

A Manchu speller:

With a practical introduction to the natural language processing of minority languages*

You Hyun-Jo (Seoul National University)

You Hyun-Jo. 2014. A Manchu speller: With a practical introduction to the natural language processing of minority languages. *Altai Hakpo* 24. 39-67. The Altaic Society of Korea.

This article presents a Manchu speller developed using open-source proofing tools for use in a large-scale digitization of Manchu lexical and textual resources published during 17th and 18th centuries. A speller requires the list of words and affix rules. Morphological analysis is inevitable to develop a spell checker especially for morphologically rich languages. The dictionaries compiled for a speller would be a primary resource for the development of a morphological analyzer. The present Manchu speller, therefore, is introduced as a preparation for further development of morphological analyzer.

I intended to propose possibly best practices for non-computational linguists who want to take an initial step of natural language processing of minority languages without serious programming knowledge. Developing a spell checker and morphological analyzer is the initial and critical step for natural language processing. This article introduces the way in which researchers of minority languages only need to organize the list of lexical items and morphological rules of the target language into the format specified by the spell-checking engine. The main sections are dedicated to introducing popular spell-checking engines, briefly reviewing major issues and concepts of computational morphology with concrete examples, and finally demonstrating how to implement a working speller and morphological analyzer with the explanation of how to translating efficiently linguistic knowledge into the technical description.

* This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2012S1A5B4A01035397). I would like to thank prof. Kim Juwon, prof. Ko Dongho, and prof. Choi Woonho for careful comments and advice. This article would not be written without their guidance.

Keywords: the Manchu Language, Open-Source Spell Checker, Affix Rule, Computational Morphology, Morphological Analyzer

1. Background

This article presents a Manchu speller developed using open-source proofing tools for use in a large-scale digitization of Manchu lexical and textual resources published during 17th and 18th centuries under the project “Building of Manchu Dictionary and Literature DB for Integral Study of Manchu Language and Literatures.” The digital text collection¹⁾ is organized to contain most of major dictionaries and vocabulary books including *Daicing gurun i yooni bithe*²⁾ (大清全書, 1683), *Han i araha manju gisun i buleku bithe* (御製清文鑑, 1708序), *Han i araha nonggime toktobuha manju gisun i buleku bithe* (御製增訂清文鑑, 1771序); seminal Chinese literary texts translated into Manchu; and educational texts published in Korea.

A speller requires the list of lexemes and inflectional rules that describe prefixation, suffixation, infixation, stem alternation, and word composition. To make a complete list of all possible word forms is inefficient and technically impossible even in the newest personal computers, especially for morphologically rich languages. For example, a Hungarian single nominal base can yield up to a million word forms (Trón 2005). Morphological analysis is inevitable to develop a spell checker. The dictionaries compiled for a speller would be a primary resource for the development of a morphological analyzer.

I will discuss about spell checking and briefly introduce a prototypical morphological analyzer as the first step in natural language processing of Manchu language. “Creating an automatic morphological analyzer/generator is just one step in starting natural language processing; but especially for minority, emerging or generally lesser-studies languages, it is often a practical and extremely valuable first step, making use of corpora, lexicons, morphological grammars and phonological rules already produced by field linguists and descriptive linguists” (Beesley 2004). The spell

1) The list can be found at <<https://github.com/youhyunjo/manchu-spell>>.

2) There are several transliteration systems of the Manchu script. This article respects Möllendorff's with replacement of *ū* and *š* to *v* and *x* respectively.

checker for the Manchu language is hard to be implemented without the morphological analysis in that Manchu language is one of morphologically rich, agglutinative languages. Although a spell checker and a morphological analyzer share common lexical resources and algorithms, they are different in purpose and function. A morphological analyzer and part-of-speech tagger is more complex and has more functionalities for resolving the problem of homonym, syncretism, unknown word (Smrž and You 2010). A speller usually would work satisfactorily through a rather simple processing except that it should be equipped with an additional suggestion algorithm for misspelled words.

We must be clear about why we desire to develop a spell checker but not a morphological analyzer at this moment. We need a spell checker for improving the accuracy of the digitization of the Manchu texts. We did not intend to construct a pos-tagged or morphologically analyzed corpus. There is too little digital Manchu text to start meaningful natural language processing. We are currently digitizing a large collection of Manchu texts as a raw corpus, which will be served as language resources for developing natural language processing tools including the automatic morphological analysis and finally building Manchu corpora with morphological annotations.

2. Computational tools

2.1 Program based method: Its extensibility and limitation

I propose that a prototypical natural language processing tool for a minority language be developed using freely available software packages that “can take the static data and compile them into active computer programs that are interesting in themselves and which are necessary components in larger natural-language applications” (Beesley 2004). It is the easiest and fastest way to create a real-world linguistic tool even without programming skill if one has sufficient linguistic knowledge and can collect and organize linguistic data in the format required by the given software packages.

There are several free software open-source spell checking programs. The most actively used spell checkers are Ispell, Aspell, MySpell and Hunspell.

Ispell currently supports 51 languages³⁾ and Aspell 91 languages.⁴⁾ Hunspell is a spell checker and morphological analyzer. It reuses Aspell's lexical resources and provides dictionaries via OpenOffice.org.⁵⁾ The supported list already includes several modern Altaic languages, that is, Azerbaijani, Kirghiz, Mongolian, Turkish, Turkmen and Uzbek but it is hard to be expected to support minority languages like Manchu, Sibe, Ewenki, etc.

We do not need to rewrite the source codes of these spell-checking programs to support other languages. The data files for a new language can be separately created and added into their list of supported languages. The spell checkers require a dictionary file containing the list of words and affix file containing the list of morphological rules. For example, Mongolian dictionary for Aspell consists of 70 thousands of words and 2 thousands of affix rules. Finnish dictionary contains 88 thousands of words and 18 thousands of affix rules. Such a huge number of affix rules would not be written manually. They are mostly generated according to more sophisticated rules that cannot be directly exploited by the given spell checker.

Although their extensibility to the new language would be satisfactory, "the program-based spell-checking methods have their limitations because they are based on specific program code that is extensible only by coding new features into the system," and consequently new tools for several languages, for example, emberek (Turkish), hspell (Hebrew), uspell (Yiddish) and voikko (Finnish) are reimplemented to work around the limitations of hunspell (Pirinen and Lindén 2010).

Agglutinative languages with minor irregularities can be easily described by affix rules implemented in current spell checkers. It is challenging to create a spell checker for highly inflected languages, which however can be successfully described by two-level morphology. Two-level morphology is a computational model of word-form recognition and production using finite state transducers (Koskenniemi 1983). Two-level description can be adapted in a semi-automatic way to the present spell checkers that natively are not based on two-level morphology (Alegria *et al.* 2008). Two-level morphology is applicable for the analysis of morphologically complex languages (Koskenniemi 1983) but it is not the best model. It is not widely used in

3) <<http://fmg-www.cs.ucla.edu/geoff/ispell-dictionaries.html>>

4) <<ftp://ftp.gnu.org/gnu/aspell/dict/0index.html>>

5) <<http://www.openoffice.org/lingucomponent/dictionary.html>>

practice except for Finnish or other similar languages. It should be noticed that the use of finite-state transducers in two-level model does not guarantee efficient processing (Barton 1985).

2.2 Ispell and Aspell

Ispell is the oldest spell checker and has been most widely used in Unix-like operating systems. The international version of Ispell⁶⁾ supports not only English but also many European languages and several other families of languages including Hungarian, Hebrew, Vietnamese, Quechua, etc. It is possible but not ideal for morphologically rich languages since “only sixty four paradigms can be defined and it is not possible to link new morphemes after the suffixes” (Alegria *et al.* 2008). Such a limitation comes from the efficiency-oriented design on old computers, which can be easily improved for supporting much more paradigms on modern personal computers.

GNU Aspell⁷⁾ is a spell checker written and maintained by K. Atkinson. It did not improve the expression power of Ispell but is claimed to provide better replacements suggestion for misspelled word than Ispell and supports Unicode.

