The 2016 Conference on Computational Linguistics and Speech Processing
ROCLING 2016, pp. 284-298

Design of an Input Method for Taiwanese Hokkien using Unsupervized Word Segmentation for Language Modeling

Pierre Magistry

國立成功大學台灣語文測驗中心

pierre@magistry.fr

Abstract

This paper presents the challenges and the methodology followed in the design of a new Input Method (IME) for the Taiwanese (Hokkien) language. We first describe the context, the motivations and some of the main issues related to the input of text in Taiwanese on modern computer systems and mobile devices. Then we present the available resources which our system is based on. We will describe the whole architecture of our system. But since the cornerstone of modern IME is the Language Model (LM), the main Natural Language Processing issue on which we will focus in this paper is the estimation of a LM in the case of this under-resourced language. The solution we propose to rely on unsupervised word segmentation which preserves some degree of ambiguity.

Keywords: Unsupervized Word Segmentation, Language Modeling, Input Method, Taiwanese

1. Introduction

Taiwanese Hokkien (or simply "Taiwanese," Tâi-gí 台語, throughout the rest of the paper) is a language spoken by a vast majority of the Taiwanese people. It is a Sinitic language of the Minnan (bân-lâm-gí, 閩南語) group. Since our work is based on readily available resources which describe the variety in use in Taiwan, it is better fitted for Taiwanese, but it may be useful to more than 60M people in and outside Taiwan who speak closely related variants.

Although this language is still widely spoken in Taiwan, Taiwanese has never been the official language, the efforts in standardization and institutionalization only started in the last decades.

Even without state-run institutionalization, written Taiwanese has been in use in printed and handwritten documents for centuries. Depending on the situation, different scripts have been used, including Hàn characters (Hàn-jī), Latin alphabet, adapted versions of Japanese kana and Zhuyin fuhao (注音符號). Nowadays, Hàn-jī and Latin are the two scripts which cover the vast majority of produced texts. Zhuyin is mostly used for annotation of rare Hàn characters or in teaching materials, and for code-mixing in spontaneous writing.

Texts written using the Latin script can be divided between different Romanization types, the two more important which are encountered in our resources are Pe̍h-ōe-jī (POJ) and Tâi-lô ("Taiwan Romanization System", hereafter TRS). The first one is also called "Church Romanization" due to its origin in missionary works and the latter is recommended by Taiwan's Ministry of Education since 2006.

As a result, the actual situation of written Taiwanese is an interesting case of poly-orthography where one scripter can choose between Hàn-jī and Romanization or (more frequently) mix the two scripts. This requires some specific features from an IME.

In the past decades, the status of Taiwanese at school has changed from being forbidden to being taught in classes of "Mother Languages" in primary schools. However the curriculum is still very limited and even if a large majority of Taiwanese people can speak the language, only a very small proportion is actually literate in Taiwanese. However, almost all Taiwanese are familiar with Hàn-jī and Zhuyin phonetic transcription (taught to be used for writing Mandarin down).

This recent history also leads to a very limited place for Taiwanese in the computing world and this language is usually neglected by computer software designers (even its ISO code 'nan' is very rarely recognized as an option). In addition to the various political and sociolinguistics factors which may lead some to consider Taiwanese Hokkien as an endangered language, we want to stress the impact of the ease to use a language on modern devices. The possibility and the convenience to input texts seem to us to be of first importance to ensure language preservation. This is especially the case for Taiwanese as modern technologies are omnipresent in Taiwan and an important part of language use among

Taiwanese people is made online.

For more details about the history of written Taiwanese, interested readers may refer to [1]. For an overview of the current state of Taiwanese text processing, one can refer to [2].

Multiple Input Methods (IME) have already been developed for desktop computers by different organizations over the years, the most noticeable being probably the FHL's Taigi IME[1] to type in POJ and 吳守禮臺語注音輸入法[2] to type in Zhuyin. The Ministry of Education also provides an IME for desktop computers[3]. More details about available IMEs can be found in [2] (p. 144).

As mobile devices progressively took the largest share of online communications, IMEs for Taiwanese did not follow and no IME was available on mobiles until very recently (2014 for our own first try on Android[4] and 2016 for iOS[5]). We believe that not only such softwares are crucially needed, but they also have to catch up with state-of-the-art Mandarin IME. For now, they are still behind in terms of functionalities, performance and convenience to be adopted by a large number of users (who are typically bilingual with Mandarin). There is still a long way to go.

