Inf Retrieval (2009) 12:416–435
DOI 10.1007/s10791-008-9080-x

Classifying Amharic webnews

Lars Asker · Atelach Alemu Argaw · Björn Gambäck · Samuel Eyassu Asfeha · Lemma Nigussie Habte

Received: 11 May 2008 / Accepted: 24 November 2008 / Published online: 20 February 2009

Abstract We present work aimed at compiling an Amharic corpus from the Web and automatically categorizing the texts. Amharic is the second most spoken Semitic language in the World (after Arabic) and used for countrywide communication in Ethiopia. It is highly inflectional and quite dialectally diversified. We discuss the issues of compiling and annotating a corpus of Amharic news articles from the Web. This corpus was then used in three sets of text classification experiments. Working with a less-researched language highlights a number of practical issues that might otherwise receive less attention or go unnoticed. The purpose of the experiments has not primarily been to develop a cutting-edge text classification system for Amharic, but rather to put the spotlight on some of these issues. The first two sets of experiments investigated the use of Self-Organizing Maps (SOMs) for document classification. Testing on small datasets, we first looked at classifying unseen data into 10 predefined categories of news items, and then at clustering it around query content, when taking 16 queries as class labels. The second set of experiments investigated the effect of operations such as stemming and part-of-speech tagging on

L. Asker · A. A. Argaw
Department of Computer and Systems Sciences, Stockholm University, Stockholm, Sweden
e-mail: asker@dsv.su.se

A. A. Argaw
e-mail: atelach@dsv.su.se

B. Gambäck (✉)
Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: gamback@idi.ntnu.no; gamback@sics.se

B. Gambäck
SICS, Swedish Institute of Computer Science AB, Kista, Sweden

S. Eyassu Asfeha · L. Nigussie Habte
Department of Information Science, Addis Ababa University, Addis Ababa, Ethiopia
e-mail: samueleya@yahoo.com

L. Nigussie Habte
e-mail: lemma_nigussie@yahoo.com

text classification performance. We compared three representations while constructing classification models based on bagging of decision trees for the 10 predefined news categories. The best accuracy was achieved using the full text as representation. A representation using only the nouns performed almost equally well, confirming the assumption that most of the information required for distinguishing between various categories actually is contained in the nouns, while stemming did not have much effect on the performance of the classifier.

Keywords Web mining · Text classification · Semitic languages

1 Introduction

The emergence of the World Wide Web has provided new and important opportunities to easily access and combine data and information from several different sources and thereby enabling the construction of new resources for researchers and people everywhere. Although most webpages on the Internet are in English, there is a growing number of non-English web pages, and today there are more than twice as many non-English as English speaking Internet users (Miniwatts Marketing Group 2008). This entails that, for people in many parts of the World, there is an urgent need to develop tools and resources that can allow them to better access, process and disseminate information on the Internet while using their own language. Although there is an increasing trend to investigate and apply language processing methods to languages other than English, most of the work is done on a limited number of mainly European and East-Asian languages. For the vast number of the World's languages that lack computational linguistic resources there still remains plenty of work to be done. The main obstacles to progress in language processing for these languages are 2-fold. Firstly, the peculiarities of the languages themselves might force new strategies to be developed. Secondly, the lack of already available resources and tools makes the creation and testing of new ones more difficult and time-consuming.

Amharic is a Semitic language used for countrywide communication in Ethiopia. It is highly inflectional and quite dialectally diversified. With more than 20 million speakers, it is the second most spoken Semitic language in the World (after Arabic) and today probably one of the five largest on the African continent (albeit difficult to determine, given the dramatic population size changes in many African countries in recent years). In spite of the relatively large number of speakers, it is still a language for which very few computational linguistic resources have been developed, and little has been done in terms of making useful higher-level Internet or computer-based applications available to those who only speak Amharic.

The Ethiopian culture is ancient, and so are the written languages of the area, with Amharic using its own script. Several computer fonts for the script have been developed, but for many years it had no standardized computer representation, which was a deterrent to electronic publication.[1] More and more digital information is now being produced in Ethiopia, but no deep-rooted culture of information exchange and dissemination has been established. Different factors are attributed to this, including lack of digital library facilities and central resource sites, inadequate resources for electronic publication of journals and

[1] An international standard for Amharic was agreed on as late as in 1998, following Amendment 10 to ISO-10646-1. It was incorporated into Unicode in 2000: www.unicode.org/charts/PDF/U1200.pdf.

books, and poor documentation and archive collections. The difficulties to access information have led to low expectations and under-utilization of existing information resources.

The need for accurate and fast information access (acknowledged as a major factor affecting the success and quality of research and development, trade and industry; cf. Furzey 1996) has lead to an increasing awareness of the need to develop Amharic language processing resources and digital information access and storage facilities. To this end, some work on Amharic has now been carried out, with efforts in areas such as:

- word formation (Fissaha and Haller 2003a, b; Amsalu and Gibbon 2005),
- stemming (Alemayehu and Willett 2002),
- treebank building (Argaw et al. 2003),
- the collection of an (untagged) corpus (Yacob 2005), and
- character recognition (Cowell and Hussain 2003).

The need for investigating Amharic *information access* has been acknowledged by the European Cross-Language Evaluation Forum (CLEF), which added an Amharic–English track in 2004. However, the task addressed cross-lingual information access from an English database, with the original questions being posed in Amharic (and then translated into English). Three groups participated in this track, with Argaw et al. (2005) reporting the best results. Amharic has continued to appear as a query language at the CLEF tracks for the past few years, although only one group has participated. Thus bilingual Amharic to English (Argaw and Asker 2007a; Argaw 2008) and Amharic to French (Argaw et al. 2006) retrieval tasks have been reported.

