A Multilingual Language Processing Tool for Uyghur, Kazak and Kirghiz

Mijit Ablimit*, Sardar Parhat*, Askar Hamdulla*, Thomas Fang Zheng†

*Xinjiang University, Urumqi, China
†Tsinghua University, Beijing, China

***Abstract*— Natural language processing for less popular languages is difficult, partly due to the high variations in the writing form. On the other hand, many minority languages in the same region share similar properties and can be processed in a similar way. This paper publishes an integrated multilingual language processing tool. Our aim is to provide an open, free and standard toolkit for minority language processing tasks, by a uniform user interface to support multiple languages. The present implementation supports Uyghur, Kazak, Kirghiz, three major minority languages in the Western China, and our focus was put on phonetic and morphological analysis. For the phonetic analysis, we build a multilingual parallel phoneme list, with similar phonemes grouped and character codes standardized. A multilingual syllable analyzer is also developed to detect spelling mistakes, and extract irregular spelling. For the morphological analysis, we build a multilingual morpheme segmentation tool that can extract morphemes by statistical analysis. This toolkit is extendable in terms of both functions and languages.**

I. INTRODUCTION

Natural language processing (NLP) for less popular languages is highly difficult. A particular problem is the high uncertainty of the writing forms of these languages. As an example, for the three major minority languages in Northwest China: Uyghur, Kazak, Kirghiz, there are nearly no widely recognized text corpora. Once we tried to construct, we found significant variations in words and expressions, which makes the regular NLP tasks very complex.

A main reason for this variance is the deep influence of the major language, Chinese and English, to these local and minority languages. This influence is greatly aggravated by the rapid development of information technology, which triggers a broad spectrum of cross-lingual and cross-cultural interaction, leading to unceasing coining of new words, new concepts and new expressions. Most of these new items are borrowed from or filtered in by Chinese, and the integration is forms that are full of noise, caused by the different spelling habits and different dialectal metamorphosis.

Another source of the uncertainty in the writing form is the historical changes of the writing system. For example, the Uyghur language uses Arabic characters at present, but 30 years ago, the roman characters were used. Even more various writing systems were used in more ancient times. These different written systems leave their heritage in the modern society, although less possible in the official medias, but everywhere in online forums and chatting tools. This problem was further accented by some software that use multiple or private coding schemes.

The third reason for the text form uncertainty is the mutual influence on pronunciations among people speaking different minority languages. For example, in the Xinjiang Uyghur Autonomous Region of China, people speaking Uyghur, Kazak, Kirghiz live together, and all these languages are written as pronounced. This means that when people of different nations talk with each other, their pronunciations will influence with each other, and this influence will quickly reflect in the writing system.

Due to the above three reasons, it can be seen that text resources collected from Internet for the three minority languages are very flexible, full of multiple codes , new words and phrase in various forms, expression clearly influenced by Chinese, English and other minority languages. For human beings, this flexibility is not a problem and we can quickly understand what the underlying `base-form' is, however for computers, it imposes a severe difficulty and has impeded the entire NLP research on the minority languages.

In this paper, we develop a compact and extendable framework to improve minority language NLP. Our goal is to provide a standard interface to perform various NLP tasks for multiple minority languages. With this framework, the basic functions will be published, and developers can contribute using the same API. The present implementation includes text normalization, stemming, and morphological analysis. Note that some researchers have developed some tools for Uyghur language [1], but these tools cannot address other minority languages.

Our work is part of the Multilingual Minorlingual Automatic Speech Recognition (M2ASR), which is supported by the National Fundamental Science of China (NFSC). The project is a three-party collaboration, including Tsinghua University, the Northwest National University, and Xinjiang University. The aim of this project is to construct speech recognition systems for five minor languages in China (Tibetan, Mongolia, Uyghur, Kazak and Kirgiz). However, our ambition is beyond that scope: we hope to construct a full set of linguistic and speech resources and tools for the 5 languages, and make them open and free for research purposes. We call this the M2ASR Free Data Program. All the data resources, including the tools published in this paper, are released on the website of the project http://m2asr.cslt.org.