Our objective is thus to design an IME for Taiwanese on mobile devices which would benefit from modern NLP techniques. To do so, we need efficient Language Models (LM) to provide smarter candidate ranking and prediction. LMs are the cornerstone of modern IMEs for such features. However, unlike Mandarin, Taiwanese lacks of linguistically annotated resources such as segmented corpora to train word-level models. This pushed us to look for unsupervised solutions to be able to benefit from (raw) language corpora without the need for costly and time consuming manual annotation. In this paper, we will present the core architecture of our IME, with a special focus on how we address word segmentation to train the LM needed for input prediction.

1 See http://taigi.fhl.net/TaigiIME/
2 See http://xiaoxue.iis.sinica.edu.tw/download/WSL_TPS_IME.htm
3 See http://depart.moe.edu.tw/ED2400/cp.aspx?n=BB47AA61331DDAC8
4 See https://play.google.com/store/apps/details?id=fr.magistry.taigime
5 See https://itunes.apple.com/tw/app/id1080190324

In the next Section we will sum up the specificities of our task. We will then present the resources we used to train our models and build the IME in Section 3. In Section 4 we describe the whole architecture and our main design choices and in Section 5 we focus on the word segmentation and the language modeling part. We finally conclude with a description of some functionalities that are still to be implemented to provide a more appealing and efficient IME.

2. Specific Constraints for a Taiwanese IME

In the introduction, we sketched the unique situation of written Taiwanese, these observations lead us to define a number of constraints and goals we set for ourselves.

2.1 Taking into Account the Diversity of Scripts

As the users are likely to have different habits in the selection of the script, we have to allow for a large spectrum of possibilities. It is important to stress that the same user may want to use different scripts for different genres of documents. For example, one may be willing to use hàn-jī to write poetry but prefer POJ when chatting online.

A related issue is the choice of phonetic input given to the system. Romanization is a natural candidate as it is both a transcription and an orthography, but many potential users are not literate in POJ or TRS. On the other hand, everyone in Taiwan is used to the Zhuyin system to transcribe the sounds of Mandarin. This transliteration system was first designed one hundred years ago for Mandarin but was extended in the 1940s to cover other Sinitic languages such as Taiwanese. It is now part of the norm ISO 15924 and included in the Unicode standard. This fact is often ignored by users of Zhuyin, but only a subset of the symbols need to be learned by native speakers to complete the set of symbols used for Mandarin and enable them to write, almost as easily as they speak. However the Zhuyin is not directly used in formal documents (it is more a transcription system than a script) where mixed script is essentially in Hàn-jī and Latin. As a result, we shall also provide both Hàn-jī and Romanization output for input in Zhuyin. We believe it may even be a way to help the users learn the Romanization.

To sum things up, input has to be allowed either in Zhuyin, POJ or TRS and conversion is provided into Hàn-jī or Romanization.

2.2 Privacy and Security

The Input Method is a very sensitive component in a computer system, as it sees and controls everything the user is writing. It is a position of choice for spyware or other kind malware. To prevent security risks and allow users to trust our software, we choose not to require Internet access permission for the software. This is a special feature of Android that tells the Operating System to forbid any attempt by the IME to communicate over the network.

This design choice has a heavy cost to compete with other systems as it prevents us to crowdsource any data directly from all the user inputs and to provide an online language model that may evolve as other users use the system. We will mention some possible solutions we plan to experiment to get users actively involved in the evolution of the software database and statistical models.

2.3 Taiwanese Hàn-jī （台語漢字）

Some of the Hàn-jī used to write in Taiwanese are specific to this language and are not used for Mandarin. Unfortunately, these are typically absent from OSes's fonts, especially on mobile devices. It is possible to include a font within the package to be installed along and to be used by the UI of the IME. However we cannot enforce its use by other applications so we cannot guarantee that all the characters will be correctly displayed after selection. There is no obvious and user-friendly solution to this issue which is a limitation at the OS level. The only workaround we can think of is to provide an online text editing platform as a website or independent APP. Such software can specify the correct font to have a nice editing environment but this won't fix the OS and the display in other applications.