Pioneering the work on *morphological analysis* of Amharic verbs was a knowledge-based system for parsing verbs and nouns derived from verbs which used root pattern and affixes to determine the lexical and inflectional category of the words (Bayou 2000). In a later experiment, an unsupervised learning approach based on probabilistic models was utilized to extract morphemic components (prefix, stem and suffix) to construct a morphological dictionary (Bayu 2002). The system was able to successfully parse 87% of a small testdata set of 500 words. The first larger-scale morphological analyser for Amharic verbs used XFST, the Xerox Finite State Tools (Fissaha and Haller 2003a). This was later extended to include all word categories (Amsalu and Gibbon 2005). Testing with 1620 words text from an Amharic bible, 88–94% recall and 54–94% precision were reported.

In the present paper, we briefly describe an attempt to mine Amharic text from the Web and then discuss several classification experiments that were performed on the compiled corpus. We report three groups of experiments, in the first two we used the Self-Organizing Map (SOM) model of artificial neural networks (ANNs) for the task of classifying a collection of Amharic news items. The documents were classified both by using a set of predefined classes and by taking a set queries as class labels.

In the third group of experiments we compared three representations (full, nouns only, and stemmed) using 10-fold cross validation while constructing classification models based on bagging of decision trees. It is generally believed that applications such as information retrieval, text classification, or document filtering could benefit from the existence and availability of basic tools such as stemmers, morphological analysers or part-of-speech taggers (see Sect. 3.4). However, since so few language processing resources for Amharic are available, very little is known about their effect on retrieval or classification performance for this language, and the issue has therefore been explored in this work.

The rest of the paper is laid out as follows. In Sect. 2 we describe the Amharic language and its writing system in more detail, while Sect. 3 introduces the methods and techniques that we will use in the paper and discusses some previous work on applying machine learning and stemming to information access. Section 4 describes how we mined the web for the news items corpora used in classification experiments and how the corpora were preprocessed and analysed.

Section 5 detail the actual experiments, with the ones using SOMs in Sect. 5.1 (predefined classes) and Sect. 5.2 (query-based classes), and the ones combining decision trees with stemming in Sect. 5.3, while Sect. 6 discusses the overall results and put them in perspective to related work. Finally, Sect. 7 contains concluding remarks and sums up the main contents of the paper.

2 The Amharic language and script

Ethiopia is the third most populous African country and harbours more than 80 different languages.[2] Three of these are dominant: Oromo, a Cushitic[3] language spoken in the South and Central parts of the country and written using the Latin alphabet; Tigrinya, spoken in the North and in neighbouring Eritrea; and Amharic, spoken in most parts of the country, but predominantly in the Eastern, Western, and Central regions. Amharic and Tigrinya are Semitic and about as close to each other as are Spanish and Portuguese (Bloor 1995). Both languages are written using their own unique script, horizontally and left-to-right (in contrast to many other Semitic languages).

The actual size of the Amharic speaking population has to be based on estimates: Hudson (1999) analysed the national Ethiopian census from 1994 and indicated that more than 40% of the (then) 53 million Ethiopians understood Amharic, with at the time about 17 million first language speakers. The current size of the Ethiopian population is estimated[4] to be some 82 million people (CIA 2008), and Amharic to be spoken by well over 20 million people as first language, making it the second most spoken Semitic language in the World (after Arabic). It is today probably the second largest language in Ethiopia (after Oromo).[5] Following the Constitution drafted in 1993, Ethiopia is divided into nine fairly independent regions, each with its own nationality language. However, Amharic is the language for countrywide communication and was also for a long period the principal literal language and medium of instruction in primary and secondary schools of the country, while higher education is carried out in English.

Amharic and Tigrinya speakers are mainly Orthodox Christians, with the languages drawing common roots to the ecclesiastic Ge’ez still used by the Coptic Church. Written

[2] The number of languages in a country is as much a political as a linguistic issue. The number of languages of Ethiopia thus differs from 70 up to 420, depending on the source; however, the 1994 Ethiopian census listed 77 distinct, living languages plus a category for “other languages” (Hudson 1999), while the Ethnologue (Gordon Jr 2005) claims 82 (plus 4 extinct) and Hudson (2006) only gives 75 (including 4 extinct).

[3] Together with the Semitic languages, the Cushitic languages make up two of the branches of the Afro-Asiatic language family; the other branches are Berber, Chadic, Egyptian, and Omotic (Gordon Jr 2005).

[4] There should be a census every 10 years, according to the Ethiopian constitution. However, the census of 2004 was delayed due to political unrest, and initiated only in 2007. No results have been published so far.

[5] The number of speakers of a language is also an issue influenced by political and economical interests. Thus the makers of the ‘Wazéma2001’ software for Ethiopic character encoding (www.gzamargna.net) state that there are some 90 million speakers of Amharic (Negga 2008). However, it is a generally accepted fact that Amharic is the second largest Semitic language, since the size-wise differences are in the order of a magnitude: the first-language speakers of Arabic count to well over 200 million, while the ones for Hebrew and Tigrinya are in the order of 5 million, and Gurage (a group of Ethiopian languages) about 2 million—with other Semitic languages counting their speakers in thousands (see, e.g., Gordon Jr 2005).

Table 1 The orders for ስ (/s/) and ም (/m/)

Order	1	2	3	4	5	6	7
C \ V	/ə/	/u/	/i/	/ɐ/	/e/	/ɨ/	/o/
/s/ /m/	ሰ መ	ሱ ሙ	ሲ ሚ	ሳ ማ	ሴ ሜ	ስ ም	ሶ ሞ

Ge'ez can be traced back to at least the 4th century A.D. The first versions of the script included consonants only, while the characters in later versions represent consonant-vowel (CV) phoneme pairs.

