed where the mappings are sorted by their frequencies.

When we do not know the deep root of a root word, we can still propose deep root candidates by replacing the trailing substrings according to the deep root mapping table. For example, we know that the root word of "*hligan*" is "*haluy*" but we do not know its deep root. According to the mapping table, the trailing substring "*uy*" often maps to "*ig*". By replacing the trailing substring, we guess that its deep root is "*halig*". The result structure of "*halig-an*" matches indeed with the target word "*hligan*".

To recap, deep root candidates in Figure 1 are generated according to the following order:

(1) The deep root induced from the CIP Seediq by Method 1 (if any)

(2) The original root word

(3) Trail-replacement results according to the deep root mapping table built by Method 2

3.2 Affixation with Infixes and Suffixes

With the list of deep root candidates of the root words, we then move to the next step. Each deep root candidate will be combined with known infixes and suffixes for string matching in the next steps. Infixes and suffixes are first considered because their sets are rather fixed; we only see 3 kinds of infixes {"<*m*><*n*>", "<*m*>", "<*n*>"} and 10 kinds of suffixes {"*-an-an*", "*-an-un*", "*-ane*", "*-ani*", "*-ano*", "*-an*", "*-un*", "*-e*", "*-i*", "*-o*"} in the training set. Note that an infix appears after the first consonant. For example, when the word "*quyux*" is infixed with "<*m*>", it becomes "*qmuyux*" ("*q*<*m*>*uyux*") (raining, 下雨). But if a root word starts with a vowel, the infix appears at the beginning of the word. For example, when the word "*apa*" is infixed with "<*n*>", it becomes "*napa*" ("<*n*>*apa*") (carry, 揹). For convenience, we leave the infixes in such cases together with the prefix part (extracted in Section 3.3) to be processed in Section 3.4.

3.3 Transformation Rules to Generate Writing Form

For the case when a root word is affixed with only prefixes and infixes, its writing form can be derived directly from the combination by removing structural symbols. For instance, "*m-p-k-beyax*" becomes "*mpkbeyax*" (work hard, 努力) and "*h<m>aduc*" becomes "*hmaduc*" (send, 送).

But when a root word is suffixed, two cases should be considered. The first case is vowel reduction (元音脫落) where vowels other than the last two are neutralized and omitted. For example, "*hetur-ani*" (block out, 擋) becomes "*htrani*" where the first two vowels '*e*' and '*u*' are omitted. The rule of vowel reduction can be applied by programs easily.

One exception of vowel reduction happens when only one of the two adjacent identical vowels is about to be omitted. In such case, both vowels will not be omitted. Take "*osa-an-un*" (go, 去) as an example. According to the general rule of vowel reduction, both vowels '*o*' and '*a*' in the root word "*osa*" should be omitted. However, the second vowel '*a*' of the root word "*osa*" is followed by the suffix "*-an*" which starts with the same vowel. Therefore, the second vowel '*a*' is not omitted, and the final writing form becomes "saanun" where only the first vowel '*o*' is omitted.

The second case is the addition of '*y*' or '*w*'. We find some cases that a '*y*' or '*w*' is added between the root word and the suffix. For example, the final writing form of "*chungi-an*" is "*chngiyan*" (forget, 忘記) and the final writing form of "*cebu-an*" is "*cbuwan*" (to be shot, 被擊中). We have not figured out the rules for such cases. Currently we simply insert a '*y*' or '*w*' to see if the transformation result matches the final writing form.

The complete transformation rules to generate the final writing form are defined as follows. Given a morphological structure represented as *pfx-root<ifx>str-sfx*, *rootstr* is the root part (root word or deep root), *pfx* is the prefix part, *ifx* is the infix part, and *sfx* is the suffix part. Any affix part may be empty. The writing form of the morphological structure is generated by the following steps:

Step 1. When the suffix part is not empty, the last two vowels in the structure remain unchanged. Vowels other than the last third one are all omitted. As for the last third vowel,

a) If it is the same as the last second vowel and they are adjacent to each other, the last third vowel remains unchanged

b) Otherwise the last third vowel is omitted

Step 2. If the suffix *sfx* starts with a vowel but is different from the word-final vowel of rootstr, one '*y*' or '*w*' may be inserted between them to generate a correct writing form.