This issue would be better addressed by Google or by mobile phone constructors.[6]

3. Resources

As we mentioned in the Introduction, we do not have annotated training corpora at hand. However, recent years have seen the development of many resources which are of great importance for our work. Many of them are distributed as Open Data. Without this, our

6 And many constructors are indeed Taiwanese !

contribution would simply be impossible. The resources we rely on can be divided into lexicons and corpora.

3.1 Lexicons

In 2008, The Ministry of Education launched the online 「臺灣閩南語常用詞辭典」[3]. Later, it was decided to release the data under a permissive license (Creative Common CC-BY-ND). This alone was the starting point for my first Input Method on Android. This dictionary contains more than 25,000 entries with pronunciation in TRS, definitions in Mandarin, grammatical information, example sentences and regional variations.

Later, we were also provided with a reference word list of more than 8000 entries with pronunciation in TRS and translation in Mandarin, aligned on the levels defined in the Common European Framework of Reference for Languages (CEFR) [4]. This list was compiled by the Center for Taiwanese Languages Testing (CTLT) at National Cheng Kung University [5]. The valuable CEFR alignment is not used yet but will be important to address the literacy issue more adequately.

We are also in the process of integrating data collected during the digitizing of the Mandarin-Taiwanese dictionary「國台對照活用辭典」[6] authored by Prof. Ngôo Siú-lé (吳守禮). As the right holders decided to make it available online under a permissive license and seek the help of the Wikimedia foundation and G0V-tw to face the technical issues. The main goal is to make the dictionary available on Wikisource but once properly structured, the data can also benefit to other projects, including ours.

Finally, the word list used in the FHL IME for desktop computers has also be made available under Creative Commons license by Tân Pektiong (陳柏中) and 林哲民 (Lin Zhemin) [7]. It is a very large word list with 160k entries with Hàn-jī and pronunciation.

3.2 Reference Corpus

In order to estimate a language model for our candidates ranking and input prediction features, we needed a large corpus written in Taiwanese. For this part we benefited from the

results of a NSC project lead by Prof. 楊允言 (Iûn Ún-giân) [8] aiming at compiling a reference corpus for modern written Taiwanese. This is a vital resource for us, but it comes as a raw corpus in plain text without any annotation. It requires some pre-processing and word segmentation to be useful for our goal. This corpus contains close to 9M syllables and is divided into two parts, one is written in POJ and the other is in mixed Hàn-jī and POJ. For the moment, we only use the latter one in this work.

4. System Description

In this paper, we don' t address all the GUI and user interaction aspects of the project. Those are less relevant for Computational Linguists and more specific to the Android SDK. For our concerns, an input method is essentially a function that turns an input **I** and a context **C** into an ordered list of transliteration candidates **T**.

$$\text{IME: } I \times C \rightarrow T$$

Where **I** and **C** are two Strings and **T** is a List of Strings with a score.
I can be any input as valid POJ, TRS or Zhuyin.
C is expected to be some Hanlo (mix of Hanji and one Romanization)
and strings in **T** are in Hàn-jī, TRS or POJ

The Input can be null, if it is so, depending on previous user actions the system will either try to guess the next word or to suggest other alternatives for the previous word.

To deal with the various scripts used to transliterate the sounds of Taiwanese, we convert them to an internal representation using the International Phonetic Alphabet (IPA) as a basis for this step of normalization.

4.1 Language and licensing Choice

Until recently, only Android provided an API to create third-party IME. We thus naturally started with Android. The fact that Android APPs all run into a JVM also allows us to write a large part of our code in a cross-platform way, and to use it on a desktop for data pre-processing or system evaluation. It will also enable us to easily provide a Web version of the IME in the future. To speed up development and enable us to easily share code between

Android, web server and web client (after compilation in JavaScript), we choose to write everything in Scala.

To ensure the continued existence of the software, we release it under an open source license (AGPL3). The Source code can be found by following this link: https://github.com/a-tsioh/TaigIME2

4.2 Architecture

As far as the LM and conversion function are concerned, the global architecture of our software split into two different processing pipelines (which share a large amount of code). The first one is the data preparation and the estimation of a LM described in Section 4.3 (to be run on a desktop computer). The second one described in Section 4.4 is the use of the LM and the database to make predictions (run on Android devices).