In modern written Amharic, each syllable pattern comes in seven different forms (called *orders*), reflecting the seven vowel sounds. The first order is the basic form; the other orders are derived from it by more or less regular modifications indicating the different vowels. There are 33 basic forms, giving 7*33 syllable patterns, or *fidels*.[6]

Two of the base forms represent vowels in isolation (ዐ and አ), but the rest are for consonants (or semivowels classed as consonants) and thus correspond to CV pairs, with the first order being the base symbol with no explicit vowel indicator (though a vowel is pronounced: C+/ə/). The sixth order is ambiguous between being just the consonant or C+/ɨ/. The writing system also includes 20 symbols for labialised velars (four five-character orders) and 24 for other labialisation. In total, there are 275 *fidels*. The sequences in Table 1 (for ስ /s/ and ም /m/) exemplify the (partial) symmetry of vowel indicators. Amharic also has its own numbers (though not widely used nowadays) and its own punctuation system with eight symbols. For more thorough discussions of the Ethiopian writing system, see, for example, Bender et al. (1976) and Bloor (1995). Like many other Semitic languages, Amharic is an SOV[7] language with a rich verb morphology based on triconsonantal roots with vowel variants describing modifications to, or supplementary detail and variants of the root form. A verb can have well over 150 different forms.

There is no agreed upon spelling standard for compounds and the writing system uses multitudes of ways to denote compound words. In addition, not all the letters of the Amharic script are strictly necessary for the pronunciation patterns of the language; some were simply inherited from Ge'ez without having any semantic or phonetic distinction in modern Amharic (though they do in Ge'ez and Tigrinya): there are many cases where numerous symbols are used to denote a single phoneme, as well as words that have extremely different orthographic form and slightly distinct phonetics, but the same meaning. For example, most labialised consonants are basically redundant, and there are actually only 39 context-independent phonemes (monophones): of the 275 symbols of the script, only about 233 remain if the redundant ones are removed.

3 Methods and techniques

As a result of the character redundancy, and of the size of Ethiopia leading to vast dialectal dispersion, lexical variation and homophony is very common in Amharic, and obviously

[6] '*fidel*' (lit. 'alphabet' in Amharic) refers both to the characters as such and the entire script. The script is also known as 'Ethiopic'. This is a bit misleading since it (or variants of it) is (or has been) used by several languages in the Horn of Africa region, including Amharic, Tigrinya, Gurage (Semitic); Sidamo and Blin (Cushitic); and Wolaytta (Omotic)—even though Eritrea, following its independence in 1993, has adopted a policy that all non-Semitic languages should use Roman-based alphabets.

[7] SOV (Subject–Object–Verb) refers to the basic word-order of the language. In contrast, most Western-European languages have an SVO word-order.

deteriorates the effectiveness of Information Access systems based on strict term matching; hence for the experiments in this paper we use the approximative matching enabled by applying machine learning strategies. These strategies are introduced in this section.

3.1 Artificial Neural Networks

Artificial Neural Networks is a computational paradigm inspired by the neurological structure of the human brain, and ANN terminology borrows from neurology: the brain consists of millions of neurons connected to each other through long and thin strands called axons; the connecting points between neurons are called synapses.

ANNs have proved themselves useful in deriving meaning from complicated or imprecise data; they can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computational and statistical techniques. Traditionally, the most common ANN setup has been the backpropagation architecture (Rumelhart et al. 1986), a supervised learning strategy where input data is fed forward in the network to the output nodes (normally with an intermediate hidden layer of nodes, adding nonlinear capabilities) while errors in matches are propagated backwards in the net during training.

Neural networks have been widely used in text classification, where they can be given terms and having the output nodes represent categories. Ruiz and Srinivasan (1999) utilize a hierarchical array of backpropagation neural networks for (nonlinear) classification of MEDLINE records, while Ng et al. (1997) use the simplest (and linear) type of ANN classifier, the perceptron. Nonlinear ANN methods have so far not been shown to add any performance to linear ones for text categorization (Sebastiani 2002; Schütze et al. 1995).

3.2 Self-Organizing Maps

Self-Organizing Maps is a type of ANN utilizing an unsupervised learning scheme. SOMs were invented by Kohonen (1999, 2001) and originally developed to project multidimensional vectors on a reduced dimensional space. Self-organizing systems can have many kinds of structures, a common one consists of an input layer and an output layer, with feed-forward connections from input to output layers and full connectivity (connections between all neurons) in the output layer.

A SOM is provided with a set of rules of a local nature (a signal affects neurons in the immediate vicinity of the current neuron), enabling it to learn to compute an input-output pairing with specific desirable properties. The learning process consists of repeatedly modifying the synaptic weights of the connections in the system in response to input (activation) patterns and in accordance to prescribed rules, until a final configuration develops. Commonly both the weights of the neuron closest matching the inputs and the weights of its neighbourhood nodes are increased. At the beginning of the training the neighbourhood can be fairly large and then be allowed to decrease over time.

SOMs have been used for information access since the beginning of the 90s (Lin et al. 1991). A SOM may show how documents with similar features cluster together by projecting the *N*-dimensional vector space onto a two-dimensional grid. The radius of neighbouring nodes may be varied to include documents that are weakly related. The most elaborate experiments of using SOMs for document classification have been undertaken using the WEBSOM architecture developed at Helsinki University of Technology (Kaski et al. 1996; Honkela et al. 1997; Kohonen et al. 2000). WEBSOM is based on a hierarchical two-level SOM structure, with the first level forming histogram clusters of words.

The second level is used to reduce the sensitivity of the histogram to small variations in document content and performs further clustering to display the document pattern space.

