Portable Spelling Corrector for a Less-Resourced Language: Amharic

Andargachew Mekonnen Gezmu, Andreas Nürnberger, Binyam Ephrem Seyoum
Otto-von-Guericke-Universität Magdeburg, Addis Ababa University
Otto-von-Guericke-Universität Magdeburg,
Fakultät für Informatik,
Data and Knowledge Engineering Group,
Universitätsplatz 2,
39106 Magdeburg.
{andargachew.gezmu, andreas.nuernberger}@ovgu.de, binyam.ephrem@aau.edu.et

Abstract
This paper describes an automatic spelling corrector for Amharic, the working language of the Federal Government of Ethiopia. We used a corpus-driven approach with the noisy channel for spelling correction. It infers linguistic knowledge from a text corpus. The approach can be ported to other written languages with little effort as long as they are typed using a QWERTY keyboard with direct mappings between keystrokes and characters. Since Amharic letters are syllabic, we used a modified version of the System for Ethiopic Representation in ASCII for transliteration in the like manner as most Amharic keyboard input methods do. The proposed approach is evaluated with Amharic and English test data and has scored better performance result than the baseline systems: GNU Aspell and Hunspell. We get better result due to the smoothed language model, the generalized error model and the ability to take into account the context of misspellings. Besides, instead of using a handcrafted lexicon for spelling error detection, we used a term list derived from frequently occurring terms in a text corpus. Such a term list, in addition to ease of compilation, has also an advantage in handling rare terms, proper nouns, and neologisms.

Keywords: spelling corrector, corpora, noisy channel, less-resourced language

1. Introduction

Documents in many languages have been digitized and are available in different media especially on the web. Giant software vendors (e.g., Google and Microsoft) are also localizing their products to the native languages of their target customers. There is a need to develop computational solutions to the classic problems of computational linguistics for the respective languages. Spelling error detection and correction are among the oldest computational linguistics problems (Blair, 1960). Spelling correction is considered from two perspectives: non-word and real-word correction. When typographical or cognitive errors accidentally produce valid words we get real-word errors, otherwise, we get non-word errors. These problems are mostly treated separately. In this paper, we dealt with non-word errors.

Many spelling correctors are developed using rule-based approaches. However, it is difficult to develop and maintain all language-dependent rules (Norvig, 2009). In addition, such systems limit themselves to isolated-word correction without considering the context. Therefore, we proposed and evaluated an approach that takes into account the context of misspellings and infers linguistics knowledge from a text corpus.

2. Related Work

Earliest research on spelling correction is based on phonetic and string similarities such as Metaphone and Damerau-Levenshtein edit distance algorithms (Damerau, 1964). Candidate corrections are ranked from manually compiled lexicons with the help of these algorithms. GNU Aspell and Hunspell are good examples that follow this approach. Mekonnen (2012) has followed the same approach for Amharic. In a related approach, Ahmed et al. (2009) used similarity scores of letter n-grams to rank candidate corrections. In these approaches, the lexicons along with some linguistics rules are used for spelling error detection. Yet there was also an attempt to detect errors without using lexicons (Morris and Cherry, 1975). This approach depends on n-gram letter-sequences from a target text. It generates an "index of peculiarity"; and based on the index, determines which words are spelling errors in the target text. For example, the typo 'exmination' contains 'exm' and 'xmi', trigrams which are peculiar and will be included in the list. Even though this approach has the advantage of being language independent and appears to work for less-resourced languages, many misspellings do not contain the unusual n-grams and so would not appear in the list (Mitton, 2010).

Recent research on spelling error correction focuses on using some web corpora to infer knowledge about spellings. Most of these systems are based on the noisy channel model (Kernighan et al., 1990; Kukich, 1992; Brill and Moore, 2000; Whitelaw et al., 2009; Gao et al., 2010). Also, additional features of spellings such as phonetic similarities and modified edit distance (e.g., Winkler (2006)) are used to generate plausible candidates for spelling correction (Toutanova and Moore, 2002).

