A Finite-State Morphological Analyzer for Wolaytta

Tewodros A. Gebreselassie[1(✉)], Jonathan N. Washington[2], Michael Gasser[3], and Baye Yimam[1]

[1] Addis Ababa University, Addis Ababa, Ethiopia
wolaytta.boditti@gmail.com
[2] Swarthmore College, Swarthmore, USA
jonathan.washington@swarthmore.edu
[3] Indiana University, Bloomington, USA
gasser@indiana.edu

Abstract. This paper presents the development of a free/open-source finite-state morphological transducer for Wolaytta, an Omotic language of Ethiopia, using the Helsinki Finite-State Transducer toolkit (HFST). Developing a full-fledged morphological analysis tool for an under-resourced language like Wolaytta is an important step towards developing further NLP (Natural Language Processing) applications. Morphological analyzers for highly inflectional languages are most efficiently developed using finite-state transducers. To develop the transducer, a lexicon of root words was obtained semi-automatically. The morphotactics of the language were implemented by hand in the lexc formalism, and morphophonological rules were implemented in the twol formalism. Evaluation of the transducer shows as it has decent coverage (over 80%) of forms in a large corpus and exhibits high precision (94.85%) and recall (94.11%) over a manually verified test set. To the best of our knowledge, this work is the first systematic and exhaustive implementation of the morphology of Wolaytta in a morphological transducer.

Keywords: Wolaytta language · Morphological analysis and generation
HFST · Apertium · NLP

1 Introduction

This paper describes the development of Free/Open-Source morphological analyzer and generator for Wolaytta, an Omotic language of Ethiopia with almost no computational resources. This tool was created as part of the research for developing a framework for exploiting cross-linguistic similarities in learning the morphology of under-resourced languages.

In language technology research, morphological analysis studies how the internal structure of words and word formation of a language can be modelled computationally. Word analysis involves breaking a word into its morphemes, the smallest forms paired with a particular meaning [1, 14]. The function of a morphological analyzer is to return a lemma and information about the morphology in a word. A morphological generator

F. Mekuria et al. (Eds.): ICT4DA 2017, LNICST 244, pp. 14–23, 2018.
https://doi.org/10.1007/978-3-319-95153-9_2

does exactly the reverse of this; i.e., given a root word and grammatical information, a morphological generator will generate a particular form of a word [2]. Morphological analysis is a key component and a necessary step in nearly all natural language processing (NLP) applications for languages with rich morphology [2]. The output of morphological analysis can be used in many NLP applications, such as machine translation, machine-readable dictionaries, speech synthesis, speech recognition, lexicography, and spell checkers especially for morphologically complex languages [3].

In this work, we have considered the standard written Wolaytta text and used Helsinki Finite State Toolkit and tools from Apertium to build the morphological analyzer. All of the resources prepared for the development of the Wolaytta morphological transducer, including the lexicon, the morphotactics, the alternation rules, and the 'gold standard' morphologically analysed word list of 1,000 forms are all freely available online under an open-source license in Apertium's svn repository[1]. This paper is organized as follows. Section 2 briefly reviews the literature on morphological analysis generally and morphological analysers implemented in a similar way to the one described in this paper. Section 3 provides a brief overview of the Wolaytta language. The implementation of the morphological analyzer follows in Sect. 4. Section 5 then covers the evaluation and results. Finally, the paper concludes in Sect. 6 with some discussion of future research directions.

2 Literature Review

The importance of the availability of a morphological analyzer for NLP application development is reviewed by different researchers. Malladi and Mannem [19] stated that NLP for Hindi has suffered due to the lack of a high-coverage automatic morphological analyzer. Agglutinative languages such as Turkish, Finnish, and Hungarian require morphological analysis before further processing in NLP applications due to the complex morphology of the words [20]. In machine translation for highly inflectional (morphologically complex) and resource-limited languages, the presence of a morphological analyzer is crucial to reduce data sparseness and improve translation quality [2, 22]. It is with this reality that there exist fully functional morphological analyzers for languages like English, Finnish, French, etc.

Since Kimmo Koskenniemi developed the two-level morphology approach [15], several approaches have been attempted for developing morphological analyzers. The rule-based approach is based on a set of hand-crafted rules and a dictionary that contains roots, morphemes, and morphotactic information [14, 16, 17]. In this approach, the morphological analysis requires the existence of a well-defined set of rules to accommodate most of the words in the language. When a word is given as an input to the morphological analyzer and if the corresponding morphemes are missing in the dictionary, then the rule-based system fails [15].

[1] Available at: https://svn.code.sf.net/p/apertium/svn/incubator/apertium-wal/.