The SOM model preparation passes through the processes undertaken by the Latent Semantic Indexing (LSI) model (Deerwester et al. 1990; Dumais 1995), and the classical vector space model (Salton and McGill 1983). Hence those models can be taken as particular cases of the SOM, when the neighbourhood diameter is maximized. For instance, one can calculate the LSI model's similarity measure of documents versus queries by varying the SOM's neighbourhood diameter, if the training set is a singular value decomposition reduced vector space. A SOM is capable of simulating new data sets without the need of retraining itself when the database is updated; something which is not true for LSI. Moreover, LSI consumes ample time in calculating similarities of new queries against all documents, but a SOM only needs to calculate similarities versus some representative subset of old input data and can then map new input straight onto the most similar models without having to recompute the whole mapping. Tambouratzis et al. (2003) use SOMs for categorizing texts according to register and author style, and show that the results are compatible to those generated by statistical methods.

3.3 Decision trees and bagging

A decision tree is a predictive model that can be automatically constructed from labelled training examples by initially partitioning the examples into two or more groups according to the values of the most discriminative attribute (cf. Quinlan 1986). The decision tree construction is then continued by recursively partitioning each sub-group in the same way until all examples in each sub-group have the same label and no further partitioning is possible. The resulting decision tree can be used to predict the category of new and unseen examples.

Bagging (bootstrap aggregation; Breiman 1996) is a machine learning technique used to improve the performance of a combined classifier by training and combining the predictions of several (usually 20–50) different component classifiers. The component classifiers vary by being trained on different variants of the original training data set. The variants are constructed by sampling (with replacement) N examples from the original, where N is equal to the total number of examples in the original training set.

3.4 Stemming

Stemming is a technique whereby morphological variants are reduced to a single stem. It is language-dependent and should be tailored for each language, since languages have a varying degree of differences in their morphological properties. It should also be tailored to the specific task at hand.[8] Stemming is commonly applied in text classification tasks as a preprocessing method to reduce morphological variants to a single feature.

The effect of stemming in automatic classification tasks is not consistent and depends on the specific language and the domain of the document collection. For example, Gaustad and Bouma (2002) report results from experiments on Dutch email and news text classification using simple suffix stripping and a dictionary-based stemming. Neither method

[8] Stemming is then normally used as "a poor man's version" of full-scale morphological analysis, mainly aiming at stripping off prefixes and suffixes, while leaving the root forms unchanged. For morphology-poor languages such as English, this basically amounts to the same thing as morphological analysis, while for most other natural languages procedures altering the roots (and infixes), splitting compounds, handling derivate processes, etc., are needed in order to perform a complete morphological analysis.

improved classification accuracy in their experiments. Classification and retrieval tasks for English commonly apply stemming in the preprocessing stage, although the effects on the performance are not conclusive. Karlgren and Sahlgren (2001) report a clear improvement when comparing stemmed and non-stemmed English words on a synonym-finding task, while adding part-of-speech tags actually deteriorated performance, possibly due to the increase in vocabulary size that follows from separating out different part-of-speech versions of the same word from each other.

On the other hand, similar research on highly inflected languages such as Arabic report an increase in performance due to stemming both in information retrieval (Xu et al. 2002; Larkey 2002) and text classification (Syiam et al. 2006). Amharic is a language with very rich morphology; hence we assumed that stemming would have a positive effect for classification and related tasks. The main previous contribution in the area is the work by Alemayehu and Willett (2002, 2003) which investigated the effect of stemming for information retrieval on a limited Amharic document collection (consisting of 548 documents and 40 queries). Those studies also indicated a positive effect from using the stemmed forms: Alemayehu and Willett (2003) compared performance of word-based, stem-based, and root-based retrieval, and showed better recall levels for stem- and root-based retrieval over word-based. However, no information on the precision of the experiments was provided.

We have developed a stemmer for Amharic (Argaw and Asker 2007b). The stemmer finds all possible segmentations of a given word according to the morphological rules of the language and then selects the most likely prefix and suffix for the word based on corpus statistics. It strips off the prefix and suffix and then tries to look up the remaining stem (or alternatively, some morphologically motivated variants of it) in a dictionary to verify that it is a possible stem of the word. The frequency and distribution of prefixes and suffixes over Amharic words is based on a statistical analysis of a 3.5 million word Amharic news corpus. The stemmer had an accuracy of $\sim$85% when evaluated on a limited text consisting of 50 sentences (805 words) from CLEF 2006 (Argaw and Asker 2007a).

Table 2 shows some examples of words and their stemmed form (all examples are transliterated into SERA, see Sect. 4.2). In this context it is interesting to note that incorrectly stemmed words will have no negative effect on classification performance so long as other words that belong to a different category are not reduced to the exact same form. See, e.g., the incorrect form *ezo* in Table 2: it works just as well as the correct form *zon* as a stemmed form for the words *yezon*, *yezonu*, and *bezonu*.

4 Web mining for an Amharic corpus

In this article we describe a number of text classification and clustering experiments that have been performed on Amharic news text. All these news texts originate from the Walta Information Center, which is a private news and information service located in Addis Ababa, Ethiopia. At its website www.waltainfo.com, it provides Ethiopia-related news in English and Amharic on a daily basis.

4.1 Data collection

We have collected a total of 8715 Amharic news articles from Walta Information Center using a web crawler.[9] The 8715 news articles originate from the years 2001 to 2004. The

[9] Offline Explorer Pro 3.5 from MetaProducts Corporation: www.metaproducts.com.