Aspell since version 0.60 began to support for affix compression, which is imported from MySpell. It is much more flexible than previous system that requires the list of all possible word forms. For example, French dictionary is made up with over 221 thousands entries. The list contains all inflected forms for each word. This method is impossible to apply to agglutinative languages where a word can take unbound number of affix combinations. Aspell now has a better description scheme for morphologically rich languages.

For a simple example to demonstrate the affix compression, we can consider the regular English past tense ending. There are four cases:

like → liked,	if a verb stem ends with /e/
stay → stayed,	if a verb stem ends with a vowel and /y/

6) “It is originally written in PDP-10 assembly in 1971 by R. E. Gorin. The C version was written by Pace Willisson of MIT. Walt Buehring of Texas Instruments added the emacs interface and posted to the net. Geoff Kuenning added the international support and created the current release.” <<http://www.lasr.cs.ucla.edu/geoff/ispell.html>>

7) <<http://aspell.net>>

fly → flied, if a verb stem ends with a consonant and /y/
 talk → talked, otherwise

Here is an example from the online user's manual⁸⁾ for Aspell that shows how to define the English past tense endings according to the Aspell's affix file format.

SFX D	0	d	e
SFX D	0	ed	[aeiou]y
SFX D	y	ied	[^aeiou]y
SFX D	0	ed	[^ey]

The first column indicates suffix or prefix. The second column is an arbitrary label for the suffix. The third column is the characters to be deleted. The fourth column is suffixes. The fifth column is the conditions. For example, The third rule indicates that *ied* will be added after removing final *y* if a verb stem ends with *y* following any letter except one of *a, e, i, o, u*. The square brackets mean any character from its items and the ones with caret means any character not from its items: *[aeiou]* means any character from *a, e, i, o, u* while *[^aeiou]* means any single character except *a, e, i, o, u*.

2.3 MySpell and Hunspell

MySpell is another spell checker that supports affix compression. It is written by Kevin Atkinson in C++ based on Ispell, which later imported affix compression from MySpell. It was introduced into OpenOffice.org as an integration of open source spell checkers but soon was replaced by Hunspell, which is an advanced spell checker, morphological analyzer, stemmer and generator extended for Unicode encoding, compounding and complex morphology. Hunspell is originally developed keeping in mind Hungarian natural language processing but the program itself is designed to be language independent (Halácsy *et al.* 2004; Németh *et al.* 2004). It is currently selected as the spell checker of several major softwares, for example, OpenOffice.org, Firefox, Google Chrome, Mac OS X, etc.

The two-fold suffix stripping is a noteworthy improvement of Hunspell. MySpell's affix compression algorithm strips only one suffix. Hunspell "allows

8) <<http://aspell.net/man-html/Affix-Compression.html>> for version 0.60.7-pre

a two-stage process of suffix stripping, whereby it can trade its efficiency to overcome memory limitations resulting from productive suffix-combinations” (Halácsy *et al.* 2004). It provides better facilities to describe agglutinative languages where multiple suffixes are concatenated at the end of a word.

Let us take an example from English to explain two-fold suffix stripping, nevertheless it is not a good practical example of the application of two-fold suffix rules. Below is an example from the Hunspell manual⁹⁾ that shows how to describe the double suffixing, for example, *drinkables*.

Dictionary:	Affix rules:	
drink/A	SFX S Y 1 SFX S 0 s .	SFX A Y 1 SFX A 0 able/S .

The entry *drink/A* in the dictionary file means that *drink* is a word of affix class A. In the affix file, the suffix A is defined as that it is rewritten to *able* and the suffixed word, for example *drinkable*, belongs to another affix class S. The suffix rule S can be subsequently applied to *drinkable*.

drink/A → *drinkable/S* ; by the rule *0/A* → *able/S*
drinkable/S → *drinkables* ; by the rule *0/S* → *s*

Hunspell is a morphological analyzer based on affix stripping, which is one of the two closely intertwined methods for word analysis — the other is the finite state transducer (Trón 2004). “The proof that phonological and morphological alternation rules, as used by linguists, were only finite-state in power, and could be implemented as finite-state transducers” (Beesley 2004). The Xerox Finite State Toolkit (XFST) is the most popular, and one of the most powerful tools for morphology and lexicon development but it is not free software (Trón 2004). Hunspell is “backward compatible with already existing and freely available spellchecking resources” (Trón 2004). Hunspell provides no less expressive power than FSTs as mentioned previously that it is possible to convert two-level description to Hunspell (Alegria *et al.* 2008).

9) <<http://sourceforge.net/projects/hunspell/files/Hunspell/Documentation/>>

3. Morphology

3.1 Computational morphology

Morphology deals with words, their internal structure, and how they are formed (Aronoff and Fudeman 2011: 1). Computational morphology deals with “how to identify words of distinct types in human languages, and how the internal structure of words can be modeled in connection with the grammatical properties and lexical concepts the words should represent” (Smrž and You 2010). Computational morphology provides a fundamental basis for many practical applications: morphological analysis, spell checking, stemming, word segmentation, input method, text-to-speech system, speech recognition, OCR, digital dictionary.

This section will discuss major issues of the computational morphology with examples of the Manchu language. However, The topics are general enough and important for describing the morphology of other languages. I will overview only the most significant and frequent morphological patterns of Manchu language, referring to Möllendorff (1892), Avrorin (2000), and Gorelova (2002). This article is not intended to discuss theoretical aspects of Manchu morphology, nor is it intended to exhaustively describe inflection, derivation, compounding and other possible morphological phenomena of the Manchu language.

3.2 Word units

The computational morphology is concerned with the structure of “words”. What is a word? It is generally defined as the smallest linguistic units that can be realized as a standalone utterance. Words, however, are intuitive units. It is not always easy to achieve a clear consensus in deciding word boundaries. We have to begin with defining word unit for the target language.

A word in a Manchu text can be easily distinguished because it is separated from other words by spaces or punctuation. We can provide a clear operational definition of word as a graphical unit delimited by spaces or punctuation. It is widely accepted practical definition in computational morphology but may not be perfectly satisfactory in linguistic perspective.

Mölldorff (1892: 5–6) presents the combinations of nouns with affixes: *i* and *ni* for the genitive case; *de* for the dative and the locative; *be* for the

Function	Stand-off	Suffix
Nominative	<i>boo</i>	
Genitive	<i>boo i</i>	<i>booi</i>
Accusative	<i>boo be</i>	<i>boobe</i>
Dative-Locative	<i>boo de</i>	<i>boode</i>
Ablative	<i>boo ci</i>	<i>booci</i>
Prosecutive	<i>boo deri</i>	
Predicative possessive		<i>boosingge</i>

Table 1. Nominal Suffixes

accusative, *ci* for the ablative. Gorelova (2002: 163) identifies the same five cases and notices that Manchu has only few cases in comparison with other languages of the Tungus-Manchu language family. Avrorin (2000: 75) suggests six cases with additional *deri* to the case system as the prosecutive¹⁰⁾ case.

A case marker may be written in one word with the preceding noun in such cases where it behaves like a nominal declension. It may be written separately from the noun as if an independent functional word, that would be call the postposition. The Table 1 summaries the combination of case markers with the noun *boo* ‘house’.

The word form *booderi* is not found in our text collection. The prosecutive *deri* is found to be written in one word only with a small set of lexical items referring to places, for example, *siden* ‘interval’, *jaka* ‘thing’, *dorgi* ‘inside’, *oilorgi* ‘outside’, *aibi* ‘where’. As *deri* is better described as a functional word than as a suffix, it is not an object of computational morphology.

Avrorin (2000: 87) introduces *-ingge* as a predicative possessive form.¹¹⁾ It is a suffix, i.e., always written in one word with the noun. There are found many words in the *-ingge* form.¹²⁾ As it is better to be described as a

10) It is translated from Russian *продольный*. Kawachi and Kiyose (2002) identify the suffix *deri* as prolative. The prosecutive case is a variant of the prolative.