3. Approach

Like other Semitic languages, word formation in Amharic depends mainly on root-and-pattern morphology and exhibits prefixes, suffixes, and infixes. Amharic is morphologically-rich in the way that grammatical relations and syntactic information are indicated at the word level. These features are some of the main hurdles for rule-based computational processing (Fabri et al., 2014). It is difficult to develop and maintain all language-dependent rules for spelling correction especially when the languages have complex morphology like Amharic (Norvig, 2009). Thus, we have applied a data-driven (corpus-driven) approach with the noisy channel for spelling correction. According to the noisy channel approach, for a misspelled word x, the most likely candidate correction w_n out of all possible candidate corrections C with $w_1w_2...w_{n-1}$ preceding words context is suggested by the maximum probability of $P(w_n|w_1w_2...w_{n-1}x)$, which is computed by Equation 1

below. $P(w_1w_2 \ldots w_{n-1}w_n)$ is the prior probability and $P(x|w_n)$ the likelihood where both are represented in the language and error models; see sections 3.1 and 3.2 for details. Obviously, x is conditionally dependent only on w_n and assumes the preceding words are correct.

$$\text{argmax}_{w_n \in C} P(w_1w_2 \ldots w_{n-1}w_n)P(x|w_n) \qquad (1)$$

Based on the proposed approach, the spelling error detection and correction processes are as follows. An input word that is not in the term list, which is compiled from the most frequent words in a text corpus, is flagged as a spelling error. Candidate corrections that are closer (nearer) to the misspelling are generated from the term list. For language independence, we measure nearness using Damerau-Levenshtein edit distance (Damerau, 1964). Since most of the misspellings fall within two edit distance from their corrections (Damerau, 1964; Gezmu et al., 2017), we selected all words in the term list that are one up to two edit distance from the misspelled word. Then the candidates are scored and ranked according to their prior and likelihood probabilities. In case there is no candidate correction, the misspelled term will be split. This step is needed to correct misspellings result from missed out spaces between words, like የሃንስነገይመጣል. The correction is to segment the expression as የሃንስ /johannɨs/[1], ነገ /nəgə/ and ይመጣል /jɨməṭ'al/.

3.1 Language Model and Corpora

In a text corpus, linguistic knowledge resides in the n-grams of the corpus and it is possible to acquire new knowledge using a large number of documents. It also contains rare terms, proper nouns, technical terms, brand names, and newly coined terms (neologisms). Manually compiled or handcrafted lexicons normally do not include most of these types of terms. But by using the most frequent words in the corpus, we can easily create a term list that incorporates the most widely used terms.

Tachbelie and Menzel (2007) evaluated n-gram word-based and morpheme-based Amharic language models. They have used a news corpus to build the models. The word-based model smoothed with the Kneser-Ney (Kneser and Ney, 1995) method has a better performance on a test data than the morpheme-based model. The result complies with the empirical study of Chen and Goodman (1998). The researchers have found that among the best performing n-gram smoothing methods is the Kneser-Ney with its modified version. To this end, we build a trigram word-based language model smoothed with the modified Kneser-Ney method.

For Amharic language model, being a less-resourced language, the only available sizable text corpora are HaBiT (HaBiT, 2016; Rychlý and Suchomel, 2016) and Crúbadán (Scannell, 2007). Both are created from automatically crawled web pages. HaBiT contains about 17.6 million tokens (words) whereas Crúbadán contains about six million tokens. Except for their size difference, both corpora are essentially the same. Since they obtain text from all types of web documents, we expected spelling errors in these corpora. We have found out that they contain a number of spelling errors through a manual check. Therefore, we build our own Contemporary Amharic Corpus (CACO) of about 19 million tokens from sources which we assumed are proofread. We have also used HaBiT for comparison.