Table 2 Stemmed form and original variants of some Amharic words

Stemmed	Non-stemmed
br	br bebr
bexta	bexta bextaw bextawoc bebextaw kebextaw
mkr	mkr yemkr bemkr
xeT	yetexeTebet texeTe mexeTun
mekelakeya	mekelakeyana mekelakeya yemekelakeya
mrCa	mrCa yemrCa
mnzari	mnzari yemnzari
guba	gubaE gubaEw
bEt	bEt bEtu yebEt bEtoc bEtocn bEtocna bEte
projekt	projektoc projektocu projektocn projekt yeprojekt yeprojektu projektu
bank	bank bebankoc banku bankoc yebank
guday	guday gudayu gudayoc
zqteNa	zqteNaw zqteNa
amakay	beamakay amakaynet
ezo	yezon yezonu bezonu
mrmr	mrmr bemrmr yemrmr
hzb	hzbu hzboc hzb yehzb lehzb
lmat	lmat yelmat lelmat
Ec	Ec yeEc beEc

Walta website included a large number of links to other webpages that were irrelevant to our purposes. A web crawler allowed us to control and automate the exploration of the website in such a way that we could access and collect only the specific news articles that were useful for our purposes.

The news texts at the Walta website were structured in folders with subfolders for each year, month, and day, respectively. The Amharic news is archived under folders according to the Ethiopian calendar.[10] For example, `AmNews/1994/tir/24tir94/tir24a04.htm`. A particular day would typically contain between five and ten separate news items. Since the file names do not contain sufficient information to identify the date (information about the year is missing) we downloaded all available articles from the archive while retaining the original folder structure.

4.2 Data preprocessing

When the file structure for the relevant news articles had been downloaded, it was flattened and the HTML code for each page was removed using a publicly available freeware.[11] This software allows for controlled removal of HTML tags as well as whitespace and other special characters from HTML files. An extra degree of control was required since the representation for the Amharic fonts includes some special characters (e.g., '{', '}' and 'l') that would create problems for the transliteration if they had been removed. In addition to

[10] The Ethiopian calendar runs approximately 7 years and 8 months behind the Gregorian calendar, so the data came from the Ethiopian years 1993–1997.

[11] Emsa HTML Tag Remover v1.0 Build 20.

this, it was important to preserve portions of the original file structure in order to simplify the parsing of the processed files into separate fields such as title, place name, and date.

The news texts from Walta have a semi-structured format that includes title, place name, date, news agency, and body. In order to simplify the processing of the texts, we stored them in an XML structure that identifies each of these fields separately.

4.2.1 Transliteration

The *fidels* in the Amharic texts are represented using a variety of fonts. For the Ethiopian years 1993 until the first half of 1996, Visual Geez 2000 (or VG2 Main) was the most common, while after that, a mixture of fonts have been used, which complicated the transliteration step considerably. In order to further simplify the analysis and to have a unified representation of the Amharic texts, we transliterated all Amharic texts into SERA (Firdyiwek and Yacob 1993; Yacob 1997). SERA (System for Ethiopic Representation in ASCII) is a convention for the transcription of *fidel* (Ethiopic script) into the seven bit ASCII format.[12] We worked with the transliterated form in order to make it compatible with the machine learning tools used for the experiments and in order to simplify spelling normalization.

4.2.2 Normalization

Pertaining to the fact that there are redundant symbols in the Amharic alphabet, a single word can be written using different variations. For example, the words *‘sr‘at* (ሥርዓት), *srat* (ስርአት), *sr‘at* (ስርዓት), and *‘srat* (ሥርአት) represent the same word (‘order’) with different orthography: *s* (ስ) and *‘s* (ሥ) represent variations of the sound /s/, while *a* (አ) and *‘a* (ዓ) represent two of the variations of the vowel /ɐ/. There are, in general, two variations of representation for the same sound in *fidel* (except for the case of *h* described next). In SERA, such variants of a single sound are represented using one English letter, with and without the symbol ‘ preceding the letter. By removing the symbol ‘ from the entire transliterated text, we have normalized the representation of words in different forms to one common form. SERA is case sensitive, i.e., upper and lower cases of the English alphabet represent different symbols in the Amharic alphabet. There is one exception where *H*, *h*, and *‘h* represent one sound (/hə/), and by replacing *H* by *h* we have been able to get more uniformity throughout the text.

4.2.3 Stopword removal

Stopword removal is another technique used in the preprocessing stage. Here the purpose is to remove commonly occurring words such as ‘the’, ‘on’, ‘he’, etc., since they do not help in discriminating between documents. To this end, a negative dictionary of 745 Amharic words was created, containing both stopwords that are news specific and the Amharic text stopwords collected by Alemayehu and Willett (2002). The news specific common terms were manually identified by looking at their frequency, in order to extend the stopword lists created in previous research on Amharic news texts (Sintayehu 2001; GebreMeskel 2003).

[12] The transliteration was done using a file conversion utility called g2 available in the LibEth package (LibEth is a library for Ethiopic text processing written in ANSI C; www.libeth.sourceforge.net).

Table 3 The 10 categories and the number of articles in walta_1065 belonging to each category

Cat	Topic	#
0	Sport	9
1	Hot news	55
2	Editorials	0
3	Politics	140
4	Business & Economy	351
5	Social	356
6	Culture	11
7	Science & Technology	47
8	Health	93
9	Art	3
	Total	1065

4.3 The datasets

For the experiments described in this article, we have used three subsets of the above-mentioned Amharic news collection. The first two are comparatively small, consisting of 101 and 105 news texts, respectively. We will refer to these as walta_101 and walta_105. The walta_101 articles were collected by Amsalu (2001), while walta_105 comes from GebreMeskel (2003). They are interesting for comparison purposes, since they have been used in previous experiments by several other researchers.