11) It is translated from Russian *предикативно-притяжательная форма*. Kawachi and Kiyose (2002) identify *-ngge* as a suffixed bound noun. The English expression *John’s* is an example of predicative-possessive form.

12) For example, *adalingge*, *aibingge*, *aniyangge*, *biyangge*, *boosingge*, *deocingge*, *endurinngge*, *feingge*, *ferkingge*, *gingge*, *gubcingge*, *jeringge*, *kesingge*, *lingge*, *ninggucingge*, *niyalmaingge*, *sebsingge*, *sekjingge*, *sunjacingge*, *tubaingge*, *ubaingge*, *ulhingge*, *urkingge*; *dorgingge*, *urseingge*, and etc.

productive nominal suffix, it is an object of computational morphology.

3.3 Productivity

In morphology, we decompose a word into its component morphemes and analyze its internal structure. If the composition process is productive and there are many other words with the same pattern, it is practically useful to describe the pattern. If the pattern is specific only for small number of words, there is no advantage to decompose the words in the perspective of the computational morphology. It is better to be described as a whole, that is, as an indecomposable word.

Manchu language has plural suffixes *-sa|se|so*, *-ta|te*, *-si*, *-ri* used with human nouns denoting age, generation and relatives; or with nouns denoting peoples, nations, posts, ranks, titles and occupations¹³⁾ (Gorelova 2002: 134–136). Plural forms necessarily behave like a noun and can take a case marker, for example, *manju-sa-i* ‘of Manchu people’. Examples for each suffix are presented in Table 2.

It is not always easy to determine whether a morphological process is productive or not. The plural suffixes in the Manchu language may not be considered to be productive while the case markers are obviously productive. If we can provide a small fixed set of plural words, it is probably efficient to describe a plural word as an atomic word like English irregular plural word form *children*.¹⁴⁾ If Manchu plural suffixes can be applied to an unbound set of lexical items, it is better to describe plural forms through morphological rules.

Sg.	Pl.		Sg.	Pl.	
<i>sakda</i>	<i>sakdasa</i>	‘old man’	<i>amban</i>	<i>ambasa</i>	‘high official’
<i>age</i>	<i>agese</i>	‘prince’	<i>eshen</i>	<i>eshete</i>	‘father’s younger brother’
<i>gucu</i>	<i>gucuse</i>	‘friend’	<i>oke</i>	<i>okete</i>	‘wife of <i>eshen</i> ’
<i>solho</i>	<i>solhoso</i>	‘Korean’	<i>amji</i>	<i>amjita</i>	‘father’s elder brother’
<i>aha</i>	<i>ahasi</i>	‘slave’	<i>mafa</i>	<i>mafari</i>	‘grandfather’

Table 2. Examples of Plural Nouns

13) Avrorin (2000: 69–71) describes the combination of the suffix *-ri* with non human nouns, for example, *dobon* (sg.) ‘night’ and *dobori* (pl.) ‘nights’.

14) Creating a plural morpheme *ren* which attached only to the stem *child* is less elegant than simply entering *children* in the lexicon (Ritchie *et al.* 1992: 115).

	1st sg.	1st pl. exclusive	1st pl. inclusive	2nd sg.	2nd pl.	3rd sg.	3rd pl.
N.	<i>bi</i>	<i>be</i>	<i>muse</i>	<i>si</i>	<i>suwe</i>	<i>i</i>	<i>ce</i>
G.	<i>mini</i>	<i>meni</i>	<i>musei</i>	<i>sini</i>	<i>suweni</i>	<i>ini</i>	<i>ceni</i>
A.	<i>mimbe</i>	<i>membe</i>	<i>musebe</i>	<i>sinbe</i>	<i>suwembe</i>	<i>imbe</i>	<i>cembe</i>
D.	<i>minde</i>	<i>mende</i>	<i>musede</i>	<i>sinde</i>	<i>suwende</i>	<i>inde</i>	<i>cende</i>
Ab.	<i>minci</i>	<i>menci</i>	<i>museci</i>	<i>sinci</i>	<i>suwenci</i>	<i>inci</i>	<i>cenci</i>
PP.	<i>miningge</i>	<i>meningge</i>	<i>museingge</i>	<i>siningge</i>	<i>suweningge</i>	<i>iningge</i>	<i>ceningge</i>

Table 3. Personal Pronouns

3.4 Irregularity

Irregularity in word forms and structures refers to a morphological process that cannot be described by a prototypical rules. For example, the genitive form *mini* of 1st person singular pronoun *bi* cannot be described properly by the general affix rule: adding the suffix *-i* to the end of the stem. In contrast, the demonstrative pronouns *ere* ‘this’, *ese* ‘these’, *tere* ‘that’, *tese* ‘those’ get regular case suffixes. Table 3 shows the irregularity of personal pronouns.

Some irregularities can be understood by extended rules that are usually constructed on the basis of phonological information, but lexically conditioned irregularities mostly cannot be described by rules (Smrž and You 2010: 8–10).

3.5 Derivation versus inflection

Compounding, derivation and inflection are the main morphological processes. Compounding process consists of two or more lexical items while derivation process consists of a lexical item and an affix. Inflectional process is also characterized by affixation. It is important to distinguish between derivation and inflection for describing a new language but the boundary between two morphological processes is not strict.¹⁵⁾

Although the Manchu plural nouns are better to be described as inflection in consideration of that a plural suffix does not affect the lexical category of the word or semantics, they have a characteristic of derivation since the plural forms are restricted to only small group of nouns. If we are not biased

15) There are many possible criteria for the distinction of derivation and inflection. Payne (1997: 20–26) compared two processes and summarized widely known guidelines to distinguish them.

toward the traditional idea of European languages where case and number are undoubtedly well described in inflectional system, Manchu plural nouns would be described as derived ones.

Manchu passive-causative verbal form is another example. Considering the high productivity of the suffix *-bu*, it must be part of the inflectional system. Gorelova (2002: 244-252) describes the suffixes *-bu*, *-mbu*, *-nggi*, *-nu*, *-ca|ce|co* as the grammatical category of voice and notices that “in Tungus-Manchu studies voice remains one of the most problematic grammatical categories.” The polyfunctionality makes it difficult to develop a syntactic and semantic model. The suffix *-bu* produces passive and causative voices, for example:

gida-bu-mbi ‘to be pressed’ (passive), *gida-mbi* ‘to press’
gene-bu-mbi ‘to make somebody to go’ (causative), *gene-mbi* ‘to go’

The suffix *-mbu* has a different shade of meaning in contrast with the suffix *-bu* (Gorelova 2002: 246; Zakharov 1879: 160). It is used to causative for some verbs, for example, *dosimbu-* “to order smb. to enter” for *dosi-* “to enter” and some verbs may have two different forms meanings (Gorelova 2002: 249). For example, there exist two causative forms of the verb *wasi-*:

wasi- 1) to descend, 2) to fall, 3) to decline (of value)
wasibu- 1) causative of *wasi-*, 2) to demote,
wasimbu- 1) causative of *wasi-*, 2) to demote, 3) to issue (an order), to send down (an edict)

Möllendorff (1982: 8–10) categorized the suffix *-bu* and *-mbu* as a suffix added to the stem of verb to express mood and tense whereas described other productive suffixes *-na|ne|no*, *-nu|ndu*, *-ca|ce|co*, and etc. as derivational ones. Gorelova (2002: 246) described it in the frame of word-formation and stated that the suffix *-bu* “belongs to a verbal stem and correspondingly to all verbal forms, which are derived from this stem.”