The CACO is compiled from various sources that are published since the mid of twentieth century. It was collected from publicly available archives of three Amharic newspapers (አዲስ አድማስ, አዲስ ዘመን, and ሪፖርተር), two magazines (ንቁ and መጠበቂያ ግንብ), eight fictions (ኦሮማይ, የልምዣት, አልወለድም, ግርዶሽ, ልጅነት ተመልሶ አይመጣም, የአመጽ ኑዛዜ, የቅናት ዛር, and አግዐዚ), four historic novels (አሉላ አባነጋ, ማዕበል የአብዮቱ ማግሥት, የማይጨው ቁስለኛ, and የታንጉት ሚስጢር), two short novels (የዓለም መስታወት and የቡና ቤት ስዕሎችና ሌሎችም ወጎች), five history books (አጭር የኢትዮጲያ ታሪክ, ዳግማዊ አጤ ምኒልክ, ዳግማዊ ምኒልክ, የእቴጌ ጣይቱ ብጡል (፲፰፻፴፪ - ፲፱፻፲) አጭር የሕይወት ታሪክ, and ከወልወል እስከ ማይጨው), two politics books (ማርክሲዝምና የቋንቋ ችግሮች and መሬት የማን ነው), and two children books (ፒኖኪዮ and ውድድር). In addition, Amharic news articles and legal documents (ነጋሪት ጋዜጣ) from ELRA-W0074 (2014), news articles from Ethiopian News Agency, and the Amharic Bible[2] are used.

Paragraphs from the body of the documents are extracted. Then the paragraphs are transliterated to Latin-based characters using a modified version of the System for Ethiopic Representation in ASCII (SERA) (Yitna and Yacob, 1997). The modification is in transliterating labiovelars, which represent consonants followed by a back low diphthong ʷa, and vowels that are written independently. For example, the labiovelar ቧ /bʷa/ and the vowel ኡ /ʔu/ using the original SERA is transliterated as *bWa* and *'u* but with the modified version as *bua* and *u*, respectively. The same modification is adapted for ease of typing by the popular Amharic keyboard input methods such as Google's and Keyman's. Besides, four of Amharic phonemes have one or more homophonic character representations and there are other peculiar labiovelars (e.g., ቐ /kʰʷ/, ጒ /gʷi/, and ጔ /gʷe/). In the contemporary Amharic writings, the homophonic characters are commonly observed to be used interchangeably and there is no uniform use of the peculiar labiovelars. For consistent spelling, the Ethiopian Languages Academy (ELA) proposed a spelling reform (ELA, 1970; Aklilu, 2004). Following their reform, homophonic characters are merged into their common forms; ሐ and ኀ are replaced with ሀ, ሠ with ሰ, ዐ with አ, and ፀ with ጸ. The replacement includes their variant forms. This process can be considered as case folding in English (Yacob, 2003). We have normalized the peculiar labiovelars by substituting them with their closer counterparts (e.g., ቐ /kʰʷ/ withቈ /kʰu/). However, unlike, the spelling reform we preferred ጸ to ፀ; and kept ኧ and ዀ because in many input methods they are easily accessible and are commonly found in Amharic writings.

After the transliteration of the paragraphs, numbers are replaced by a placeholder (e.g., "በ1990 ዓ. ም." is preprocessed as "be # a m"); hyphenated words are split (e.g., "ስነ-ስርዓት" as "sne sr'at"); unique sentences are identified and extracted by their boundaries either double colon-like symbols (።) or question marks (? or ፧); and are tokenized based on orthographic-word boundaries, a white space or a colon-like symbol (፡).

[1] The International Phonetic Alphabets (Hayward and Hayward, 1992; IPA, 2015) are written only for the sake of readability.

[2] We used the New World Translation of the Bible which is translated into the contemporary (not archaic) Amharic.