The third corpus is larger and consists of all the 1,065 Amharic news texts (210,000 words) from year 1994 in the Ethiopian calendar (parts of the Gregorian years 2001–2002). We will refer to this one as walta_1065. The walta_1065 corpus has been morphologically analysed and manually part-of-speech tagged by staff at the Ethiopian Languages Research Center at Addis Ababa University (Demeke and Getachew 2006). The tagset consists of ten basic classes: Noun, Pronoun, Verb, Adjective, Preposition, Conjunction, Adverb, Numeral, Interjection, and Punctuation, plus one extra tag for problematic (unclassified) words. The 10 basic classes were then further divided into a total of thirty subclasses.[13]

In addition to this, each article has been manually classified as belonging to one of ten predefined categories. The ten categories are presented in Table 3. The walta_1065 corpus was classified by linguists at the Ethiopian Languages Research Center. The walta_101 and walta_105 corpora were also matched against 25 queries. The selection of documents relevant to a given query was made by two domain experts (two journalists), one from the Monitor newspaper and the other from the Walta Information Center. A linguist from Gonder College participated in making consensus of the selection of documents made by the two journalists. Only 16 of the 25 queries were judged to have a document relevant to them in the walta_101 corpus. These 16 queries were found to be different enough from each other, in the content they try to address, to help map from document collection to query contents. Thus the queries as such were taken as class labels, in effect mapping the documents to 16 distinct classes. The walta_105 corpus was manually classified into these 16 classes by the experts. For both corpora, some documents

[13] A tagged version of the walta_1065 corpus is available online at http://nlp.amharic.org.

were found to tentatively belong to more than one class; however, the experts were compelled to assign them to the class to which they were most similar content-wise.

The walta_101 and walta_105 corpora were preprocessed to normalize spelling and to filter out stopwords. The normalization preprocessing step (see Sect. 4.2.2) tried to solve the problem that the same sound may be represented with two or more distinct but redundant written forms. The different forms were reduced to common representations using an extended version of a tool originally developed by Sintayehu (2001). In a second preprocessing step, the stopwords were removed from the word collection before indexing. After the preprocessing, the number of remaining terms in the walta_101 and walta_105 corpora together was 10363.

5 Experiments

The corpora described in the previous section were used for three rounds of experiments. The first two sets of experiments used the smaller datasets (walta_101 and walta_105) and investigated document classification using SOMs, while the third round of experiments used the larger dataset (walta_1065) to investigate the effect that operations like stemming and part-of-speech tagging can have on text classification performance for such a highly inflectional language as Amharic; here a decision tree-based classification strategy was used.

The SOMs in the first two sets of experiments were implemented using the Matlab Neural Network Toolbox from MathWorks Inc.,[14] while the third experiment set used the Rule Discovery System (RDS) from Compumine AB.[15] RDS is a rule-based machine learning platform that supports a variety of rule-based induction techniques such as rule sets and decision trees, and ensemble methods like bagging, boosting, and random forests.

5.1 SOM-based classification with predefined categories

In the first group of experiments, we looked at using Self-Organizing Maps for classification into a set of predefined categories. The news items (from walta_101 and walta_105) were classified into the ten different categories shown in Table 3. In order to normalize category size, fifteen news items selected from each category were indexed. The training set consisted of the first ten items in each category, and the test set of the remaining five. Normalization of category size has both advantages and disadvantages: while equally-sized categories is clearly not the common case in real-world problems, it does assure that there is at least a certain number of items belonging to each category even in experiments with small datasets. Normalization also has the property of minimizing the baseline. Since the largest category is equal in size to the smallest one, simply guessing that an item belongs to the most common category has minimal effect on performance (thus the baseline of this experiment is 10%).

A weighted matrix was generated from the original matrix using the log-entropy weighting formula (Dumais 1991). This helps to enhance the occurrence of a term in representing a particular document and to degrade the occurrence of the term in the document collection. The weighted matrix was then dimensionally reduced by Singular

[14] www.mathworks.com.

[15] www.compumine.com.

Value Decomposition, SVD (Berry et al. 1995). SVD makes it possible to map individual terms to the concept space.

The weighted term-by-document matrix, the ratio of the number of terms in a news item to the total number of terms weighted, was given to a SOM. The SOM was trained for 4,000 epochs. The predictive trend in classifying the vector of unique terms (in the news items) to a category achieved its peak result after 2500 epochs: 76.5% on the training set and 72.9% on the test set.

5.2 SOM-based document clustering as text classification

In the second round of experiments, we treated document clustering as text classification using SOMs. The walta_101 and walta_105 document collections were used for network training and testing, respectively.

The SVD-reduced vector space of pseudo-documents was assigned a class label (query content) to which the documents of the training set were identified to be more similar (by experts) and the neural net was trained using the pseudo-documents and their target classes. This was performed for 100–20,000 epochs and the neural net with best accuracy was considered for testing. A matrix of simple queries merged with the 101 documents (that had been used for training) was taken as input to a SOM-model neural net and the 101-dimensional document and query pairs were mapped and plotted onto a two-dimensional space. Those documents on the node on which the single query lies and those documents in the immediate vicinity of it (the neighbourhood was defined to be six nodes) were taken as being relevant to the query, that is, to belong to the class identified by the label given by the query. The classification accuracy over the sixteen classes on the training set (walta_101) was found to be 72.8%, while the performance on the test set (walta_105) was 69.5%.

In order to locate the error sources in the experiment, the documents missed by the SOM-based classifier (documents that were supposed to be clustered on a given class label, but were not found under that label), were examined. The missed documents were found to contain only a line or two relating to the topic of interest (for the training set as well as for the test set), but some relevant documents in the test set had been missed for unclear reasons.[16] Those documents that had been assigned to a class without having any relevance to it had some words that frequently occur in the documents that had been accurately classified.

5.3 Decision tree-based classification using stemming and POS-tagging

The third round of experiments was set up and run using the RDS on the walta_1065 corpus and the ten predefined categories of Table 3. We used a bag-of-words approach represented as a vector-space model: each article in the corpus was represented as a (sparse) binary vector where each position in the vector corresponds to a specific unique word that occurs in at least one of the news articles in the corpus.