II. SYSTEM ARCHITECTURE

The goal of our development is to provide a unified text processing interface for multiple minority languages, and make it convenient to extend and upgrade. Our framework roughly involves two layers, where the first layer focuses on Phonological processing and the second layer focuses on morphological processing. Some supporting tools are also developed. Note that this study focuses on three minority languages: Uyghur, Kazak, Kirghiz, and other languages will be added in the future work.

2.1 Phonological processing

The phonological tool involves code normalization and spell checking. The aim of this tool is to produce a normalized, cleaned text corpus for the subsequent morphological processing.

A. Code normalization

As mentioned, multiple spelling systems exist in the three languages, where Roman and Arabic characters are the mostly used, plus some less frequently used symbols, e.g., Cyrillic characters. Although Unicode is the default encoding scheme, various codes are still being used on different operating system and by different organizations. Table 1 shows some examples of the diverse coding schemes. Each Arabic character is expressed in a number of different codes. The first of our work was to build a code mapping table that normalizes all the diverse codes to a set of unified Roman codes. This code normalization is the first step of multilingual text normalization and phonetic processing.

TABLE I
SAMPLE OF PHONEME MAPPINGS

Unified Roman	IPA	Uyghur	Kazak	Kirghiz
A	ä	ه	ۇ	ه
		1749	1653	1749,1577,1607
e	ë	ې	ە	ئ
		1744	1609	1574,1569
y	j	ي	ي	ي
		1610	1610	1610
G	ɣ	غ	ع	ع
		1594	1593,65228, 65227	1593,1594

We designed a light-weight tool to perform the code normalization. It transforms various code schemes, like unicode, ASCII, and other codes into a set of unified normalized character codes. The default normalized codes are in the basic ASCII code zone, but can be also the standard unicode zone to avoid code ambiguity.

This open source tool is designed to be extendable. Users can freely change the code-map contents, and add their own personalized codes.

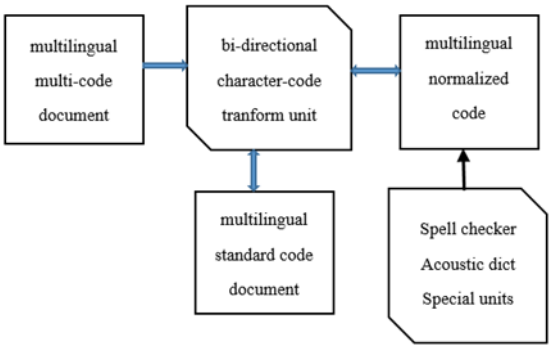

Fig. 1 The structure of multilingual phonetic processing tool.

B. Multilingual rule-based spelling checker

Due to the great flexibility (or, uncertainty) in the writing system, a spelling checker is predominantly important. We provided a rule-based spelling checker that can detect most of the misspelled words. As the syllable structure of these languages is stable, it is not hard to design a multi-lingual syllable segmenter, and based on the segmentation result, spelling mistakes could be detected. Some scholars have developed various syllable segmenter tools for individual languages, e,g., [2], but these tools are not flexible enough to be extended to involve additional functions and support other languages. In this work, we developed a multilingual syllable segmenter tool, which is based on a flexible architecture and can be extended to other languages that have clear syllable structures. Each language only needs to provide a syllable template. For example, the template for Uyghur language is {BA[B], BA[BB], BBA[BB], BAA[B]}. By this information, the program will iteratively segment a word into all possible syllable sequences.

Syllable analysis not only can detect most of the spelling mistakes, but also extract irregular foreign words which are imported from other languages and not transcribed correctly. Furthermore, syllable analysis can be used to detect the language of a sentence. A net crawler, for example, can use this tool as a filtering component to discover the target documents from the multilingual and multi-encoding environment of the internet.

C. Acoustic dictionary builder

Since the writing form reflects the pronunciation, acoustic dictionary can be automatically built from the spelling of the words. This convenience migrates to the multilingual scenario only if all the multilingual sentences are normalized to the unified code scheme. We designed a tool that provides a unified interface to make unified format of acoustic dictionaries of different languages.

Besides the regular words, there are some special words in each language, for example various forms of acronyms, numbers and time. Users can freely add the pronunciations of the special words to the dictionary automatically created.