Step 3. Remove all morphological structural symbols (including -, <, and >).

An interim summary for Section 3.1 to Section 3.3: Given a Seediq word and its root word, a list of deep root candidates is generated by methods proposed in Section 3.1. Each deep root candidate is then combined with every infix and suffix (including empty strings) as described in Section 3.2. Each combination is then transformed into the writing form by rules explained in this Section 3.3. If this writing form matches the trailing substring of the target Seediq word, this combination of deep root, infix and suffix is proposed as the predicted morphological structure, and the unmatched part is extracted as the prefix part for further analysis by methods proposed next in Section 3.4.

3.4 Prefix Structure Analysis

The prefix structure analysis also encounters ambiguity problem. One prefix string can be segmented into several different prefix combinations. For example, the prefix string "*kn-*" can be either "*kn-*" (NMLZ, nominalization, 名物化) or "*k<n>-*" (STAT<PFV>, 靜態<完成貌>), and the prefix string "*sk-*" can be either "*sk-*" (deceased, 已故) or "*s-k-*" (existential-STAT, 有-靜態).

To our best effort, we so far cannot find much information about prefix combinations. To solve the prefix problem in Seediq, we here propose several approaches similar to the classical solutions for Chinese word segmentation, including probability models and machine learning, which will be discussed in details below. Our goal is to find the best system in which we can predict the morphological structures of words in the CIP Seediq dictionary with high accuracy in order to reduce the effort of human checking in the future.

First of all, we need to prepare a list of atomic prefixes. There are 29 atomic prefixes found in the Seediq syntax book, including {"*k-*", "*n-*", "*kn-*", "*m-*"…}. We further found 10 different atomic prefixes in the test data, including {"*de-*", "*gn-*", "*km-*"⋯}. The following experiments are based on these atomic prefixes. We do not know whether there will be more new atomic prefixes in the CIP Seediq dictionary or not.

Reduplication (introduced in Section 2.2) also appears in the prefix part. It is used to emphasize the amount of something or the intensity of an action. It can be attached to a root word or an atomic prefix. It repeats either the first consonant (e.g. "*s-*" in "*s-seediq*" and the first "*p-*" in "*p-p-heyu*"), or the first consonant with the word-initial vowel (e.g. "*le-*" in "*k-le-eluw*"), or the first two consonants (e.g. "*kr-*" in "*m-kr-kere*").

During training, all reduplication prefixes are replaced with a special symbol and treated as one type of the atomic prefixes. Therefore, there are totally 40 types of atomic prefixes in the experiments in Section 4. When segmenting a prefix string, a segment matching any of the 3 reduplication cases shown in the previous paragraph is considered to be a reduplication prefix.

Probability Models

One common approach for Chinese word segmentation is to build probability models. In a similar way, we propose unigram and bigram models for prefix structure analysis in Seediq. Given a prefix string *pfx* and one of its segmentation $x_1x_2\ldots x_m$ where x_i is an atomic prefix, the probability of this segmentation is defined as follows, where \$ denotes the beginning of the prefix string.

$$\text{Unigram model: } P(pfx) = \prod_{i=1}^{m} P(x_i) \quad (1)$$

$$\text{Bigram model: } P(pfx) = P(x_1|\$) \prod_{i=2}^{m} P(x_i|x_{i-1}) \quad (2)$$

Because the amount of training data is not large enough, we still need to apply smoothing methods to avoid zero probabilities. But some well-known smoothing methods such as Witten-Bell or Good-Turing are good for large training data. We did not choose them in this paper. Instead, we use Lidstone smoothing to build our unigram model. That is, the frequency of each atomic prefix (seen or unseen) is added with a value λ before building the probability model. Let *N* be the original sum of the frequencies of all atomic prefixes and *B* be the number of types of atomic prefixes. Lidstone smoothing will assign a probability of λ / (*N*+*B*λ) to each unseen atomic prefix.