It was expected that stemming would reduce the size of the representation considerably and also improve classification accuracy. In order to investigate this, we used three different representations for each article. The first representation used the full text for each article and represented it in the form of a binary vector. We will refer to this representation

[16] An inherent problem with ANN-based methods is that the results produced are not human-transparent, i.e., that it is not necessarily easy for a human to understand why the network classified its input as part of a specific output class. In contrast, decision tree-based methods (Sect. 5.3) are inherently human-transparent.

as “full”. The second representation (“stemmed”) instead used a stemmed version of the text to represent each document in the form of a binary vector.

In the third representation (“nouns”), we only used the stemmed form of those words in each news article that had been manually labelled as nouns by the human expert annotators. The reasoning behind this is that nouns tend to carry more information than any other word classes. For example, Hulth (2004) investigated the frequencies of different POS patterns in keywords that had been assigned to documents by professional indexers, and found that as many as 90% of the keywords consisted of nouns and noun phrases. We therefore investigated to which extent it was possible to use only the nouns to represent the text without loosing too much in classification performance compared to that of the other representations.

In the experiments, we compared the three representations using 10-fold cross validation while constructing classification models based on bagging of decision trees. The best performance was obtained using the representation “full” which gave a 69.39% accuracy, followed by “nouns” at 68.92%, and “stemmed” at 68.08% (all with a 34.27% baseline).

6 Discussion

The results of the experiments are summarized in Table 4. Obviously, the results of the three experiments are not directly comparable with each other: the SOM-based experiments were performed on a smaller dataset, with the first experiment using pre-defined classes on a normalized data sample, while the second used the complete corpora (walta_101 for training and walta_105 for testing) for classification using classes dynamically obtained from query labels. Finally, the third experiment used the substantially larger walta_1065 corpus, decision tree-based classification, and 10-fold cross validation over the 1065 document corpus.

The results of our first two experiments are compatible with those of GebreMeskel (2003) who used the standard vector space model and latent semantic indexing for text categorization. He reports that the vector space model gave a precision of 69.1% on the training set. LSI improved the precision to 71.6%. In a previous attempt, Sintayehu (2001) received good results (90.5% average accuracy) using only cosine similarity of weighted document vectors. However, that experiment included only three classes, with a baseline as high as 68.8% (221 of 321 documents taken from the governmental Ethiopian News Agency, ENA, belonged to the largest class).

Most notably, the classifier using the stemmed representation only performed *almost* on par with the other classifiers in the decision tree experiment. It might seem a bit surprising

Table 4 Correctly classified categories

Method	Data	Accuracy (%)
SOM predefined	training	76.5
	test	72.9
SOM query label	training	72.8
	test	69.5
Decision trees	full	69.4
	nouns	68.9
	stemmed	68.1

that the stemmed representation failed to improve the performance of the text classifier on such a morphologically rich language as Amharic. An explanation for the lack of improvement might be that the main morphological variation in Amharic pertains to the verbs, not the nouns. However, the latter are the main carriers of the information content of the language, and hence the main sources of information for the classifiers, which is also partly confirmed by the results of the experiments. This is also in correlation with some experiments on English which indicate that stemming and lemmatisation might not necessarily have significantly positive effects on classification, even if they have on information retrieval tasks (Arampatzis 2001, p. 116).

Further experiments are, however, still needed in order to get a better understanding of the factors that influence text categorization performance for Amharic. Thus the work described here certainly does not present the final word on text classification for Amharic, but rather aim to highlight some of the practical issues that working with a less-researched language entails. In the future, it should be complemented with experiments trying out other algorithms and representations, both the ones that already have proven successful for languages such as English, and others that might still prove their worth for highly-inflectional languages like Amharic.

Going outside Amharic, the work on the Walta corpora can be compared[17] to the work on the English Reuters-21578 news text corpus[18] which contains 21578 documents classified into 135 topic categories.

Cai and Hofmann (2003) used an approach combining LSA with AdaBoost (Freund and Schapire 1997) to obtain a macro-average (average over all classes)[19] F1-score of 74.29% for binary classification on the 50 most common classes of the Reuters-21578 corpus, when training on 9603 documents and testing on 3299 documents (a subset of the corpus also known as the *Modified Apte* split).

Amine et al. (2008) also experimented with Reuters-21578, but using a SOM-based classifier. The best F1-score using only straight-forward n-gram-based classification with cosine similarity was as low as 46.2%, but removing stopwords and mapping the terms in the documents to WordNet concepts improved the F1-score to 62.5%.

Hoi et al. (2006) compare the performance of support vector machines (SVM), logistic regression (LogReg), and a batch mode active (semi-supervised) learning strategy on normalized subsets (100 documents each) of the 10 most common classes in Reuters-21578, again on binary classification. Calculating the macro-averages of the figures given by Hoi et al. (2006), SVM performed at a 65.20% F1-score level and LogReg 63.18%. Adding margin-based active learning increased the F1-scores on 100 samples to 77.00% for SVM and 76.44% for LogReg, while a batch mode active learning algorithm hit a macro-average F1-score of 79.07%.

[17] "Compared" should not be taken in a strict sense when it comes to the performance figures, since the results on English discussed in this section are not straight-forwardly compatible with each other: only the figures inside one particular paper are comparable, while figures from one author to another seldom are. The differences pertain to which data (and which subset of that data) has been used, how many categories were classified, which metrics for the evaluation and which metrics for counting averages were used, as well as if the results apply to binary classification only or directly to multi-class classification. We have aimed to even out some of those differences in the present discussion, though.

[18] Available at www.daviddlewis.com/resources.

[19] Macro-averages gives equal weight to all classes while micro-averages count the averages over all documents and thus gives higher weight to the more common classes. Loosely speaking, the micro-average figures on the Reuters corpus tend to be some 20–25% higher than the macro-averages.