We have not addressed additional ambiguity that are very special for certain languages. For example in Kazak, some phonemes like “y, w” are used as both vowels and consonants.

This ambiguity is mostly caused by spelling of foreign words. We will solve this problem in the next release.

2.2 Morphological processing

All the three languages are agglutinative languages, meaning that words are formed by a stem augmented by unlimited number of suffixes. The stem is an independent semantic unit while the suffixes are auxiliary functional units. Both stems and suffixes are called morphemes. Morphemes are the smallest functional units in agglutinative languages. Because of this agglutinative nature, the number of words of these languages can be almost infinite, and most of the words appear very rarely in the text corpus. Modeling based on a smaller unit like morpheme can provide stronger statistics hence robust models [8-11].

The total number of suffixes in each these 3 languages is around 120. New suffixes may be created, but this is the typical case. We developed a semi-supervised morpheme segmenter based on the suffix set. For a candidate word, an iterative searching algorithm is designed to produce all possible segmentation results by matching the stem set and the suffix set. Fig. 2 shows the flow chart.

When the morphemes are merged to a word, the phonemes on the boundaries change their surface forms according to the phonetic harmony rules. Morphemes will harmonize each other, and appeal to each other's pronunciation. When the pronunciation is precisely represented, the phonetic harmony can be clearly observed in the text. We train a statistical model using a word-morpheme parallel corpus, based on unit frequency and length parameter [6-7].

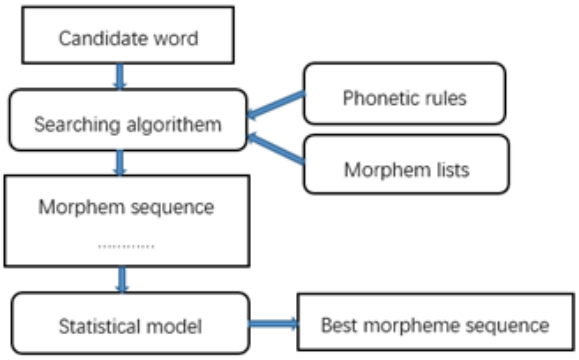

Fig. 2 Morpheme segmentation flow chart.

III. IMPLEMENTATIONS AND CORPUS CONSTRUCTION

A multilingual morphological processing tool is implemented for the three languages. The tool has been tested on Uyghur, Kazak, Kirghiz, and the results show that many spelling mistake and detected by this tool, and some detections are new morphemes.

3.1 Implementations

This tool is designed to reduce repeated programming work as much as possible. There are roughly three layers in this framework: (1) The parent layer is the multilingual phoneme-character processing unit, which conducts character normalization and acoustic rule analysis. (2) The middle layer that perform language-related processing such as definition of irregular phonetics, specific spelling rules, syllable templates etc. (3) The interface layer that provides users the file interface. Files and strings are processed at this layer, including code normalization, acoustic dictionary construction, pronunciation file preparation, as well as the definition of other particles including words, syllables, and characters.

We also implemented a multilingual morpheme segmentation tool that resides at the interface layer. This tool can learn morphological and phonetic rules from the word-morpheme parallel corpora. There are several constraints for the multilingual morpheme segmenter. First, the suffix set is a closed set, and all the surface forms are known. Second, the edit distance (Lowenstein) of the surface forms is no more than 1. Third, there is an optional stem list. These constraints ensure the quality of the learning and are mostly reasonable in practice.

Two segmentation tools were developed, one is trained in a supervised fashion and is based on the word-morpheme parallel training corpus; the second is trained in a semi-supervised fashion based on a suffix set and some phonetic rules. This tool can be extended to other similar language. Table 2 shows the training corpora for these languages

TABLE 2
MULTILINGUAL WORD-MORPHEME PARALLEL CORPORA

Language	Uyghur	Kazak	Kirghiz
word-morpheme parallel corpus (sentences)	10 000	5 000	3 000
suffix set (types)	124	124	124

For some applications, pseudo-morphemes are sufficient. For example, in speech recognition, pseudo morphemes can be used as the basic units for language modeling, without the necessity to obtain precise morpheme segmentations. In this case, the semi-supervised morpheme segmenter can provide a good trade-off between precision and efficiency.