2.1 Related Work

The transducer for Wolaytta presented in this paper was developed using a rule-based approach, implemented using a Finite State Transducer (FST). As outlined in some of the sources below, the finite state methodology is sufficiently mature and well-developed for use in several areas of NLP. Other works overviewed show the application of finite-state transducers to other Afroasiatic languages.

Among languages of Ethiopia, there is some research on developing morphological analyzers, including for Amharic [2, 3, 21], Afan Oromo [2] and Tigrigna [2]. Amharic and Tigrigna are classified as Semitic languages, and Afan Oromo is classified as a Cushitic language. One of the most well-known of these is HornMorpho [2], which is accessible online. HornMorpho is a system for morphological processing of the most widely spoken Ethiopian languages—Amharic, Oromo, and Tigrinya—using finite state transducers. For each language, it has a lexicon of roots derived from dictionaries of each language. To evaluate the system, words from different parts of speech are selected randomly from each word list. The system shows 96% accuracy for Tigrinya verbs and 99% accuracy for Amharic verbs.

Washington et al. [9] describes the development of a Free/Open-Source finite-state morphological transducer for Kyrgyz using the Helsinki Finite-State Toolkit (HFST). The paper described issues in Kyrgyz morphology, the development of the tool, some linguistic issues encountered and how they were dealt with, and issues left to resolve. An evaluation is presented showing that the transducer has medium-level coverage, between 82% and 87% on two freely available corpora of Kyrgyz, and high precision and recall over a manually verified test set. In the other work using the same formalism, Washington et al. [23] describe the development of Free/Open-Source finite-state morphological transducers for three more Turkic languages—Kazakh, Tatar, and Kumyk—also using HFST. These transducers were all developed as part of the Apertium project, which is aimed at creating rule-based machine translation (RBMT) systems for lesser resourced languages. This paper describes how the development of a transducer for each subsequent closely-related language took less development time because of being able to reuse large portions of the morphotactic description from the first two transducers. An evaluation is presented shows that the transducers all have a reasonable coverage around 90% on freely available corpora of the languages, and high precision over a manually verified test set.

Yona and Wintner [18] describe HAMSAH (HAifaMorphological System for Analyzing Hebrew), a morphological processor for Modern Hebrew, based on finite-state linguistically motivated rules and a broad coverage lexicon. The set of rules comprehensively covers the morphological, morpho-phonological and orthographic phenomena that are observable in contemporary Hebrew texts. They show that reliance on finite-state technology facilitates the construction of a highly efficient and completely bidirectional system for analysis and generation.

3 Morphology of the Language

Wolaytta belongs to the Omotic language family, which is a branch of the Afroasiatic language phylum, and is spoken in the Wolaytta Zone and some other parts of the Southern Nations, Nationalities, and People's Region of Ethiopia [4]. Wolaytta has had a formal orthography since the 1940s, and is written in the Latin alphabet. A Bible was published in Wolaytta in 1981 [5].

Wolaytta is an agglutinative language and word forms can be generated from root words by adding suffixes. From a single root word, many word forms can be generated using derivational and inflectional morphemes. The order of added morphemes is governed by the morphotactic rules of the language. While suffixation is the most common word formation strategy in Wolaytta [6], compounding is also used [5].

In forming a word, adding one suffix to another, or "concatenative morphotactics", is an extremely productive element of Wolaytta's grammar [24]. This process of adding one suffix to another suffix can result in relatively long word forms, which often contain the amount of semantic information equivalent to a whole English phrase, clause or sentence. For example, "7imisissiis" is one word form in Wolaytta, which is equivalent to the expression in English "He caused someone to make someone else cause giving something to someone else". When we analyze this word, it consists of 7im-is-iss-iis give-CAUS.CAUS.-PF.3 M.SG. Due to this complex morphological structure, a single Wolaytta word can give rise to a very large number of parses.

The second word formation process in Wolaytta is compounding. Compounding is the process in which two or more lexemes combine into a single new word [6]. Although Wolaytta is very rich in compounds, compound morphemes are rare in Wolaytta and their formation process is irregular. As a result, it is difficult to determine the stem of compounds from which the words are made [5].

Wolaytta nouns are inflected for number, gender and case. According to Wakasa [4], common nouns in Wolaytta are morphologically divided into four subclasses, three of which are masculine and one of which is feminine. Place-name and personal nouns are inflected differently from common nouns. Numerals are morphologically divided into four subclasses. They inflect according to case, and concrete forms (singular and plural) of the common noun can be derived from them. Verbs in Wolaytta are inflected for person, number, gender, aspect and mood. Wolaytta has two genders (masculine and feminine), two numbers (singular and plural), three persons (first, second and third), and five cases (absolutive, oblique, nominative, interrogative, and vocative).