3.6 Allomorphs

An allomorph is a variant form of a single morpheme, which is defined as a smallest linguistic unit delivering its own meaning or function. There are phonologically conditioned allomorphs and lexically conditioned ones. For example, the variants *-ha|he|ho* of Manchu preterite tense suffix depend on

No	Moods and Tenses	Example	Suffix
1	Imperative	<i>ara</i>	
2	Present Tense	<i>ara-mbi</i>	mbi
3	Infinitive	<i>ara-me</i>	me
4	Preterite	<i>ara-ha</i>	(h k ngk)[aeo]
5	Future	<i>ara-ra</i>	r[aeo] nd(ara ere oro)
6	Conditional	<i>ara-ci</i>	ci
7	Subjunctive Present	<i>ara-ki</i>	ki
8	Past Gerund	<i>ara-fi</i>	fi pi mpi
9	Imperfect	<i>ara-mbihe</i>	mbihe
10	Indefinite Past	<i>ara-habi</i>	(h k ngk)[aeo]bi
11	Pluperfect	<i>ara-habihe</i> ¹⁶⁾	(h k ngk)[aeo]bihe
12	Past Conditional	<i>ara-habici</i> ¹⁷⁾	(h k ngk)[aeo]bici
13	Adversative	<i>ara-cibe</i>	cibe
14	Concessive	<i>ara-cina</i>	cina
15	Optative	<i>ara-kini</i>	kini
16	Gerund I	<i>ara-mbime</i>	mbime
17	Gerund II	<i>ara-mbifi</i> ¹⁸⁾	mbifi
18	Gerund III	<i>ara-nggala</i>	ngg(ala ele olo)
19	Passive	<i>ara-bumbi</i> ¹⁹⁾	bu
20	Causative or Passive	<i>ara-bubumbi</i> ²⁰⁾	bubu
21	Verbal Noun	<i>ara-ha-ngge</i>	ngge
22	Indefinite	<i>ara-ha-le</i>	le, lengge
23	Adverbial	<i>ara-ra-lame</i>	l[aeo]me
24	Durative	<i>ara-hai</i>	(h k ngk)[aeo]i
25	Terminal	<i>ara-tala</i>	t(ala ele olo)
26	Extreme degree of action	<i>ara-tai</i>	t[aeo]i

Table 4. Manchu Verbal Suffixes

- 16) Kawachi and Kiyose (2002) describe *-ha bihe* as pluperfective verb phrase but do not identify *-habihe* as a suffix. There are very rare cases in our text collection, for example, *jihebihe*, *sehebihe*, *henduhebihe* and few more, while *-ha bihe* phrases are quite frequent.
- 17) There is only one case found in our text collection: *bihebici* (自已有之有。若有時。) in *Dacing gurun i yooni bithe*, while *-ha bici* phrases are frequent.
- 18) The suffix *-mbifi* is not found in our text collection while *-me bifi* occurs several times. Kawachi and Kiyose (2002) do not list both *-mbifi* and *-me bifi* in their index of particles and endings.
- 19) It is presented as *arambumbi* in Möllendorff (1892).
- 20) It is presented as *arambubumbi* in Möllendorff (1892). The passive-causative forms *arambumbi* and *arambubumi* are not found in our text collection. They are probably misspelled.

the phonological condition of the stem: *ara-ha* ‘made’, *gene-he* ‘went’, *oyo-ho* ‘bent’. And a minor group of verbs get the variants *-ka|ke|ko*, which cannot be expected from their phonological condition because it is the lexical property of those verbs.

Möllendorff (1892: 8–10) describes 23 verbal forms (No. 1–23 in Table 4) that express the moods and tenses.²¹⁾ Gorelova (2002: 267–321) describes a few more verbal forms: *-hai*, *-tala*, *-tai*.

The three suffixes *-ngge*, *-le*, and *-lame* (No. 21–23) do not attach to a verb stem but after other suffixes *-ha* or *-ra*. These complex forms will be discussed in the subsequent section. Möllendorff (1892: 9) presented several examples: *ara-ha-ngge*, *ara-ra-ngge*; *ara-ha-le*, *ara-ra-le*²²⁾; *ara-ha-le-ngge*, *ara-ra-le-ngge*²³⁾; *ara-ra-lame*.

There are nine allomorphs for the preterite tense form and other related forms: *-habi*, *-habihe*, *-habici*, *-hai*, *-hangge*, *-hale*, and *-halengge*.²⁴⁾ Most of verbs conjugate in *-ha|he|ho* or *-ka|ke|ko* forms and only small number of verbs have *-ngka|ngke|ngko* forms. The phonologically conditioned variations *a|e|o* and the lexically conditioned variations *h|k|ngk* are summarized below with verb examples:

<i>ara</i> - ‘to write’	<i>-ha</i>	<i>gene</i> ‘to go’	<i>-he</i>	<i>oyo</i> - ‘to bend’	<i>-ho</i>
<i>jala</i> - ‘to pause’	<i>-ka</i>	<i>tuhe</i> ‘to fall’	<i>-ke</i>	<i>foso</i> - ‘to shine’	<i>-ko</i>
<i>sa</i> - ‘to stretch’	<i>-ngka</i>	<i>guwe</i> - ‘to tweet’	<i>-ngke</i>	<i>bo</i> - ‘to pierce’	<i>-ngko</i>

- 21) One of the reviewers informed that Möllendorff’s terminology is not preferred in contemporary Altaic studies and suggested Kawachi and Kiyose (2002)’s terminology: 2) non-perfective finite, 3) non-perfective converb, 4) perfective participle, 5) prospective finite, 6) conditional converb, 7) desiderative finite, 8) perfective converb, 9) perfective processive, 10) perfective finite, 13) concessive converb, 14) optative finite, 15) optative finite, 16) simultaneous converb, 18) prefatory converb, 19) passive-causative, 20) denominal adjective, 22) deverbal nominal, 24) durative converb, 25) terminative converb, 26) intensive verb suffix.
- 22) The two complex forms *-hale* and *-rale* are rarely used in text. Kawachi and Kiyose (2002) present *-hale* as a deverbal nominal suffix but do not list *-rale* in their index. The suffix *-rale* is used in only few cases: *generale urse* (所去の人) and *jiderele urse* (凡来人凡来者) in *Manju nikan xu adali yooni bithe*; *jeterale* (凡吃的) and *isinarale* (凡所到) in *Dacing gurun i yooni bithe*.
- 23) The two complex forms *-halenngge* and *-ralengge* are not found in text except *bisirelengge*.
- 24) For example, *wasi-ka*, *wasi-kabi*, *wasi-kabihe*, *wasi-kabici*, *wasi-kai*, *wasi-kangge*, *wasi-kale*, and *wasi-kalengge* for the verb *wasimbi*.

There are six allomorphs for the future tense form and other related forms: *-rangge*, *-rale*, *-ralengge*, and *-ralame*.²⁵⁾ Most of verbs take *-ra|re|ro* form except small number of verbs. The phonologically and lexically conditioned variations are summarized below:

<i>ara-</i> ‘to write’	<i>-ra</i>	<i>gene-</i> ‘to give’	<i>-re</i>	<i>oyo-</i> ‘to bend’	<i>-ro</i>
<i>haira-</i> ‘to regret’	<i>-ndara</i>	<i>guwe-</i> ‘to tweet’	<i>-ndere</i>	<i>jo-</i> ‘to recall’	<i>-ndoro</i>

There are three allomorphs for the past gerund form. Most of verbs conjugate in *-fi* form. Only small number of verbs conjugate in *-pi* or *-mpi*. There is no phonological variation. Examples are presented below:

<i>ara-</i> ‘to write’	<i>-fi</i>	<i>gene-</i> ‘to go’	<i>-fi</i>	<i>oyo-</i> ‘to bend’	<i>-fi</i>
<i>jala-</i> ‘to pause’	<i>-pi</i>	<i>elde-</i> ‘to shine’	<i>-pi</i>	<i>nioro-</i> ‘to become green’	<i>-pi</i>
<i>sa-</i> ‘to stretch’	<i>-mpi</i>	<i>we-</i> ‘to melt’	<i>-mpi</i>	<i>jo-</i> ‘to recall’	<i>-mpi</i>

3.7 Inflectional classes

Aronoff (1994: 64) defines an inflectional class as a set of lexemes whose

Present	Preterite	Past Gerund	Future	
<i>ara-mbi</i>	<i>ara-ha</i>	<i>ara-fi</i>	<i>ara-ra</i>	‘to write’
<i>ubaliya-mbi</i>	<i>ubaliya-ka</i>	<i>ubaliya-fi</i>	<i>ubaliya-ra</i>	‘to change’
<i>fara-mbi</i>	<i>fara-ka</i>	<i>fara-pi</i>	<i>fara-ra</i>	‘to faint’
<i>jura-mbi</i>	<i>jura-ka</i>	<i>jura-fi</i>	<i>jura-ndara</i>	‘to set out’
<i>jala-mbi</i>	<i>jala-ka</i>	<i>jala-pi</i>	<i>jala-ndara</i>	‘to desist’
<i>ca-mbi</i>	<i>ca-ngka</i>	<i>ca-fi</i>	<i>ca-ra</i>	‘to stretch’
<i>sa-mbi</i>	<i>sa-ngka</i>	<i>sa-mpi</i>	<i>sa-ra</i>	‘to stretch’ ²⁶⁾
<i>je-mbi</i>	<i>je-ngke</i>	<i>je-mpi</i>	<i>je-ndere</i>	‘to bear’ ²⁷⁾

Table 5. Inflectional Classes

25) For example, *gene-rengge*, *gene-rele*, *gene-relengge*, and *gene-releme*.