To train the English language model, all sentences from British National Corpus (BNC) are extracted (BNC XML Edition, 2007). In order to equate the preprocessing steps of both languages, the sentences are preprocessed in a manner similar to Amharic corpora as follows: as case folding letters are lowercased, numbers are replaced by a placeholder # symbol, hyphenated words are split, contracted forms (clitics) are conflated (e.g., is n't into isn't), part-of-speech tags are discarded, and they are tokenized based on white space. Table 1 shows the number of sentences and tokens in each corpus.

Furthermore, through a manual check, we have analyzed that terms which appear only once in the respective corpora are mostly misspelled. Before we build the language model for each corpus, as a further preprocessing step, we have deleted sentences that contain words that occur only once in the entire corpus.

	CACO	HaBiT	BNC
No. of sentences	1,335,446	1,197,880	5,847,803
No. of tokens	18,933,305	17,605,866	97,111,951

Table 1: The number of sentences and tokens in the CACO, HaBiT, and BNC corpora.

The corpora statistics after the final preprocessing step is shown in Table 2.

	CACO	HaBiT	BNC
No. of sentences	1,010,590	873,426	5,690,343
No. of unigrams	366,654	350,789	228,999
No. of bigrams	5,811,598	4,986,029	11,008,294
No. of trigrams	9,996,057	8,302,152	40,168,232

Table 2: The corpora statistics after preprocessing.

The language models are trained using the KenLM language modeling toolkit (Heafield et al., 2013). The models are saved in the binary ARPA format for efficiency. The prior probability $P(w_1w_2\ldots\ w_{n-1}w_n)$ for trigram language model is estimated by Equation 2, based on chain rule of probability and Markov's assumption. The log probabilities that are used to compute this conditional probability along with backoff weights have been precomputed and are stored in the ARPA file language models.

$$\prod_{i=1}^{n} P(w_i|w_{i-2}w_{i-1}) \quad (2)$$

3.2 Error Model

Most Amharic characters are syllabary (Bloor, 1995; Unicode Consortium, 2017). For instance, በ /bə/, ቡ /bu/, ቢ /bi/, ባ /ba/, ቤ /be/, and ቦ /bo/ are all syllabic scripts with CV pattern. They conflate consonants and vowels even if they are typed with QWERTY keyboard input methods with direct mappings between keystrokes and characters. Hence there is a need to separate the two components to properly model spelling errors. This is done by transliteration of the letters into Latin alphabets by using the modified version of the SERA.

To train the error model, there is no sizable Amharic spelling error corpus. But we have made an assumption: as Amharic scripts are typed with English QWERTY keyboard, the key slips that cause spelling errors in English and Amharic are related. So, a substring based English spelling error model that represents the likelihood probability, $P(x|w_n)$, is useful for languages that can be transliterated into Latin alphabets. Such an error model is created by Norvig (2009) based on forty thousand spelling errors. Since this approach suits our need, we have adapted the error model.

3.3 Term Splitting

For spelling errors resulting from missed out spaces, term splitting is necessary. To generate candidate corrections for a spelling error, the expression was segmented to all possible valid words using a term list. Then using a language model a prior probability to each candidate was assigned. The candidate which has the highest probability is the plausible spelling correction. For example, Table 3 demonstrates how to split the above-mentioned example (i.e., ዮሃንስነገይመጣል transliterated with the modified SERA as *yohansnegeymeTal*), using the CACO language model and the corresponding term list. The probability of *yohans nege ymeTal* is the highest of all. Thus, the expression is split as such and transliterated back into Amharic as ዮሃንስ ነገ ይመጣል.

Candidates	Log 10 probability	Probability
yo hans nege ymeTal	-19.15934944	$6.92868 * 10^{-20}$
yoha ns nege ymeTal	-20.61217499	$2.44245 * 10^{-21}$
yohan s nege ymeTal	-19.17063332	$6.75098 * 10^{-20}$
yohans nege ymeTal	-11.64624405	$2.25817 * 10^{-12}$

Table 3: Example of a term splitting.