4.3 From Raw Corpus data to Language Model

The first stage of data processing aims at preparing the Language Model data, the whole process is illustrated on Figure 1 and described in this Section.

After some preliminary experiments with syllable-based Language Models, we decided to turn to word-based Language Model. Such LM is likely to give more relevant insights but requires a step of Word Segmentation prior to language modeling.

Word Segmentation is a typical task for written Chinese processing. Just like Mandarin, when using Hàn-jī Taiwanese is written without word boundaries (except some punctuation marks). However, it does mark word boundaries with spaces when romanized. Hence it marks some but not all boundaries when written in mixed Hàn-lô.

Closely related to the issue of Chinese Word Segmentation (CWS) is the issue of Multi-Word Expressions (MWE). Although they seem to be addressed separately by two distinct communities of researchers, we believe that these two NLP issues are two aspects of the same linguistic question regarding the definition of the units of language analysis. In the specific

applied case of designing an Input Method, we want to segment "words" but we are also interested in predicting and suggesting larger MWE. Users are very likely to input and expect such larger units from our predictions (untrained native speakers don't perform very well at the CWS task, the pervasiveness of MWE is one possible explanation to this observation).

4.3.1 Tokenization

Our previous works in unsupervised CWS[7] have shown us that the quality of the initial tokenization may have an important impact on the quality of the segmentation. For Mandarin, focus was made on distinguishing between various scripts (Latin, numbers, Chinese, punctuation...) and on spotting some kind of regularly formed named entities (addresses, dates…)

In the case of Taiwanese, we also have to deal specifically with the mixed scripts. When encountering Latin characters, we must also decide whether it is POJ/TRS or a foreign word. Then, we consider each romanized Taiwanese words as a single token. We use regular expressions to do this. In ambiguous cases, we favor the Romanization hypothesis.

4.3.2 Segmentation

To segment the corpus into words prior to language model training, we use two different strategies:

a) A classic Maximum Matching based on our aggregated word list. This is to ensure that words from our lexicon have been seen by the model if present in the corpus.

b) An unsupervised Word Segmentation System which relies only on the raw data to statistically segment the text into words. This allows us to catch words and frequent phrases (MWEs) which are missing in the word list.

After the raw text has been turned into a sequence of tokens, we can start the unsupervised segmentation. We use the system ELeVE presented in Magistry & Sagot [9] which is now available off-the-shelf at https://github.com/kodexlab/eleve. For training and we propose a modified version of the decoding algorithm implemented in Scala for our IME, which keeps some ambiguity in the segmentation output.