3.2 Corpus construction

We have collected three text corpora, for Uyghur, Kazak, and Kirghiz respectively. The text was crawled down from publications and the internet. The phonetics processing tool was used to detect spelling mistakes and normalize the text. For example, Kazak language imports a lot of Cyrillic spellings which cause ambiguity. Removing this ambiguity reduced much human labor on acoustic dictionary construction. Table 3 shows the statistics of text corpora.

TABLE 3
MULTILIGNAL CORPORA FOR MORPHOLOGICAL PROCESSING

language	Uyghur	Kazak	Kirghiz
text copra (sentences)	500K	200K	40K

We have also implemented a general purpose multilingual morphological tool. It can segment a word into all possible morpheme sequences and a statistical algorithm is used to pick up the best candidate. Any statistical method can be easily incorporated into the architecture.

IV. CONCLUSION

We have discussed the system structure of our recent public tools used for multilingual phonetic and morphological processing. Originally, these tools were constructed to facilitate the corpus construction for less popular languages, but they can be used for minority NLP in general. This tool was designed by separating the program from the data. Users can easily use their own language-specific data. We hope these tool will provide a uniformed multilingual information processing platform and assist multiple speech processing tasks for minority languages, for example multilingual speech recognition.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (NSFC; grant 61462085, 61662078, and 61633013).

REFERENCES

[1] M. Ablimit, G. Neubig, M. Mimura, S. Mori, T. Kawahara, A. Hamdulla, "Uyghur *Morpheme-based* Language Models and *ASR,"* In Proc. ICSP, Beijing, 2010.

[2] M. Ablimit, A. Hamdulla, T. Kawahara, "Morpheme Concatenation Approach in Language Modeling for Large-Vocabulary Uyghur Speech Recognition," In Proc. Oriental-COCOSDA Workshop, 2011.

[3] M.Y. Tachbelie, S. T. Abeta, L. Besacier, "Using different acoustic, lexical, and language modeling units for ASR of an under-resourced language – Amharic," Speech Communication, 2013.1.

[4] Lee, T. Kawahara, and K. Shikano, "Julius -- an open source real-time large vocabulary recognition engine," In Proc. Eurospeech, pp. 1691--1694, 2001.

[5] Kwon and J. Park, "Korean large vocabulary continuous speech recognition with morpheme-based recognition units," Speech Communication, vol. 39, pp. 287–300, 2003.

[6] Graham Neubig, "Unsupervised Learning of Lexical Information for Language Processing Systems," PhD thesis, Kyoto University. 2012.

[7] M. Creutz, "Introduction of the morphology of natural language: Unsupervised morpheme segmentation with application to automatic speech recognition", PhD. Thesis, Helsinki University of Technology, Finland, 2006.

[8] M. Ablimit, T. Kawahara, A. Hamdulla, "Lexicon optimization based on discriminative learning for automatic speech recognition of agglutinative language," Speech Communication, 2014.5.

[9] M. Nuβbaum-Thom, A. El-Desoky Mousa, R. Schluter, Hermann Ney, "Compound Word Recombination for German LVCSR," In Proc. Interpeech, 2011.

[10] B. Xiang, K. Nguyen, L. Nguyen, R. Schwartz, J. Makhoul, "Morphological decomposition for Arabic broadcast news transcription", In IEEE-ICASSP 2006.

[11] A. El-Desoky, C. Gollan, D. Rybach, R. Schluter, H. Ney, "Investigating the use of morphological decomposition and diacrit", In Proc. Interspeech, 2009.

[12] M. Jongtaveesataporn, I. Thienlikit, C. Wutiwiwatchai, S. Furui, "Lexical units for Thai LVCSR," Speech Communication, pp.379–389, 2009.

[13] T. Pellegrini, L. Lamel, "Using phonetic features in unsupervised word decompounding for ASR with application to a less-represented language," In Proc. Interspeech, 2007.

[14] E. Arisoy, M. Saraclar, B. Roark, I. Shafran, "Discriminative language modeling with linguistic and statistically derived features," IEEE Trans. Audio, Speech & Language Processing, vol. 20, no. 2, pp. 540-550, 2012.