4. Evaluation

To evaluate the performance of our approach and to demonstrate its easy portability to other languages, first we made an evaluation based on Amharic test data and compared the results with the baseline systems: GNU Aspell and Hunspell; and then we performed an evaluation on English. For evaluation of spelling error detection capability, precision, recall, and F1 measure were used as metrics; and the relative positions of the correct spellings in the plausible suggestions list were used to evaluate spelling error correction. To interface with Aspell and Hunspell we used the PyEnchant[3] with their latest dictionaries available for both languages.

4.1 Test Data

We used manually annotated spelling error test corpora for evaluation. For Amharic we used a test corpus compiled by Gezmu et al. (2017)[4]; and for English the one that was compiled by Mitton (1985) from the book "English for the Rejected" (Holbrook, 1964) which is available in the Oxford Text Archive. Even though this one was originally handwritten by poor spellers, its contextual information makes it still useful for evaluation purposes.

For Amharic test data, 367 sentences were tagged with 287 non-word spelling errors, but 35 of the non-word misspellings appear twice in the documents with different

[3] Available at https://pypi.python.org/pypi/pyenchant/

[4] The annotated test corpus is available at the appendix of Gezmu et al. (2017).

contexts. For the sake of making a comparative evaluation with the baseline systems, 252 of the unique non-word misspellings were used in the evaluation. Removal of the duplicates is needed because the baseline systems do not use the context of the misspellings and two similar misspellings are identical test cases for them. In the English test data, 1,043 unique non-word errors were used, including one misspelling "o clock" which was not tagged by mistake in the original test corpus.

4.2 Evaluation Metrics

The evaluation metrics are from the perspectives of spelling error detection capability and the quality of plausible suggestions offered to each spelling error.

Spelling error detection capabilities are evaluated by precision, recall, and F1 measure, in the manner of the binary classification of terms as the misspelling and correct term classes. These evaluation metrics are calculated based on Equations 3–5 (Huyssteen et al., 2004); where True Negatives (TN) are correctly flagged misspellings, False Positives (FP) are unidentified misspellings, True Positives (TP) are correctly identified well-spelled words, and False Negatives (FN) are wrongly flagged well-spelled words. The desirable property for any spelling error detector would be to score 100% precision as it should flag all misspellings, and only misspellings; and also to score 100% recall as it should recognize all valid words as correct, and all invalid words as misspellings. Hence, recall is mostly an indication of the language coverage. F1 Measure gives an overall view of the capability of a spelling error detector. To compute the actual scores, we used the manually compiled test data as the *gold standard* for the evaluation.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F1\ Measure = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (5)$$

The quality of suggestions offered by a spelling corrector is measured by the relative positions of the correct spellings in the suggestions list (Mitton, 2008). In the best scenario, the right correction always appears on the topmost of the list.

5. Results and Discussions

This section gives a detailed description of the results of our evaluation. The results are presented broadly in terms of spelling error detection and spelling error correction for Amharic and English.

5.1 Amharic Results

Figure 1 indicates the precision and recall graph for Amharic spelling error detection. The precision and recall scores are computed based on the different word lists compiled from the most frequent words in CACO and HaBiT corpora. The optimum results were obtained when a term list is composed of seven or more frequent words from the HaBiT and eight or more frequent words from the CACO corpus were used. The corresponding precision, recall, and F1 measures are given in Table 4. It also shows the scores for the baseline systems.

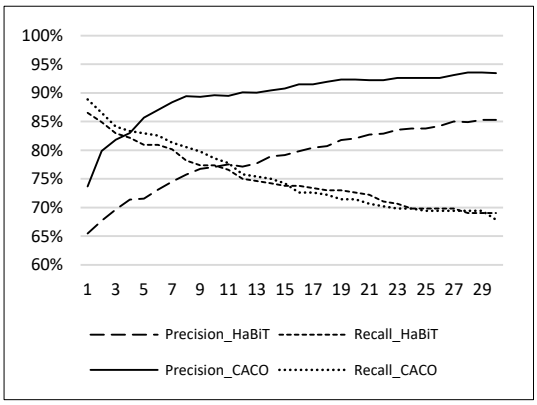

Figure 1: Precision and recall graph for spelling error detection in Amharic.