26) It should be distinguished from the frequent verb *sa-* ‘to know’. These two verbs conjugate in different way. The verb *sa-* ‘to know’ conjugates in the ordinary way: *saha*, *safi*, *sara*, and etc. There are entries *sambi* (伸開), *sangka* (疎遠), and *sangkabi* (已伸) in the dictionary NB; and *sampi* (引領之引) in *Daicing gurun i yooni bithe*.

27) It should be distinguished from the frequent verb *je-* ‘to eat’. These two verbs conjugate in different way. The verb *je-* ‘to eat’ conjugates in unique way: *jefu* (imperative), *jekje*, *jetere*, *jefi* and etc. There is no entry like *jembi* ‘to bear’ in the dictionary NB but *jempi* (忍心), *jengke* (忍了), *jengkekv* (心裏不忍); and *jendere* (忍) in *Daicing gurun i yooni bithe*.

members each select the same set of inflectional realizations. For example, Avrorin (2000: 190) proposed four classes of verbal suffixation. It is not useful for computational description because it is not organized strictly by the surface forms and, moreover, it does not consider another inflectional variants *-fi|pi|mpi*. I propose more inflectional classes in Table 5.

The Table 5 reflects the fact that Manchu verbal suffixes are categorized into five groups. The suffixes of the group (1) are constant and never change irrespective of verb stems. In the group (2), the allomorphs are phonologically conditioned. The variation of the suffixes in the groups (3-4) is phonologically and lexically conditioned. In the group (5), the allomorphs are lexically conditioned.

- (1) *-mbi, -me, -ci, -cibe, -cina, -ki, -kini*
- (2) *-nggala|nggele|nggolo*
- (3) *-ha|he|ho, -ka|ke|ko, -ngka|ngke|ngko* and related series of complex suffixes (for example, *-habi, -hangge, -hale, -halengge, -hade, -hai, -hakv, -hakvbi*);
- (4) *-ra|re|ro, -ndara|ndere|ndoro* and related series of complex suffixes (for example, *-rade, -rangge, -rale, -ralame, -rakv, -rakvci, -rakvn, -rakvngge*)
- (5) *-fi, -pi, -mpi*

3.8 Complex affixes

A word can be formed by combining stem with two or more affixes. The German circumfix *ge—t* is a typical example of prefix-suffix combination. The concatenation of inflectional suffixes is frequently observed in agglutinative languages. Affixation has been one of primary concerns in the early natural language processing. There have been proposed several “stemming” algorithms.²⁸⁾ Hunspell’s twofold suffix stripping makes it possible to describe complex suffixes as concatenations of simple suffixes.

The negative *akv* and interrogative *o* in Manchu language are attached after other verbal suffixes and form complex suffixes. The negative particle *akv* is used as a separate functional word for the negation of existence or attribute (Gorelova 2002: 260), for example: *baita akv* ‘there is no business’, *mejige akv* ‘there is no news’, *yargiyan akv* ‘is not true’, *mangga akv* ‘is not difficult’.

28) “It effectively works by treating complex suffixes as compounds made up of simple suffixes, and removing the simple suffixes in a number of steps (Porter 1980: 130).”

For the negation of a verbal form, *akv* is used as a suffix except for the present tense. The tense suffixes *-ha* and *-ra* precede the negative *akv* and are contracted into *-hakv* and *-rakv*. Some contractions are phonologically unexpected: *-he* and *-akv* combine into *-hekv* but *-re* and *-akv* into *-rakv*. Another suffix can be followed after *akv* to form complex suffixes, for example, *-hakvbi*, *-hakvci*, *-hakvn*, *-hakvni*, *-hakvngge*, *-hakvnggeo*, etc. Examples from Möllendorff (1982: 12): *bi gisurembi akv* ‘I do not speak’, *ara-hakv* ‘he has not written’, *gene-hekv* ‘he did not go’, *ara-rakv* ‘he will not write’, *gene-rakv* ‘he will not go’, *gene-rakvci* ‘if he does not go’, *bisi-rakvngge* ‘those who are not present’

In Manchu language, interrogative particles *-o* and *-ni|n* combine with verbal forms to express a question and may be added after nominal forms (Gorelova 2002: 322-324). The suffix *-ni* after the negative particle *akv* frequently but not always loses its final vowel and changes into *-akvn*. Examples of negative suffixes attached to verbal forms, attested in our text collection: *gene-mbi-ni*; *gene-he-ni*; *gene-rakv-ni*, *gene-rakv-n*; *gene-hekv-ni*, *gene-hekv-n*; *gene-mbi-o*; *gene-he-o*; *gene-re-o*; *gene-rakvngge-o*; *gene-hekvngge-o*.

Several other suffixes can be analyzed into the particle particles *-ha* and *-ra* combined with nominal suffixes *-ngge*, *-de*, *-i* or other particles (Möllendorff 1982: 8–10; Gorelova 2002: 225, 281, 319–320). For example: *tuwa-hale*, *tuwa-rale*; *tuwa-halengge*, *tuwa-ralengge*; *tuwa-ralame*; *tuwa-hangge*, *tuwa-rangge*; *tuwa-hade*, *tuwa-rade*; *tuwa-hai*.

3.9 Homonymy and syncretism

Ambiguity is one of the most challenging problems in all aspects of linguistic study. Computational morphology is not an exception. Ambiguous word forms can be analyzed in multiple ways. A homonym is a group of words that are identical in orthography or pronunciation but have different unrelated meanings, for example, English *bank* means ‘a financial institution’ or ‘sloping land’. Syncretism or systematic ambiguity occurs due to inflectional morphology, for example, English *lives* can be the third person singular conjugation of the verb *live* or the plural of the noun *life*. Polysemy is the ambiguity of a single lexical item that has two or more different but related senses, for example, *notebook* means a book with blank pages or a portable personal computer. In polysemy, part-of-speech can be different, for example, *book* as noun means a written work and as verb means reservation.

The Manchu dictionaries *Han i araha manju gisun i bithe* (HB) and *Han*

i araha nonggime toktohuha manju gisun i buleku bithe (NB) are thematic dictionaries, where a homonymous or polysemous word occurs multiple times in different thematic sections. The editors of two dictionaries did not discriminate homonymy and polysemy. Below are examples of homonymy and polysemy in Manchu, taken from the dictionary NB. English translations are based on Concise Manchu-English Dictionary edited by Jerry Norman.