7 Which was the topic of my Ph.D. dissertation [11]

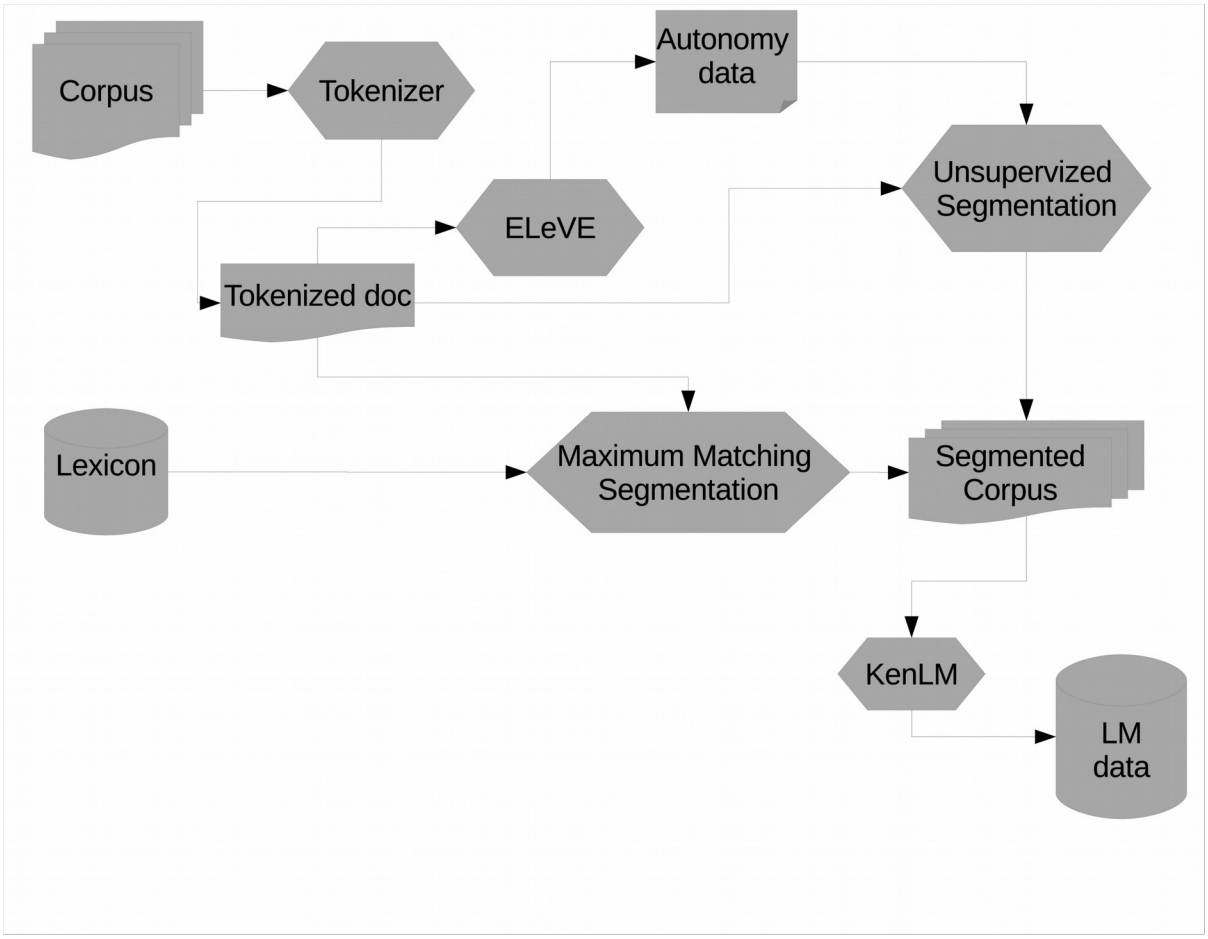

Figure 1: From Raw Text to Language Model

More details on this step are given in Section 5. We then add the lexicon-based segmentation to the list of unsupervized segmentation solutions. We obtain a corpus with many segmented sentences for each sentence of the initial raw corpus. We expect the Language Model to be better at judging between all possibilities as it uses more contextual and global information.

4.3.3 Language Modeling

To estimate the probabilities of the Language Model, we rely on the Open Source tool KenLM by Heafield et al. [10]. It computes a LM with modified Kneser-Ney smoothing and interpolation and yield a standard ARPA file containing all the probabilities and backoff values. This file can be loaded by our software and is easy to use on the android device (ultimately stored as a SQLite database) to compute word sequence probabilities.

4.4 Candidates Selection

At this stage, we have all the data we need to perform the actual IME job. Further computation is done on the Android device. We first load all the required data into a SQLite database:

1. word list with IPA and Hàn-jī conversions
2. Autonomy scores
3. ngrams probabilities and backoff values

The process to build the candidates list is illustrated in Figure 2 and described below