The evaluation results indicate that the F1 measure improves for the proposed system using the CACO corpus from 78% to 85% for spelling error detection compared to the baseline systems (see Table 4). However, we did not get any improvement when we used the term list compiled from the HaBiT corpus.

Metric	Proposed using CACO	Proposed using HaBiT	Aspell	Hunspell
Precision	89.4%	74.5%	79.4%	79.1%
Recall	80.6%	80.2%	76.6%	76.6%
F1	84.8%	77.2%	78.0%	77.8%

Table 4: Amharic spelling error detection results.

The measures of qualities of suggestions offered by the baseline and proposed systems for Amharic spelling error are shown in Table 5. According to the results, 77% of correct spellings appear in the top five suggestions list for the proposed system using CACO compared to 34% for Hunspell and 62% for Aspell. When we used the HaBiT corpus, 75% of correct spellings appear in the top five suggestions list, which is lower than that of the CACO corpus by 2%. Furthermore, when we consider the correct spellings that appear in the top first suggestions list, the proposed approach that uses the CACO corpus scored 9% higher than that uses the HaBiT corpus. This performance gain indicates that our approach is dependent on the underlying corpus used.

	Proposed using CACO	Proposed using HaBiT	Aspell	Hunspell
Top first	52.0%	42.9%	33.7%	16.7%
Top two	67.5%	61.9%	45.2%	26.6%
Top three	73.8%	69.4%	53.2%	29.0%
Top four	75.8%	73.8%	59.9%	33.7%
Top five	77.0%	75.4%	61.9%	34.1%

Table 5: Percentage of the topmost correct suggestions provided for Amharic spelling error correction.

5.2 English Results

The optimum F1 measure for English spelling error detection is obtained when we have used a term list that is compiled from fifty-seven or more frequent words from the BNC corpus. Its corresponding precision, recall, and F1

measures are given in Table 6 along with those of the baseline systems. The F1 measure for the proposed system is 96% and 97% for both baseline systems. The proposed system is lower than the baseline systems by 1%.

Metric	Proposed using BNC	Aspell	Hunspell
Precision	95.4%	99.2%	97.8%
Recall	96.1%	95.4%	95.3%
F1	95.7%	97.3%	96.6%

Table 6: English spelling error detection results.

The measures of qualities of suggestions offered by the baseline and proposed systems for English spelling error are shown in Table 7. With the proposed system, 74% of correct spellings appear in the top five suggestions list compared to 56% for Hunspell and 61% for Aspell.

	Proposed using BNC	Aspell	Hunspell
Top first	56.6%	27.4%	26.7%
Top two	66.0%	36.0%	38.8%
Top three	70.3%	50.1%	46.8%
Top four	72.3%	55.7%	52.7%
Top five	73.5%	60.5%	56.4%

Table 7: Percentage of the topmost correct suggestions offered for English spelling error correction.

6. Conclusion

We have proposed a method of an Amharic spelling corrector. Special characteristics of our approach are that it infers linguistic knowledge from text corpus and can be ported to other written languages with little effort as long as they are typed using a QWERTY keyboard with direct mappings between keystrokes and characters. The effort it requires is tokenization and transliteration to Latin characters. The proposed method was evaluated with the baseline systems. The evaluation results for Amharic and English test data confirm that our approach has a better performance than the baseline systems. This is mainly because of the application of a smoothed language model, generalized error model and the ability to take into account the context of misspellings. Since our approach is in a way to infer linguistics knowledge from a text corpus, the quality of the corpus that we have used has a direct effect on its performance. This is clearly shown with the performance differences between the two different Amharic corpora used.