We use back-off approach to deal with zero probabilities in the bigram model. That is, we consider the unigram probability (weighted by an α value) of the second prefix in an unseen bigram. When $P(x|y)=0$, we use $P(x|y)=\alpha P(x)$ instead.

The unigram model provides the probabilities of 40 atomic prefixes. The bigram model provides the probabilities of bigram of these 40 prefixes and the starting sign \$ (thus 41×40 types of bigrams). An unknown prefix x_i or a bigram containing such an unknown prefix has no probability. Smoothing is designed for known but unseen atomic prefixes in our work.

The steps of prefix structure analysis are as follows. Given a prefix string *pfx*, all segmentations $x_1x_2\ldots x_m$ are enumerated by inserting one or zero '-' between any two adjacent letters. For example, the prefix string "*mss-*" can be segmented into {"*mss-*", "*m-ss-*", "*ms-s-*", "*m-s-s-*"}. The segmentation having the best probability is selected as the final answer. Note that the strings "*mss-*" and "*ss-*" do not appear in the list of atomic prefixes and thus have no probability; so the probability of "*mss-*" and "*m-ss-*" is also 0.

Machine Learning and Deep Learning Methods

Machine learning methods are also tried to guess the prefix structure. However, we have too little features so far, and the only features we know are contextual information and the list of atomic prefixes. More useful features need to be discovered in the future. The following example illustrates the features of each letter in the prefix string "*psq-*" where its correct structure is "*ps-q-*".

c	c_{-2}	c_{-1}	c_1	c_2	[B	E	S]	Class
p	$	$	s	q	1	0	1	B
s	$	p	q	$	1	1	1	E
q	p	s	$	$	1	0	1	S

The feature c_k denotes the letter in the context. The Boolean features [B E S] denotes the position where this letter appear in the atomic prefixes. For example, the [B E S] values of the letter '*p*' are [1 0 1] because it appears in the beginning (B) of some atomic prefixes {"*pn*-", "*ps*-"…}, and it can be a single-letter prefix "*p*-" itself (S). E means appearing the end of an atomic prefix. Note that no atomic prefix is longer than 2 letters in our datasets. The final classification is also one of the BES labels.

In addition, deep learning methods such as the encoder-decoder model are explored. The input is the prefix string where letters and the symbol '-' are denoted by one-hot encoding. The output is the prediction of morphological structure.

4. Experiments

4.1 Experimental Datasets

The first dataset comes from the Seediq syntax book "賽德克語語法概論". There are 509 sentences provided as illustrations in this book. The morphological structures of words in the sentences are also provided. There are 817 distinct Seediq words appearing in the sentences and 394 of them contain affixes. We took these 394 affixed words as the training data.

The second dataset comes from 515 new sentences provided by Dr. Li-May Sung, the author of the Seediq syntax book and one of the authors of this paper. These sentences are also tagged with morphological structures. 322 new Seediq words with affixes are extracted from these sentences as the test data.

4.2 Infix and Suffix Detection Experiments

Sections 3.1 ~ 3.3 propose approaches to detect deep root, prefix, infix, and suffix parts in a given Seediq word (in which the structure inside the prefix part has not been predicted). Table 3 lists the performance of these approaches, where precision is the percentage of system-detected units (words or affixes) being correct, and recall is the percentage of gold-standard units being detected by the system.