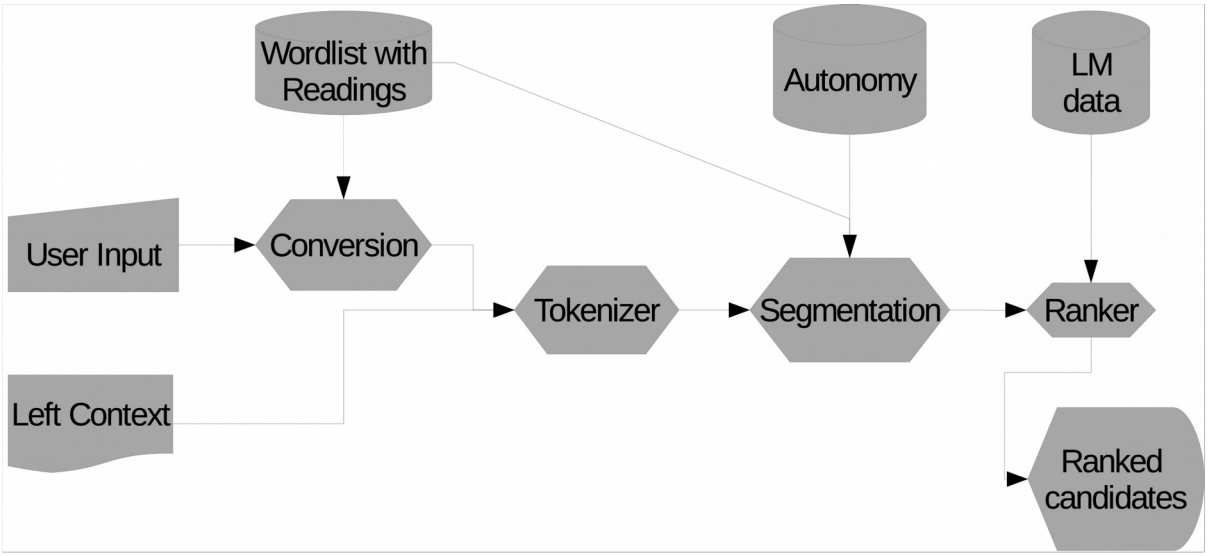

Figure 2: Building candidates list

Whenever text is input on the device, we apply the same rule-based algorithm as in 4.3 to attempt a conversion to IPA. On success we retrieve the possible correspondences in the word list.

We retrieve the left context in which the text is input from the OS API and perform the very same processing chain as in Section 4.3: Tokenization, IPA normalization, and segmentation.

Then we combine the candidates and the possible segmentations and rank the combinations according to the LM, we use the probability of the word sequences (left context + suggested conversion of user input) to define an order on the possible conversions to be provided to the user. For each possible conversion we obtain multiple segmentations. We consider the most probable segmentation for each candidate to build the ranking.

In the case where the input in empty, we rely on the context and the LM data to retrieve possible next words on which we apply the same ordering method.

5. Unsupervised Segmentation with Ambiguous Output

We use ELeVE as the basis of our unsupervised segmentation prior to model estimation. it is based on an "autonomy" measure computed from the normalized Variation of Branching Entropy (nVBE). This measure is an estimate of the extent to which a form (ngram of tokens) is syntactically autonomous. The details of the computation are given in [9]. The original segmentation algorithm we proposed is simply to select the segmentation which would maximize the average autonomy of the words in the segmented sentence. This is computed using dynamic programming. We let the EleVE software provide us the list of autonomy scores of the forms observed in the corpus and define a new segmentation algorithm to maintain some ambiguity in the output.

We observed that due to the occurrences of autonomous and frequent forms inside larger words, the maximization strategy tends to over segment the input. To compensate for this tendency, we keep not only the best solution of the maximization but also the n-bests which contain less boundaries than the first solution. To do this we first run the segmentation algorithm with a beam-search strategy to memorize n-best candidates and then filter out the solution that yield more cuts than the best one.

For a given input sequence of tokens, the n-best segmentations will share some common sub-sequences of words, as we train the Language Model on all the different segmentations, the common sub-sequences will be seen multiple times. We use this fact as a way to give more weight to less ambiguous parts of the segmentation and less weight to the ambiguous parts. This allows us to rely on the LM to disambiguate among different solutions.

6. Evaluation[8]

Due to the applied nature of this work, we set our priority to have a first version of a “smart” IME for Taiwanese and release it to provide it to users. For this reason we can only provide a very preliminary evaluation of our output. We consider this first implementation a baseline (which already useful to our users as is) on which improvement is to be made in the future.

To evaluate the relevance of our ranking and prediction, we use three texts that are aligned with the Romanization. The Romanization is considered as the input that could be made by a user and the Hàn characters sequence is our target. We compute the proportion of cases where the next target was in the n-bests candidates ranked by our IME for n=1,3,5,10. We compare the current state of our IME which uses both input and left context to our old system (which does not use the left context). We provide such figures for two versions of the same text (1227 words extracted from a 歌仔冊, an original and a “corrected” version) and 600 words from the Chapter 2 of a Taiwanese translation of «Le Petit Prince.»