- sa* 1. imperative of *sambi* ‘to know’ (使知道), 2. hemp grass (麻草), 3. thill of oxcart (牛車轅)
niyalma 1. man (人), 2. philtrum (人中)
feye 1. wound (傷) 2. lair (窩)
jalan 1. rank (隊), 2. generation (世), 3. an administrative unit (甲喇), 4. degree of kinship (輩數)
galju 1. a quick and accurate archer (手快背中) 2. slippery ice (冰滑處)
leli 1. armor (護甲) 2. wide (寬廣)
holo 1. ditch (溝) 2. valley (山谷) 3. false (虛假) 4. aurochs (瓦隴溝) 5. Lithuanian big black wolf-size beast (獲落)
namu 1. lettuce (生菜) 2. ocean (洋)

Some morphemes having related senses differ in part-of-speech, for example, *aga* as noun means ‘rain’ (雨) and as verb stem means ‘to rain’ (下雨); *sakda* as nominal means ‘old’ (老) or ‘old man’ and as verb stem means ‘to get old’.²⁹⁾ Some morphemes are completely different grammatically and semantically, for example, the adjective *kara* means ‘black’³⁰⁾ but as the verb stem *kara* means ‘to lookout’ (瞭望). These words produce ambiguity when they are inflected or derived.

- | | |
|----------------|---|
| <i>sakdaki</i> | (1) <i>sakda</i> ‘old’ (老) + <i>ki</i> ‘breath’ (氣) → ‘oldish’ (老氣)
(2) <i>sakda</i> - ‘to get old’ + <i>-ki</i> modal particle → ‘may (he) get old’ |
| <i>karaki</i> | (1) <i>karaki</i> ‘crow’ (青鴉).
(2) <i>kara</i> - ‘to lookout’ + <i>-ki</i> modal particle → ‘may (he) lookout’ |

29) There must exist a verb stem *sakda*- even though there is no entry *sakdambi* but *sakdaka* (老了) and *gihv sakdambi* (放陳了) in the dictionary NB. Several conjugated forms are found in our text collection: *sakdabumbi*, *sakdabure*, *sakdaci*, *sakdacibe*, *sakdafi*, etc.

30) In the dictionary NB, *kara* is translated into 黑馬 ‘black horse’, which corresponds to *kara morin* in *Daicing gurun i yooni bithe*.

The present verb form *jembi* is another example on syncretism. Two verbs *je-* ‘to eat’ (喫) and *je-* ‘to bear’ (忍) coincide in present tense but differ in other conjugations: *jekē, jeter, jefi* for eating; *jengke, jendere, jempi* for bearing.

The words *sa-* ‘to know’, *sa-* ‘to stretch’, *saha-* ‘to stack’ and *saha* ‘hunting’ cause more complicate systematic homonymy. The verb *sa-* ‘to know’ (知道) and *sa-* ‘to stretch’ (伸開) coincide in *sa-mpi* form but differ in other conjugation: *sa-ngka, sa-mpi* for stretching; *sa-ha, sa-fi* for knowing. The verbal form *saha* ‘knew’ (知道了) is identical to the noun *saha* ‘hunting’ (畋獵) and the imperative of the verb *sahambi* ‘to stack’ (砌): *sa-ha* (*sa-* ‘to know’), *saha* (*saha-* ‘to stack’), *saha* ‘hunting’; *sa-hade* (*sa-* ‘to know’), *saha-de* (*saha* ‘hunting’).

4. Implementation

4.1 Computational lexicon

Lexical resources are essential for developing natural language processing components. A morphological analyzer cannot be built only with morphological knowledge. It requires the ‘computational’ lexicon, that is, the repository of lexical information for specific NLP tasks that usually require syntactic or semantic information (Pustejovsky 1999). The simplest form of a lexicon is the list of words. For morphological analysis, the words should be categorized in terms of their parts of speech and inflectional patterns.

A first computational lexicon for a minor language can be created by digitizing already existing paper dictionaries and vocabularies. For the Manchu language, I used the entries of two officially authorized dictionaries³¹⁾ *HB* and *NB*. It is enough to digitize entries only or extracting their entries. In this article, we are considering such languages that already have been well described linguistically but there has not been developed a natural language processing tool for them. If there does not exist a dictionary for the target language, we might have to go fieldwork to collect lexical information for compiling a dictionary.

4.2 List of all possible word forms

Creating the list of all existing word forms is the straightforward way

31) I appreciate Prof. Chong Chemun’s generous permission to use the list of entries of two dictionaries digitized through lengthy efforts.

to provide the dictionary for spell checkers. For example, French Aspell dictionary contains totally 641 thousands word forms. There are listed all declensions and conjugations for each word, for example, *aima*, *aimai*, *aimaient*, *aimais*, *aimait*, *aimant*, *aimas*, *aimasse* and another two dozen conjugation forms for a verb *aimer* ‘to love’. Dictionaries for Ispell and Aspell previous version that did not support affix compression are created in this way. Those word forms can be generated by using other programming tools.

It is a practically reasonable approach to precompose all possible word forms for Manchu language because there is not a great number of suffixes in Manchu as in Hungarian or Korean, for which it is irrational to write down all word forms. Manchu language has relatively smaller number of suffixes as an agglutinative language. Aspell dictionaries for Azerbaijani, Turkish, Turkmen and Uzbek languages are examples of the exhaustive list approach. For example, Uzbek Aspell dictionary consists of 97,000 entries, among which there are 86 derivational and inflectional forms for *gapir*- ‘to speak’. There are found about 60 inflected forms of a verb in our Manchu text collection. This list can be used as a spellchecking dictionary without additional processing.

If there is no available digital text collection for extracting a list of word forms, we have to generate all possible word forms through morphological rules. And it is generally true that there are few digital resources for minority languages.

I did not consider this approach even at the initial stage of the development of the Manchu speller. It does not need to generate all possible Manchu word forms in advance using other programming language since newest version of Aspell inherently supports affix rules. Manchu inflectional system is quite simple and therefore can be described easily by manually written affix rules in Aspell. Some languages have more complex morphology and require a huge number of affix rules, which could not be written directly. For example, Korean spellchecking dictionary for Hunspell includes more than 55,000 rules generated using Python script.³²⁾

Morphological rules are reusable for developing a morphological analyzer but the list of all possible word forms are not. There is no practical difference between the list of all possible word forms and the well-described dictionary with affix rules if we are considering only the spellchecking task. If it is

32) <<https://github.com/changwoo/hunspell-dict-ko>>

intended to provide initial resources for boosting further natural language processing for a new language, it is better to elaborate a lexicon with systematic morphological rules.

4.3 Two strategies: Entry in dictionary or affix rule

There is one obvious principle: derivational forms to be listed as entries in the dictionary, inflectional forms to be described by affix rules. However, the distinction between inflection and derivation is not always clear theoretically and practically. A productive “derivation” may be more efficiently described by rules. For example, English *-tion* to *-tional*, *-ize* to *-ization*, *-ate* to *ator-*, *-ous* to *-ousness*, etc. An idiosyncratic or irregular inflection may be easily processed by using the complete list of forms, otherwise we need to invent many specific rules which are applicable to only a few words. For example, it is an ineffectual attempt to create affix rules for English irregular plural form *teeth* from *tooth*; or irregular verb forms *took* and *taken* from *take*.

In many cases we have to choose an approach rather than the other even when there is no absolute correct answer. In this section I will present as such examples Manchu plural noun forms and passive-causative verb forms. In both cases, the word forms can be described by affix rules or listed as entries in the dictionary. The affix rule approach would be preferred for describing passive-causative verb forms because the suffix *-bu-* is so highly productive that almost all verbs can take the suffix *-bu-*. On the other hand, the entry-in-dictionary approach would be preferred for plural forms because several different plural suffixes are applied to only small number of lexical items.

To begin with, let me briefly sketch how the two approaches are implemented in Hunspell system. The following example codes work also in Aspell except two-fold suffix rules. For simplicity let us assume that there are four inflectional forms *gene-mbi*, *gene-fi*, *gene-bu-mbi*, *gene-bu-fi* of the verb *gene-* ‘to go’.

The causative form *genebu-* can be listed in dictionary file as an independent entry. There is no information about the relationship between *gene-* and *genebu-*. However, as it will be discussed later, Hunspell provides a mechanism to describe the derivational and inflectional relationship between entries. The tags at the end of the words */V* indicates inflectional category for which affix rules are described in the affix file.

Dictionary:	Affix rules:
gene/V	SFX V Y 2
genebu/V	SFX V 0 mbi .
	SFX V 0 fi .