In terms of derivational processes, a common noun stem may be derived from a common noun stem or a verb stem by adding a suffix that has a particular function. In the same way, a verb stem may be derived from a common noun stem.

4 Implementation of the Morphological Analyzer

The modeling and implementation of the morphology is designed based on the popular Helsinki Finite State Toolkit (HFST), which is a free/open-Source reimplementation of the Xerox finite-state toolchain [9]. HFST provides a framework for compiling and applying linguistic descriptions with finite state methods and is used for efficient

Table 1. Words in their lexical and surface forms

Lexical form	Surface form
d-. -NOM.M.SG.	d-ées
wooss-. -PF.3 M.SG.	wooss-íis
m-. -PF.3 M.SG.	miss-íis
7im-.-CAUS.-CAUS.-PF.3 M.SG	7im-is- iss-íis

Table 2. Number of stems in each of the main categories

Part of speech	**Number of stems**	**Example representation in lexc**
Noun	4,628	`LEXICON NounRoot` `aahotett:aahotett N-M-CMN-A ; ! "wid`
Verb	2,609	`LEXICON VerbRoot` `aac:aac V-IV ; ! "sprout"`
Numeral	12	`LEXICON NumberRoot` `iss:iss NUM-1 ; ! "1"`
Pronoun	22	`LEXICON Pronouns` `ta:ta PRONOUN-1-S ; ! "I/me"`
Punctuation	16	`LEXICON Punctuation` `%.%<sent%>:%. # ;`
Adjectives	2,242	`LEXICON Adjectives` `aammotida:aammotida ADJ ; ! "rotten,`
Adverb	368	`LEXICON Adjectives` `aayyee'ana:aayyee'ana ITJ ; !`
Interjection	136	`LEXICON Pronouns` `ta:ta PRONOUN-1-S ; ! "I/me"`
Preposition	16	`LEXICON Prepositions`
Postposition	40	`LEXICON POSTPOS` `%+yyo%<post%>:%>yyo CMN-To-All ; !`
Connection	26	`LEXICON Connection` `gishshaw:gishshaw CONN ; ! "because"`
Nominalizer	21	`LEXICON Connection` `%<nmlz%>%<sing%>%<masc%>%<abs%>:gaa;`
Total	10,136	

language application development [9]. HFST has been used for creating morphological analyzers and spell checkers using a single open-source platform and supports extending and improving the descriptions with weights to accommodate the modeling of statistical information [11]. It implements both the *lexc* formalism for defining lexicons, and the *twol* and *xfst* formalisms for modeling morphophonological rules which describe what changes happen when morphemes are joined together.

FSTs are a computationally efficient, inherently bidirectional approach that distinguishes between the surface and lexical realizations of a given morpheme and attempts to establish a mapping between the two. It can be used for both analysis (converting from word form to morphological analysis) and generation (converting from morphological analysis to word form) [10, 13]. Table 1 below shows examples of lexical and surface form representations for sample Wolaytta words in the two-level morphology.

While building the Wolaytta morphological analyzer using HFST, the following information is used: a lexicon of Wolaytta words, morphotactics, and orthographic rules.

The lexicon is the list of stems and affixes together with basic information about them (Noun stem, Verb stem, etc.,). One of the challenges to develop natural language processing applications for languages like Wolaytta is the unavailability of digital resources. There are no available digital resources, like corpora, for Wolaytta. The Wolaytta lexicon was extracted semi-automatically from an unpublished Wolaytta-English bilingual dictionary and other printed reference books written for academic purposes. The data in Table 2 shows the part of speech, the number of stems in the lexicon of that part of speech, and an example of how the data is represented in the system.

Morphotactics is a model of morpheme ordering that explains which classes of morphemes can follow other classes of morphemes inside a word [10]. The lexicon and morphotactics are defined in the HFST-lexC compiler, which is a program that reads sets of morphemes and their morphotactic combinations in order to create a finite state transducer. Using HFST, morphophonology is mostly dealt with by assigning special segments in the morphotactics (lexc) which are used as the source, target, and/or part of the conditioning environment for twol rules [10]. In lexc, morphemes are arranged into named sets called sub-lexicons. As shown in Fig. 1, each entry of a sub-lexicon is a pair of finite possibly empty strings separated by ":" and associated with the name of a sub-lexicon called a continuation class.

```
LEXICON Root
            VerbRoot ;
LEXICON VerbRoot
            aac:aac V-IV ; ! "sprout"
LEXICON V-IV
            %<v%>%<iv%>:%> VERB-INFL ;
LEXICON VERB-INFL
            VERB-Imfr ;
LEXICON VERB-Imfr
            %<p3%>%<sing%>%<fem%>%<imfr%>:awsu # ;
```

Fig. 1. Example lexicons representing a single path, for the form *aacawsu*.