Table 3. Performance of Infix and Suffix Detection

	Training Data					**Test Data**				
Unit	**Gold**	**System**	**Correct**	**P (%)**	**R (%)**	**Gold**	**System**	**Correct**	**P (%)**	**R (%)**
Word	394	357	353	98.88	89.59	322	286	278	97.20	86.34
Infix	79	77	77	100.0	97.47	55	47	47	100.0	85.45
Suffix	169	135	135	100.0	79.88	127	98	98	100.0	77.17
Prefix	221	207	203	98.07	91.86	194	186	180	96.77	92.78

For more details, the third row of Table 3 shows that 79 of the 394 words in the training set contain infixes, where 77 of them can be detected by the system (recall 77 / 79 = 97.47%) and all of them are correct (precision 77 / 77 = 100%). All precision scores of prefix, infix, and suffix detections are around 98% to 100%. Recall scores are a little lower, because 37 of the 394 affixed words are exceptions of morphological rules.

4.3 Prefix Structure Analysis Experiments

In the training set, only 221 words are prefixed as shown in Table 3. 116 of them are prefixed by one single-letter prefix and hence no further analysis is needed. Therefore, the training set of prefix structure analysis contain only 105 words whose prefix parts are longer than one letter. When evaluating on the training set, we adopt leave-one-out cross-validation method due to the small amount of data. Each word is predicted by the classifier trained with the other 104 words.

Table 4. Performance of Prefix Analysis by Unigram Models

	Training Data			**Test Data**		
λ	**Word**	**Correct**	**A (%)**	**Word**	**Correct**	**A (%)**
0	**105**	**86**	**81.905**	103	61	59.223
0.1	**105**	**86**	**81.905**	103	64	62.136
0.3	105	85	80.952	103	65	63.107
0.5	**105**	**86**	**81.905**	103	69	66.990
1	105	85	80.952	103	71	68.932
2	105	81	77.143	103	83	80.583
3	105	79	75.238	103	86	83.495
4	105	81	77.143	103	86	83.495
5	105	79	75.238	**103**	**87**	**84.466**

As for the testing set, there are 194 prefixed words in the test set and 103 of them have prefixes longer than one letter. The metric of evaluation is accuracy. The prefix structure prediction has to be exactly the same as the gold standard to be counted as "correct".

The experimental results of unigram models with different λ values are shown in Table 4. The λ value does not affect much performance on the training set. It means that most unseen prefixes only appear in the test set. Interestingly, when λ value is set to be 3 or larger, the performance on the test set is improved in a great degree. It seems to hint that we need a training set where each atomic prefix should appear at least 3 times.

The experimental results of bigram models with different λ values are listed in Table 5. Again, the λ value does not affect much performance on the training set, but improves the performance on the test set a lot when it is set to be 2 or larger.

Table 5. Performance of Prefix Analysis by Bigram Models with Different λ

		Training Data			Test Data		
λ	**α**	**Word**	**Correct**	**A (%)**	**Word**	**Correct**	**A (%)**
0	0.7	105	82	78.095	103	64	62.136
0.01	0.7	105	81	77.143	103	67	65.049
0.1	0.7	105	81	77.143	103	67	65.049
0.2	0.7	105	81	77.143	103	68	66.019
0.3	0.7	105	82	78.095	103	68	66.019
0.4	0.7	105	82	78.095	103	68	66.019
0.5	0.7	105	83	79.048	103	68	66.019
0.6	0.7	105	83	79.048	103	68	66.019
1	0.7	105	82	78.095	103	70	67.961
2	0.7	105	84	80.000	103	87	84.466
3	0.7	**105**	**87**	**82.857**	103	88	85.437
4	0.7	105	82	78.095	**103**	**89**	**86.408**
5	0.7	105	80	76.191	103	87	84.466
6	0.7	105	81	77.143	103	87	84.466
7	0.7	105	82	78.095	103	87	84.466
8	0.7	105	80	76.191	103	87	84.466

The experimental results of bigram models with different α values are shown in Table 6. Comparing the first system (where α = 0) with the others, we can see that back-off method does improve the performance. However, the α value does not affect the performance much.

Parameters in the best system are λ = 3 and α = 0.7, which achieves an accuracy of 82.86% on the training set and 85.44% on the test set.