The affix rule can generate the causative form *genebu-* from the verb stem *gene-*. The tag /BV at the end of *gene* indicates that the verb takes two affix categories B and V: *gene/B* expands to *genebu/V* by the B rule and then to *genebumbi* and *genebufi* by the V rules; *gene/V* expands to *genembi* and *genefi*. Consequently, the four forms can be handled successfully with a single lexical entry and two suffix groups. These two-fold affix rules are supported only by Hunspell and can be rewritten into simple affix rule for Aspell.

Dictionary:	Affix rules:	
gene/BV	SFX B Y 1	SFX V Y 2
	SFX B 0 bu/V .	SFX V 0 mbi .
		SFX V 0 fi .

The entry-in-dictionary approach results in a large dictionary and simple affix rules. Below is an example of the entry-in-dictionary approach for Manchu plural noun forms. The tag /N at the end of an entry in the dictionary corresponds to the flag of the affix rule. The example shows a dictionary for Manchu plural nouns: *sakda* (sg.) and *sakdasa* (pl.) are equally registered as entries in the dictionary. There is no information about the relationship between *sakda* and *sakdasa*. All nouns belong to one category because singular and plural nouns take the same case marker. This method can be implemented without additional effort because plural forms are registered as entries in the dictionaries *HB* and *NB*.

Dictionary:				Affix rules:
sakda/N	agese/N	aha/N	okete/N	SFX N Y 4
sakdasa/N	iregen/N	ahasi/N	amji/N	SFX N 0 i .
amban/N	iregese/N	eshen/N	amjita/N	SFX N 0 be .
ambasa/N	gucu/N	eshete/N	mafa/N	SFX N 0 de .
age/N	gucuse/N	oke/N	mafari/N	SFX N 0 ci .

Affix rule approach results in a smaller dictionary but a larger number of complex affix rules. Below is an example of the affix rule approach for Manchu plural noun forms. Two-fold affix rules are supported only by Hunspell. The example shows a dictionary and suffix rules: *sakdasa* (pl.) is not registered as an entry but can be generated by applying the suffix rule to *sakda* (sg.).

Dictionary:	Affix rules:	
sakda/Ns	SFX s Y 4	SFX N Y 4
amban/Ns	SFX s 0 sa/N a	SFX N 0 i .
age/Ns	SFX s n sa/N an	SFX N 0 be .
iregen/Ns	SFX s 0 se/N [eu]	SFX N 0 de .
gucu/Ns	SFX s n se/N en	SFX N 0 ci .

Two strategies can be mixed in order to describe a single category. Let us consider how to describe Manchu passive-causative suffix *-bu-* and infrequent allomorph *-mbu-* and the duplication *-bubu-*: *gene-bu-* causative of *gene-* ‘to go’; *dosi-mbu-* causative of *dosi-* ‘to enter’; *wasi-* ‘to descend’, *wasi-bu-* ‘to demote’, *wasi-mbu-* ‘to issue (an order)’; *tuci-* ‘to come out’, *tuci-bu-* ‘to take out’, *tuci-bubu-* causative of *tucibu-*.

The easiest solution might be to list each passive-causative form as an independent dictionary entry. Below is an example of dictionary for the verbs *gene-*, *dosi-*, *wasi-* and *tuci-*. The flag /V indicates that all these verbs take same suffixes.

Dictionary:				
gene/V	dosi/V	wasi/V	wasimbu/	tucibu/V
genebu/V	dosimbu/V	wasibu/	tuci/V	tucibubu/V

It is, however, not an efficient method. Below is an example of dictionary designed in the mixed approach: highly productive regular suffix *-bu-* is described by affix rule but other infrequent patterns are listed as entries in the dictionary.

Dictionary:				
gene/BV	dosimbu/V	wasimbu/V	tuci/BV	tucibubu/V
dosi/V	wasi/BV			

4.4 Affix Groups

Affix rules are efficiently organized when affixes are grouped together by the morphological behavior. Allomorphs would be undoubtedly described within a affix rule group. It is not explicitly noticed in the previous sections as it is very natural way to describe allomorphs as the variations of a single grammatical entity in linguistics. For example, allomorphs of Manchu preterite tense suffix *-ha*, *-he*, *-ho* are better to be described within a suffix rule group as follows:

Dictionary:	Affix rules:
ara/h	SFX h Y 3
gene/h	SFX h 0 ha a
bolgo/h	SFX h 0 he e
	SFX h 0 ho o

Compare the example presented below, where allomorphs are independently described. It is a possible solution. However, it is not efficient one if the alternation is completely predicted from the phonological condition.

Dictionary:	Affix rules:		
ara/a	SFX a Y 1	SFX e Y 1	SFX o Y 1
gene/e	SFX a 0 ha .	SFX e 0 he .	SFX o 0 ho .
bolgo/o			

We do not have to be restricted to the allomorphs. Any set of affixes that have same morphological behavior can be described within one affix rule group. For example, *-mbi*, *-me*, *-ci*, *-cibe*, *-ki*, and *-kini* are better to be described within a single affix rule group.

A lexically dependent alternation has to be described by separate rule groups. For example, *h|k|ngk* alternation cannot be described by phonological rules while *a|e|o* alternation is obviously phonologically conditioned:

	a	e	o
h	<i>ara-ha</i>	<i>gene-he</i>	<i>oyo-ho</i>
k	<i>jala-ka</i>	<i>tuhe-ke</i>	<i>foso-ko</i>
ngk	<i>ba-ngka</i>	<i>je-ngke</i>	<i>bo-ngko</i>

An example is presented below. In the dictionary, the verbs are tagged with affix group labels, in other words, the verbs are categorized into three groups: *-ha* verbs, *-ka* verbs, and *-ngka* verbs. The vowel alternation is described within each affix group.

Dictionary:		Affix rules:		
ara/h	foso/k	SFX h Y 3	SFX k Y 3	SFX g Y 3
gene/h	ba/g	SFX h 0 ha a	SFX k 0 ka a	SFX g 0 ngka a
oyo/h	je/g	SFX h 0 he e	SFX k 0 ke e	SFX g 0 ngke e
jala/k	bo/g	SFX h 0 ho o	SFX k 0 ko o	SFX g 0 ngko o
tuhe/k				

Finally, the inflection of a lexical item can be successfully described by the combination of several suffix groups. In the dictionary, the verbs are tagged with a series of affix group labels, for example, *ara-* ‘to make’ is tagged with /Vhfr, which consequentially corresponds to the inflectional group of the verb.

Dictionary:	Affix rules:		
ara/Vhfr	SFX V Y 7	SFX k Y 3	SFX m Y 1
ubaliya/Vkfr	SFX V 0 mbi .	SFX k 0 ka a	SFX m 0 mpi .
fara/Vkpr	SFX V 0 me .	SFX k 0 ke e	
jura/Vkfn	SFX V 0 ci .	SFX k 0 ko o	SFX r Y 3
jala/Vkpn	SFX V 0 cibe .		SFX r 0 ra a
ca/Vgfr	SFX V 0 cina .	SFX g Y 3	SFX r 0 re e
sa/Vgmr	SFX V 0 ki .	SFX g 0 ngka a	SFX r 0 ro o
je/Vgmn	SFX V 0 kini .	SFX g 0 ngke e	
		SFX g 0 ngko o	SFX n Y 3
	SFX h Y 3		SFX n 0 ndara a
	SFX h 0 ha a	SFX f Y 1	SFX n 0 ndere e
	SFX h 0 he e	SFX f 0 fi .	SFX n 0 ndoro o
	SFX h 0 ho o		
		SFX p Y 1	
		SFX p 0 pi .	

A lexical item may take more than two allomorphs. For example, *desere-* ‘to overflow’ and *elde-* ‘to shine’ can take both allomorphs *-fi* and *-pi*: *desere-fi*, *desere-pi*; *elde-fi*, *elde-pi*. Another verb *dule-* ‘to pass’ can take many allomorphs: *dule-re*, *dule-ndere*, *dule-fi*, *dule-pi*, *dule-ke*. The solution is straightforward: attach all applicable affix rules to the word form in the

dictionary file. For example:

desere/Vkfpr
elde/Vkfpr
dule/Vkfprn

4.5 Lexical and morphological information

Hunspell as a morphological analyzer provides functionalities that append the lexical and morphological information to dictionary items and affix rules. Additional information can be tagged with 3-character (two letters and a colon) identifiers.