One of the challenging tasks is identifying the existing roots and suffixes of each word in all the word classes, since the available linguistic studies of the language are limited. For this language, the most useful study is that of Wakasa [4], which we used to categorize the collected lexicons from the dictionary into different classes based on their morphological characteristics.

Morphophonological and orthographic rules are spelling rules used to model the changes that occur in a word when two morphemes combine. The orthographic rules for the Wolaytta language in the HFST architecture are written in the HFST-TwolC formalism. HFST-TwolC rules are parallel constraints on symbol-Pair strings governing the realizations of lexical word forms as corresponding surface strings. HFST-TwolC is an accurate and efficient open-source two-level compiler. It compiles grammars of two level rules into sets of finite-state transducers. Identifying and writing the existing rules manually is a real difficulty for under-resourced languages like Wolaytta. Even when a resource such as Wakasa [4] exists, it may fail to express all relevant conditions. Some of the rules in the Wolaytta morphological analyzer are shown in Fig. 2.

```
1.  "change {V} to next vowel"
      %{V%}:Vy<=> _ :0* :Vy ;
      except _ :0* :Vy :Vow ;
      where Vy in Vow ;

2.  "degeminate consonant before -iss"
      Cx:0 <=> _ :Cx :0* :i :s :s ;
      Except   _ :Cx %{‼%}: ;
               _ :c ; ! avoid conflict with cc >sh
      where Cx in Cns ;

3.  "first c>s when cc >sh before -iss"
      c:s<=> _ c: :0* :i :s :s ;
      except _ c:0 ; ! avoid conflict with degemination, etc.

4.  "second c>h when cc >sh before -iss"
      c:h <=> c: _ :0* :i :s :s ;
```

Fig. 2. Example morphophonological/orthographic rules for Wolaytta in the twol formalism.

The symbol ! indicates comments; % is an escape character, and archiphonemes are in {}. Whenever there are exceptions, the archiphoneme %{‼%} (which is always deleted in the output) is used to block phonology from applying.

5 Analysis and Evaluation

As mentioned before, the system is implemented using Helsinki finite state tools. Morphotactic rules and possible morphemes are defined in the lexicon file. Alternation rules of Wolaytta verbs are defined and the rules are composed with the lexicon file in a HFST-twol file. The system works in two directions, between the lexical and surface levels.

We have prepared a Wolaytta sentence corpus from the Wolaytta-English bilingual dictionary. Identifying the existing Wolaytta-only sentences requires lots of manual work in line with the programs written to identify Wolaytta-only sentences. One of the difficulties is confusion with words that can also be English (E.g. “He” refers “This” in Wolaytta).

Table 3. Results: overall coverage

Total no. tokenized words in the corpus	38,479
% Recognized words	83.13
% Unrecognized words	16.87
Translation time	0.96 S

As listed in Table 3 above, 16.87% of words are not recognized by the Wolaytta morphological analyzer. Since most Wolaytta texts use the apostrophe character (U +0027) to represent the glottal stop instead of the more proper modifier letter apostrophe (U+02BC), most words with glottal stops are unrecognised. Among the top twenty unrecognized words, more than 75% are words with glottal stop characters. The remaining words fall into out-of-vocabulary words (mostly proper nouns) and noise. The lexicon is collected mostly from the Wolaytta-English dictionary. Adding more lexical entries collected from different domains to the system could further improve the coverage.

To evaluate the accuracy of the system, one thousand forms were chosen at random from a corpus of approximately 38K Wolaytta words. These forms were tokenised and hand-annotated, creating a gold standard. When compared against the output of the transducer, precision (the percentage of returned analyses that are correct) is 94.85% and recall (the percentage of correct analyses that are returned) is 94.11%.

6 Conclusions and Future Work

We described the construction of the first known morphological analyzer for Wolaytta using HFST and the Apertium framework. This morphological analyzer acts as a preliminary step to achieving relevant output for the applications like spell checking, text mining, text summarization, etc., by providing analyses of word forms. This morphological transducer can also easily be used to for developing a machine translation system for Wolaytta-English since our system is already incorporated into Apertium.