A corpus-driven approach is related to lexicons used for spelling error detection. We can hardly find a manually compiled lexicon with reasonable coverage for a less-resourced language which has rich morphology like Amharic. Instead of using a handcrafted lexicon, using a term list derived from frequently occurring terms from a text corpus has advantages. Such a term list, in addition to ease of compilation, has also benefits in handling rare terms, proper nouns, technical terms, brand names, and newly coined terms (neologisms).

For future work, we will try to evaluate our approach for real-word spelling errors.

7. Acknowledgements

We would like to thank the anonymous reviewers for their invaluable comments. Our thanks also goes to Daniel Yacob at Ge'ez Frontier Foundation for making available some of the Amharic e-books that we have used for the CACO corpus.

8. Bibliographical References

Ahmed, F., Luca, E. W., and Nürnberger, A. (2009). Revised n-gram based automatic spelling correction tool to improve retrieval effectiveness. *Research Journal on Computer Science and Computer Engineering with Applications* (Polibits), 40: 39–48.

Aklilu, A. (2004). Sabean and Geez symbols as a guideline for Amharic spelling reform. In Proceedings of the First International Symposium on Ethiopian Philology, pages 18–26, Addis Ababa, Ethiopia, October. Addis Ababa University Press.

Blair, C. R. (1960). A program for correcting spelling errors. *Information and Control*, 3: 60–70.

Bloor, T. (1995). The Ethiopic writing system: A profile. *Journal of the Simplified Spelling Society*, 19(2): 30–36.

Brill, E., and Moore, R. C. (2000). An improved error model for noisy channel spelling correction. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pages 286–293, October. Hong Kong, ACL.

Chen, S. F., and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. Tech. rep. TR-10-98, Computer Science Group, Harvard University.

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3): 171–176.

ELA. (1970). የአማርኛ ፡ ፊደል ፡ ሕግን ፡ አንዲጠብቅ ፡ ለማድረግ ፡ የተዘጋጀ ፡ ራዎር ፡ ማስታወሻ, *Journal of Ethiopian Studies*, 8(1): 119–134.

Fabri, R., Gasser, M., Habash, N., Kiraz, G., and Wintner, S. (2014). Linguistic Introduction: The orthography, morphology, and syntax of Semitic languages. In I. Zitouni (Ed.), *Natural Language Processing of Semitic Languages*, Berlin, Germany: Springer-Verlag, pp 3–41.

Gao, J., Li, X., Micol, D., Quirk, C., and Sun, X. (2010). A large scale ranker-based system for search query spelling correction. Proceedings of the 23rd International Conference on Computational Linguistics, pages 358–366, Beijing, August. ACL.

Gezmu, A. M., Seyoum, B. E., Lema, T. T., and Nürnberger, A. (2017). Manually annotated spelling error corpus for Amharic. Tech. rep. FIN-001-2017, Data and Knowledge Engineering Group, Otto-von-Guericke-Universität Magdeburg.

Hayward, K., & Hayward, R. J. (1992, June). Amharic. *Journal of the International Phonetic Association*, 22(1-2), 48-52.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 690–696, Sofia, Bulgaria, August. ACL.

Holbrook, D. (1964). English for the rejected: Training literacy in the lower streams of the secondary school. Cambridge, Cambridge University Press. Retrieved

January 8, 2016, from http://www.dcs.bbk.ac.uk/~roger/holbrook-tagged.dat

Huyssteen, V. G., Eiselen, E., and Puttkammer, M. (2004). Re-evaluating evaluation metrics for spelling checker evaluations. In Proceedings of First Workshop on International Proofing Tools and Language Technologies, pages 91–99, Patras, Greece. University of Patras.

IPA. (2015, June 1). *Full IPA Chart*. Retrieved August 12, 2017, from International Phonetic Association: https://www.internationalphoneticassociation.org/sites/default/files/IPA_Kiel_2015.pdf

Kernighan, M. D., Church, K. W., and Gale, W. A. (1990). A spelling correction program based on a noisy channel model. In Proceedings of the 13th conference on Computational linguistics - Volume 2, pages 205–210, Helsinki, Finland, August. ACL.