Below is hunspell output with morphological analysis option. The lexical and morphological information can be categorized by using customizable identifiers, for example, st (stem), po (part-of-speech), en (English translation), is (inflectional suffix). A sentence taken from *Cheongeonogeldae*: is used as input: *gucuse i kesi de eiten babe minde gucihiyerehekv ofi*.

gucuse	st:gucuse	po:noun	en:friends	is:se[plural]
i		po:particle	en:genitive	
kesi	st:kesi	po:noun	en:favor	
de	st:de	po:particle	en:dative	
eiten	st:eiten	po:adj	en:all	
babe	st:ba	po:noun	en:place	is:be[accusative]
minde	st:bi	po:noun	en:to_me	is:de[dative]
gucihiyerehekv	st:gucihiyere	po:verb	en:be_jealous	is:hekv

Below are presented the dictionary file and the affix rule file to get above the output. Dictionary consists of word forms with additional lexical information. Affix rule file consists of rules with additional morphological information. Irregular word forms, for example *mimbe* ‘me (accusative)’, *minde* ‘me (dative)’ and *minci* ‘from me (ablative)’ are better to be listed in the dictionary with lexical and grammatical information. Unproductive plural form, for example *gucuse* ‘friends’ is listed in the dictionary in the below example although it can be analytically described by a suffix rule.

Dictionary:	Affix rules:
ba/N po:noun en:place	SFX N Y 1
de po:particle	SFX N 0 be . is:be[accusative]
eiten po:adj en:all	
gucihiyere/Vh po:verb en:be_jealous	SFX V Y 1
gucu/N po:noun en:friend	SFX V 0 mbi . is:mbi[present]
gucuse/N po:noun en:friends is:se[plural]	
i po:particle en:genitive	SFX h Y 2
kesi/N po:noun en:favor	SFX h 0 hekv e is:hekv
minde st:bi po:noun en:to_me is:de[dative]	SFX h 0 hakv a is:hakv

5. Conclusions

The proofreading tools can make a considerable improvement in the efficiency and accuracy of the digitization of text documents of a minority language. The Manchu speller presented in this article is used as a test bench for the Manchu text digitization environment and proved to be effective for the large-scale digitization project of historical documents. I should also emphasize the benefits of using open-source free software language tools. For implementing a speller, we did not need to learn a programming language nor algorithms but only have to be able to organize a lexicon and morphological rules that are familiar to every linguist.

The Manchu morphological analyzer takes the next top priority task. I proposed a prototype of the morphological analyzer using Hunspell engine and will endeavor to complete the prototype and subsequently to resolve the ambiguity and unknown word problems. The dictionaries and affix rules for Aspell and Hunspell are open to public³³⁾ and will be updated as Manchu text collection grows.

Further works will include an intelligent proofreading system that recognizes lexical and grammatical errors from the context. The current speller detects only orthographic errors within each writing unit while in practice human proofreaders could find other error types: substitution, addition, and deletion. The major problem is the substitution where the original word is replaced with another correctly spelled but contextually wrong word.

33) Available at <<https://github.com/youhyunjo/manchu-spell>>. Users who are interested in obtaining and using the speller are welcome to contact the author.

References

- Alegria, Iñaki, Klara Ceberio, Nerea Ezeiza, Aitor Soroa & Gregorio Hernandez. 2008. Spelling correction: From two-level morphology to open source. *Proceedings of LREC 2008*. 1051-1054. Marrakech, Morocco. ELRA.
- Aronoff, Mark. 1994. *Morphology by itself: Stems and inflectional classes*. Cambridge, Mass: MIT Press.
- Aronoff, Mark & Kristen Fudeman. 2011. *What is morphology?* (2nd ed.) Malden MA: Blackwell.
- Avrorin, Valentin A. 2000. *Grammatika man'chzhurskogo pis'mennogo jazyka* [Grammar of written Manchu language]. Sankt-Peterburg: Nauka.
- Barton, G. Edward. 1985. The computational complexity of two-level morphology. *Proceedings of the 24th annual meeting on association for computational linguistics*. 53-59. Stroudsburg, PA: ACL.
- Beesley, Kennech R. 2004. Morphological analysis and generation: A first-step in natural lanugage processing. *Proceedings of the SALTMIL Workshop at LREC 2004: First steps in language documentation for minority languages*. 1-8. Lisbon, Portugal: SALTMIL.
- Gorelova, Liliya M. (ed). 2002. *Manchu grammar*. Leiden & Boston: Brill.
- Halácsy, Péter, Andás Kornai, László Németh, Andás Rung, István Szakadát & Viktor Trón, V. 2004. Creating open language resources for Hungarian, *Proceedings of LREC 2004*. 1201-1204. Lisbon, Portugal: ERLA.
- Kawachi, Y. & G. Kiyose. 2002. *Manshugo bungo niyumon* [Introduction to literary Manchu]. Kyoto: Kyoto Daigaku gakuzyutsu Shuppankai.
- Koskenniemi, K. 1983. *Two-level morphology: A general computational model for word-form recognition and production*. Helsinki: Department of General Linguistics, University of Helsinki.
- Miłkowski, Marcin. 2010. Developing an open-source, rule-base proofreading tool. *Software: Practice and Experience* 40(7). 543-566. Wiley InterScience.
- Möllendorff, P. G. von. 1892. *A Manchu grammar with analyzed texts*. Shanghai: American Presbyterian Mission Press.
- Németh, László, Viktor Trón, Péter Halácsy, Andás Kornai, Andás Rung & István Szakadát. 2004. Leveraging the open source Ispell codebase for minority language analysis. *Proceedings of the SALTMIL workshop at LREC 2004: First steps in language documentation for minority languages*. 56-59. Lisbon, Portugal: SATMIL.
- Payne, Thomas E. 1997. *Describing morphosyntax: A guide for field linguists*. Cambridge, U.K. & New York, NY: Cambridge University Press.
- Pirinen, T. A. & K. Lindén. 2010. Creating and weighting Hunspell dictionaries as finite-state automata. *Investigationes Linguisticae* 21. 1-16. Poznań: Institute of

- Linguistics, Adam Mickiewicz University.
- Porter, M. F. 1980. An algorithm for suffix stripping. *Program: Electronic library and information systems* 14(3). 130–137. Emerald.
- Pustejovsky, J. 1999. Computational lexicons. In Robert A. Wilson and Frank Keil (eds), *The MIT encyclopedia of the cognitive sciences*. 160-162. Cambridge, Mass: MIT Press. <http://ai.ato.ms/MITECS/Entry/pustejovsky.html> (1 June, 2014)
- Ritchie, Graeme D., Graham J. Russel, Alan W. Black & Stephen G. Pulman. 1992. *Computational morphology: Practical mechanisms for the English lexicon*. Cambridge, Mass: MIT Press.
- Smrž, Otakar & You Hyun-Jo. 2010. Finding the structure of words. In Daniel M. Bikel & Imed Zitouni (eds.), *Multilingual natural language processing applications*, 3-28. Upper Saddle River, NJ: IBM Press.
- Trón, Viktor, György Gyepesi, Péter Halácsy, Andás Kornai, László Németh, & Dániel Varga. 2005. Hunmorph: open source word analysis. *Proceedings of the ACL 2005 workshop on software*. 77-85. Ann Arbor, MI: Association for Computational Linguistics.
- Zakharov, Ivan I. 1879. *Grammatika man'chzhurskogo jazyka* [*Grammar of the Manchu Language*]. Sankt-Peterburg: Nauka.

You Hyun-Jo
 Institute of Humanities
 Seoul National University
 SEOUL 151-745 KOREA
 <youhyunjo@snu.ac.kr>

Received 31 March 2014;
 revision received 27 May 2014;
 accepted 12 June 2014.