歌仔冊 (original)	1-best	3-best	5-best	10-best
Old (no LM)	0.43	0.61	0.72	0.83
New (with LM)	**0.62**	**0.78**	**0.83**	**0.85**

歌 仔 冊 (corrected)	1-best	3-best	5-best	10-best
Old (no LM)	0.48	0.66	0.76	0.88
New (with LM)	**0.67**	**0.83**	**0.87**	**0.90**

Le Petit Prince	1-best	3-best	5-best	10-best
Old (no LM)	0.42	0.60	0.70	0.79
New (with LM)	**0.61**	**0.72**	**0.76**	**0.80**

8 This section was added to the camera-ready version of the paper following reviewers feedback and using newly obtained data. A more comprehensive evaluation remains to be done but we wanted to deliver the software to its users as early as possible.

We see a significant contribution of the language model. we succeed in giving a better ranking of the conversion candidates, this should greatly benefit to the user experience

7. Conclusion and future work

Much more should be said about the evaluation. Many parameters can now be tested and a qualitative error analysis is very likely to provide a good overview of the diversity and complexity of the actual usage of written Taiwanese. But due to time and space constraints, we leave this discussion for a future work.

Another important challenge is also to face the issue of illiteracy. We need to find convenient ways to help users write in Taiwanese, even if they never had the opportunity to learn it from school. A fully featured input system could provide feedback to beginners. We could turn our Input Method to some kind of writing/learning assistants and include features like spelling correction or post-editing suggestions (for example, by detecting the use of "false friends" from Mandarin in Taiwanese text). Convenient access to dictionary data and assessment alignment on CEFR may also enable us to help users in learning new vocabulary, for example if we can find collocations of higher levels of difficulty in CEFR in the corpus data.

Finally, we also need to find ways to involve users in the evolution of the database to include new vocabulary. As we choose not to connect the IME to the Internet, we will need to design a website or a separate application to let the user collaboratively enhance the database. Games With a Purpose may also be an interesting and efficient option to have users involved.

Such efforts have already started with the iTaigi[9] platform. In the future we hope it can provide us newly coined Taiwanese words to keep our lexicon up to date.

7. Acknowledgements

I am very grateful for all the openly available data mentioned in Section 3. Their authors made this project possible.
This work was sponsored by the MOFA's Taiwan Fellowship program which granted funding for conducting my research on Taiwanese language.

9 http://itaigi.tw/

參考文獻 [References]

[1] KLÖTER, Henning. *Written Taiwanese*. Otto Harrassowitz Verlag, 2005.

[2] 楊允言 「台語文語料處理 kah 線頂資源研究」 2014 亞細亞國際傳播社 ISBN:9868541891

[3] 中華民國 教育部 【臺灣閩南語常用詞辭典】 2011 http://twblg.dict.edu.tw/

[4] ALDERSON et al., *The development of specifications for item development and classification within The Common European Framework of Reference for Languages: Learning, Teaching, Assessment: Reading and Listening.* (2004) Final report of The Dutch CEF Construct Project.

[5] 國立成功大學台灣語文測驗中心 【全民台語認證語詞分級寶典】 亞細亞國際傳播社 2011. ISBN：9789868541832

[6] 吳守禮 【國臺對照活用辭典】ISBN：9573240882

[7] 陳柏中、林哲民 信望愛台語語料庫 https://bitbucket.org/pcchen/nan

[8] Iûn Ún-giân et al. *Tâi-gú-bûn Gú-liāu-kò͘ So͘-chi̍p kap Gú-liāu-khò͘ ûi Pún Tâi-gú Su-bīn-gú Im-chiat Sû-pîn Thóng-kè* (台語文語料庫蒐集及語料庫為本台語書面語音節詞頻統計). Hêng-chèng-īnn Kok-ka Kho-ha̍k Uí-oân-hōe Póu-chōu Choan-tôe Gián-kiù Kè-ōe Sêng-kó Pò-kò (行政院國家科學委員會補助專題研究計畫成果報告) 2005, NSC 93-2213-E-122-001-

[9] MAGISTRY, Pierre et SAGOT, Benoît. Unsupervized word segmentation: the case for mandarin Chinese. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012. p. 383-387.

[10] HEAFIELD, Kenneth, POUZYREVSKY, Ivan, CLARK, Jonathan H., *et al.* Scalable Modified Kneser-Ney Language Model Estimation. In: *ACL (2)*. 2013. p. 690-696.

[11] MAGISTRY, Pierre *Unsupervised word segmentation and wordhood assessment: the case for mandarin Chinese* (Doctoral dissertation, Paris 7 Diderot, Labex EFL).