Kneser, R., and Ney, H. (1995). Improved backing-off for m-gram language modeling. In International Conference on Acoustics, Speech, and Signal Processing, pages 181–184, Detroit, Michigan, May. IEEE.

Kukich, K. (1992). Techniques for Automatically Correcting Words in Text. *ACM Computing Surveys*, 24(4): 377-439.

Mekonnen, A. (2012). Development of an Amharic spelling corrector for tolerant-retrieval. In Proceedings of the International Conference on Management of Emergent Digital EcoSystems, pages 22–26, Addis Ababa, Ethiopia, October. ACM.

Mitton, R. (1985). A collection of computer-readable corpora of English spelling errors, *Cognitive Neuropsychology*, 2(3): 275–279.

Mitton, R. (2008). Ordering the suggestions of a spellchecker without using context. *Natural Language Engineering*, 15(2): 173–192.

Mitton, R. (2010). Fifty years of spellchecking. *Writing Systems Research*, 2(1): 1–7.

Morris, R., and Cherry, L. L. (1975). Computer detection of typographical errors. *IEEE Transactions on Professional Communication*, PC-18(1): 54–64.

Norvig, P. (2009). Natural Language Corpus Data. In T. Segaran, and J. Hammerbacher (Eds.), *Beautiful Data: The Stories Behind Elegant Data Solutions*. Sebastopol, Canada: O'Reilly Media, Inc., pp 219–242.

Rychlý, P., and Suchomel, V. (2016). Annotated Amharic corpora. In International Conference on Text, Speech, and Dialogue, pages 295–302, Brno, Czech Republic, September. Springer International Publishing.

Scannell, K. P. (2007). The Crúbadán project: Corpus building for under-resourced languages. In C. Fairon, H. Naets, and A. Kilgarriff (Eds.), *Building and Exploring Web Corpora*. Louvain-la-Neuve, Belgium: UCL Presses, pp 5–15.

Tachbelie, M. Y., and Menzel, W. (2007). Sub-word based language modeling for Amharic. In N. Nicolov, G. Angelova and R. Mitkov (Eds.) *Recent Advances in Natural Language Processing V: Selected papers from RANLP 2007*. Amsterdam: John Benjamins Publishing Company, pp 301–310.

Toutanova, K., and Moore, R. C. (2002). Pronunciation modeling for improved spelling correction. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 144–151, Philadelphia, July. ACL.

Unicode Consortium. (2017). The Unicode® Standard: Version 10.0 – Core Specification. Mountain View, CA, USA: Unicode, Inc.

Whitelaw, C., Hutchinson, B., Chung, G. Y., and Ellis, G. (2009). Using the web for language independent spellchecking and autocorrection. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing - Volume 2, pages 890–899, Singapore, August. ACL and AFNLP.

Winkler, W. E. (2006). Overview of record linkage and current research directions. Tech. rep., Statistical Research Division, U.S. Census Bureau.

Yacob, D. (2003). Application of the Double Metaphone Algorithm to Amharic Orthography. In The XVth International Conference of Ethiopian Studies, pages 921–934, Hamburg, July. Otto Harrassowitz Verlag.

Yitna, F., and Yaqob, D. (1997). The System for Ethiopic Representation in ASCII. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.3191

9. Language Resource References

BNC XML Edition. (2007). The British National Corpus, version 3, distributed via Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium. URL: http://www.natcorp.ox.ac.uk/

ELRA-W0074. (2014). Amharic-English bilingual corpus, distributed via ELRA, 1.0, ISLRN 590-255-335-719-0.

HaBiT. (2016). Harvesting big text data for under-resourced languages, distributed via Natural Language Processing Centre, Faculty of Informatics, Masaryk University. URL: http://habit-project.eu