To develop a fully functional analyzer, the lexicon needs to be exhaustive and rich in morpho-syntactic information, and it is necessary to write additional phonological rules to cover all cases where they are needed. Our analyzer can handle inflectional and derivational morphology for native Wolaytta words, but so far not for loan words. In future work, analysis for other categories needs to be handled by adding exceptions for widely used loan words to existing rules. Moreover, we the working system is available on the web to anyone interested in further enhancing the analyzer or in need of a Wolaytta transducer for use in their own application development.

References

1. Allen, J.: Natural language understanding (1987)
2. Gasser, M.: HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya. In: Conference on Human Language Technology for Development, Alexandria, Egypt (2011)
3. Mulugeta, W., Gasser, M.: Learning morphological rules for Amharic verbs using inductive logic programming. Lang. Technol. Normalisation Less-Resourced Lang. **7** (2012)
4. Wakasa, M.: A descriptive study of the modern Wolaytta language. Unpublished Ph.D. thesis, University of Tokyo (2008)
5. Lamberti, M., Roberto, S.: The Wolaytta Language, vol. 6. Rudiger Koppe, Cologne (1997)
6. Lessa, L.: Development of stemming algorithm for Wolaytta text. Diss. aau (2003)
7. Bosch, S.E., Pretorius, L.: A finite-state approach to linguistic constraints in Zulu morphological analysis. Studia Orientalia Electronica **103**, 205–228 (2015)
8. Beesley, K.R., Karttunen, L.: Finite State Morphology. Center for the Study of Language and Information (2003)
9. Washington, J., Ipasov, M., Tyers, F.M.: A finite-State morphological transducer for Kyrgyz. In: LREC (2012)
10. Martin, J.H., Jurafsky, D.: Speech and Language Processing, International Edition 710 (2000)
11. Linden, K., Axelson, E., Hardwick, S., Silfverberg, M., Pirinen, T.: HFST—framework for compiling and applying morphologies. In: Mahlow, C., Pietrowski, M. (eds.) State of the Art in Computational Morphology. Communications in Computer and Information Science, vol. 100, pp. 67–85. Springer, Berlin Heidelberg (2011). https://doi.org/10.1007/978-3-642-23138-4_5
12. Lindén, K., Silfverberg, M., Pirinen, T.: Hfst tools for morphology—an efficient open-source package for construction of morphological analyzers. In: Mahlow, C., Pietrowski, M. (eds.) State of the Art in Computational Morphology. Communications in Computer and Information Science, vol. 41, pp. 28–47. Springer, Berlin Heidelberg (2009). https://doi.org/10.1007/978-3-642-04131-0_3
13. Karttunen, L.: Finite-state lexicon compiler. Technical report ISTL-NLTT-1993-04-02, Xerox Palo Alto Research Center, Palo Alto, California (1993)
14. Oflazer, K.: Two-level description of Turkish morphology. In: Proceedings of the Sixth Conference on European Chapter of the Association for Computational Linguistics, EACL 1993, p. 472. Association for Computational Linguistics, Stroudsburg (1993)
15. Koskenniemi, K.: A general computational model for word form recognition and production. In: Proceedings of the 10th International Conference on Computational Linguistics, pp. 178–181. Association for Computational Linguistics (1984)
16. Grac, M.: Yet another formalism for morphological paradigm. In: Recent Advances in Slavonic Natural Language Processing, RASLAN 2009, p. 9 (2009)
17. Oflazer, K., Kuruoz, I.: Tagging and morphological disambiguation of Turkish text. In: Proceedings of the Fourth Conference on Applied Natural Language Processing, ANLC 1994, pp. 144–149. Association for Computational Linguistics, Stroudsburg (1994)
18. Yona, S., Wintner, S.: A finite-state morphological grammar of Hebrew. Nat. Lang. Eng. **14** (02), 173–190 (2008)
19. Malladi, D.K., Mannem, P.: Context based statistical morphological analyzer and its effect on Hindi dependency parsing. In: Fourth Workshop on Statistical Parsing of Morphologically Rich Languages, vol. 12, p. 119 (2013)

20. Eray Yildiz, C., Bahadir Sahin, H., Mustafa Tolga Eren, O.: A morphology-aware network for morphological disambiguation (2016)
21. Amsalu, S., Gibbon, D.: Finite state morphology of Amharic. In: Proceedings of RANLP (2005)
22. Goldwater, S., McClosky, D. Improving statistical MT through morphological analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 676–683. Association for Computational Linguistics (2005)
23. Washington, J., Salimzyanov, I., Tyers, F.M.: Finite-state morphological transducers for three Kypchak languages. In: Proceedings of LREC, pp. 3378–3385 (2014)
24. Beesley, K.R., Karttunen, L.: Finite-state non-concatenative morphotactics. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pp. 191–198. Association for Computational Linguistics (2000)