Table 6. Performance of Prefix Analysis by Bigram Models with Different α

		Training Data			Test Data		
λ	**α**	**Word**	**Correct**	**A (%)**	**Word**	**Correct**	**A (%)**
3	0	105	80	76.191	103	83	80.583
3	0.1	105	86	81.905	103	87	84.466
3	0.4	105	86	81.905	**103**	**88**	**85.437**
3	0.7	**105**	**87**	**82.857**	**103**	**88**	**85.437**
3	1	105	86	81.905	**103**	**88**	**85.437**

Machine learning and deep learning methods described in Section 3.4 are also tested in this paper. Many well-known classifiers including Naïve Bayes, SVM, and decision tree are tried, and an encoder-decoder system by LSTM is also constructed. But unfortunately, the best accuracy is only 52.06%. The training set is too small for machine learning and deep learning at this stage.

4.4 Final Remarks

In general, our infix and suffix detection system can successfully predict structures for nearly 90% of words. It will greatly reduce the human effort needed to construct a larger dataset from the CIP Seediq dictionary. In our preliminary observation, only 335 of the 5,600 words in the CIP Seediq dictionary cannot be predicted.

Error analysis indicates that some words are inflected in an exceptional way. For example, the word "*kesa-un*" is "*kesun*" (do this way, 這樣做), but our system incorrectly predicts it as "*ksaun*"; and the word "*p-uqi-un*" is "*puqun*" (eat, 吃), but our system incorrectly predicts as "*puqiun*" or "*puqiyun*". A list of exceptional words should be constructed in the future.

As for our prefix analysis system, it can successfully analyze structures for around 83% of prefixed words. Again, it will greatly reduce the human effort in the future. However, it is not easy to improve the performance of the prefix structure analysis system. To solve the ambiguity problem (such as "*kn-*" vs. "*k<n>-*"), we might need the semantic information of the prefixed word or even the information about its functionality in that sentence. This will also be explored in the near future.

5. Conclusions

This paper proposes approaches to analyze morphological structures of Seediq words automatically. The experimental datasets contain 716 affixed Seediq words with their morphological structures.

Morphological analysis starts from the infix and suffix detection. Deep root candidates generated by our proposed methods are combined with known infixes and suffixes. The writing form of the combination is then generated by the transformation rules. If the writing form matches the trailing substring of the target word, this combination is selected as the result of infix and suffix detection. This approach achieves a precision of 98.88% and a recall of 89.59%.

Prefix structure analysis is treated similar to the word segmentation problem and predicted by probabilistic models. Zero probability problem in the bigram model is solved by the back-off approach, i.e. using the unigram probability weighted by α instead. Zero probability problem in the unigram model is solved by the Lidstone Smoothing. i.e. adding λ to frequencies of unigrams. We conclude that the best system is based on bigram model where $\lambda = 3$ and $\alpha = 0.7$, with an accuracy of 82.86%.

In the future, we would like to apply the techniques developed in this paper to analyze the 5,595 words in the CIP Seediq dictionary to create a larger dataset and build a more reliable probabilistic model. Moreover, if the morphological structures of all words appearing in the 6,019 exemplar sentences in the CIP Seediq dictionary are available, it will be possible to build a large bilingual corpus for machine translation then.

Acknowledgement

This research was funded by the Ministry of Science and Technology in Taiwan (Grant: MOST 109-2221-E-019 -053 -).

References

Bahdanau, D., Cho, K.H., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.

Chiang, C.-Y. (2018). Cross-Dialect Adaptation Framework for Constructing Prosodic Models for Chinese Dialect Text-to-Speech Systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(1), 108-121. doi: 10.1109/TASLP.2017.2762432

Gu, H.-Y., Zhou, Y.-Z., & Liau, H.-L. (2007). A System Framework for Integrated Synthesis of Mandarin, Min-Nan, and Hakka Speech. *International Journal of Computational Linguistics & Chinese Language Processing*, *12*(4), 371-390.

Hsieh, S.-K., Su, I-L., Hsiao, P.-Y., Huang, C.-R., Kuo, T.-Y., & Prévot, L. (2007). Basic Lexicon and Shared Ontology for Multilingual Resources: A SUMO+MILO Hybrid Approach. In *Proceedings of OntoLex Workshop in the 6th International Semantic Web Conference*.

Iunn, U.-G., Lau, K.-G., Tan-Tenn, H.-G., Lee, S.-A., & Kao, C.-Y. (2007). Modeling Taiwanese Southern-Min Tone Sandhi Using Rule-Based Methods. *International Journal of Computational Linguistics & Chinese Language Processing*, *12*(4), 349-370.

Li, P. J.-K. (1977). Morphophonemic Alternations in Formosan Languages. *Bulletin of the Institute of History and Philology (中央研究院歷史語言研究所集刊)*, *48*(3), 375-413. doi: 10.6355/BIHPAS.197709.0375

Li, P. J.-K. (1981). Reconstruction of Proto-Atayalic Phonology. *Bulletin of the Institute of History and Philology (中央研究院歷史語言研究所集刊)*, *52*(2), 235-301. doi: 10.6355/BIHPAS.198106.0235

Li, P. J.-K. (1991). *Vowel Deletion and Vowel Assimilation in Sediq*. In Papers on Austronesian languages and ethnolinguistics in honour of George W. Grace, Pacific Linguistics C-117, 163-169.

Lin, C.-J. & Chen., H.-H. (1999). A Mandarin to Taiwanese Min Nan Machine Translation System with Speech Synthesis of Taiwanese Min Nan. *International Journal of Computational Linguistics & Chinese Language Processing*, *4*(1), 59-84.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1412-1421. doi: 10.18653/v1/D15-1166

Rau, D. V., Wu, Y.-H., & Yang., M.-C. (2015). A Corpus-Based Approach to the Classification of Yami Emotion. *New Advances in Formosan Linguistics, Asia-Pacific Linguistics*, 533-554.

宋麗梅(2011)。「**原住民族語言字詞典編纂四年計畫—第 3 階段計畫**」(**賽德克語**)。新北市:原住民族委員會。[Sung, L.-M. (2011). *Revitalization of Formosan Languages: Compilation of Seediq Dictionary*. New Taipei City, Taiwan: Council of Indigenous Peoples.] 2009/8/3-2011/8/2.

宋麗梅(2018)。**臺灣南島語言叢書 5: 賽德克語語法概論**(2 版)。新北市:原住民族委員會。[Sung, L.-M. (2018). *A Sketch Grammar of Seediq, Formosan Series #5, 2018* (2nd Edition). New Taipei City, Taiwan: Council of Indigenous Peoples.]

楊秀芳(1976)。賽德語霧社方言的音韻結構。**中央研究院歷史語言研究所集刊**，**47**(4)，611-706。[Yang, H.-F. (1976). The Phonological Structure of the Paran Dialect of Sedeq. *Bulletin of the Institute of History and Philology*, *47*(4), 611-706.]

Yang, M.-C. & Rau, D. V. (2011). Constructing a Yami Language Lexicon Database from Yami Archives. In *Proceeding of the 2011 TELDAP (Taiwan e-Learning and Digital Archives Program) International Conference*.

Yang, M.-C., Rau, D. V., & Chang, A. H.-H. (2011). A Proposed Model for Constructing a Yami Wordnet. *International Journal of Asian Language Processing*, *21*(1), 1-14.

楊孟蒨、何德華(2015)。**建構台灣原住民語自然語言處理技術探討與研究**。科技部計畫期末報告(編號: MOST 103-2221-E-126-008-) [Yang, M.-C. & Rau, D.V. (2015). *Exploring the NLP Techniques for Formosa Indigenous Languages*. (MOST 103-2221-E-126-008-), 2014/8~2015/7.

Yu, M.-S. & Lin, Y.-J. (2012). The Polysemy Problem, an Important Issue in a Chinese to Taiwanese TTS System. *International Journal of Computational Linguistics & Chinese Language Processing*, *17*(1), 43-64.