A larger comparison of classifier performance on Reuter-21578 was carried out by Li and Yang (2003). They compared eight classifiers, including SVM, a two-level ANN (i.e., with no hidden layer), logistic regression, and linear regression (also known as Linear Least Squares Fit, LLSF) on the *Modified Apte* split. For the latter three classifiers, Li and Yang (2003) added a regularization term to smooth the penalty of misclassifications. They report the best macro-average F1-score for the regularized LLSF: 63.98% (followed by 62.14% for ANN and 60.84% for LogReg), while their best micro-average figure was obtained by the SVM (88.57%, but both LLSF and ANN also performed just over 88%).

The currently most promising approach to multi-class text categorization on the Reuters corpus that we are aware of is by Subramanya and Bilmes (2008). They only give precision-recall break even points (PRBEP; the values where precision and recall are equal), also testing on the *Modified Apte* portion of Reuters-21578, and the 10 most common classes. For this setup, Subramanya and Bilmes (2008) give PRBEP macro-averages of 66.3% for a strategy using alternating minimization (Csiszár and Tusnády 1984), clearly outperforming other approaches, including SVM (they report 48.9% for SVM, 59.3% for transductive SVM, 59.7% for label propagation, and 60.3% for spectral graph transduction).

7 Summary and conclusions

This paper has described how news articles in Amharic were mined from the Web and how corpora containing these articles were created. The thus created resources were used for several experiments on the classification of Amharic news items. Working with non-English, less-researched languages highlights a number of practical issues that might otherwise receive less attention or go completely unnoticed. Our purpose by running the experiments has not primarily been to develop a cutting-edge text classification system for Amharic, but rather to put the spotlight on some of these issues.

Most importantly, our experiments indicate that stemming plays a less important role than expected for text classification performance for Amharic: we have not been able to show any increased classification performance due to stemming. One possible explanation for this could be that most of the morphological variations in Amharic occurs in the verb, while the nouns are the main carriers of information of relevance for a classification task.

In addition, written languages that do not use a standardized representation require a lot of time and effort in order to create a uniformly represented text corpus. Although there has existed an international standard for *fidel* (Ethiopic script) for the last ten years, this is still a major concern for web-mining, since many of the texts that can be found on the Internet are not compatible with the standard. The lack of standardized representation also means that the few electronic resources that are available (such as stopword lists and machine readable dictionaries) often are incompatible with each other.

The first round of experiments that we carried out on small datasets investigated text classification using SOMs. The best network model performed at a 72.9% level for the test set when trying to classify news items into ten predefined categories. To investigate document clustering as text classification, a SOM model was also trained so as to cluster unseen data around query content, taking 16 queries as class labels. 69.5% of the test data was correctly classified.

It has been claimed that stemming is an important preprocessing step that will allow for improved text categorization accuracy, especially for languages like Amharic that has a rich inflectional morphology (see Sect. 3.4). In order to investigate this further, we performed a second round of experiments where the goal was to see how stemming versus non-stemming affected performance on the task of classifying the news texts into ten predefined categories. We compared three representations using 10-fold cross validation while constructing classification models based on bagging of decision trees. The best accuracy obtained was 69.4%. This was achieved using the full text as representation. A representation using only the nouns performed almost equally well, with an accuracy of 68.9%, confirming the assumption that most of the information required for distinguishing between various categories actually is contained in the nouns. However, stemming did not have that much effect on the performance of the classifier, giving an accuracy of 68.1%.

Acknowledgements The authors would like to thank to Daniel Yacob at the Ge'ez Frontier Foundation; Mesfin Getachew, Dr. Girma Demeke, Dr. Gashaw Kebede, Kibur Lisanu, and Meshesha Legesse at Addis Ababa University; and Gunnar Eriksson, Fredrik Olsson, and Dr. Magnus Sahlgren at the Swedish Institute of Computer Science. The work was partially funded by Sida, the Swedish International Development Cooperation Agency through the ICT support programme of SAREC (the Department for Research Cooperation) and through SPIDER (the Swedish Programme for ICT in Developing Regions), as well as by the Faculty of Informatics at Addis Ababa University.

References

Alemayehu, N., & Willett, P. (2002). Stemming of Amharic words for information retrieval. *Literary and Linguistic Computing, 17*(1), 1–17.

Alemayehu, N., & Willett, P. (2003). The effectiveness of stemming for information retrieval in Amharic. *Emerald Research Register, 37*(4), 254–259.

Argaw, A. A. (2008). Amharic-English information retrieval with pseudo relevance feedback. In C. Peters et al., (Eds.), *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross Language Evaluation Forum, CLEF 2007*, Budapest, Hungary, September 19–21, Revised Selected Papers (pp. 119–126). Berlin/Heidelberg: Springer.

Argaw, A. A., & Asker, L. (2007a). Amharic-English information retrieval. In C. Peters et al., (Eds.), *Evaluation of Multilingual and Multi-modal Information Retrieval: 7th Workshop of the Cross Language Evaluation Forum, CLEF 2006*, Alicante, Spain, September 20–22, 2006, Revised Selected Papers (pp. 43–50). Berlin/Heidelberg: Springer.

Argaw, A. A., & Asker, L. (2007b). An Amharic stemmer: Reducing words to their citation forms. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. Workshop on computational approaches to semitic languages* (pp. 104–110). Prague, Czech Republic: ACL.

Argaw, A. A., Asker, L., Cöster, R., & Karlgren, J. (2005). Dictionary-based Amharic–English information retrieval. In C. Peters et al., (Eds.), *Multilingual Information Access for Text, Speech and Images: 5th Workshop of the Cross Language Evaluation Forum, CLEF 2004*. Bath, UK, September 15–24, 2004, Revised Selected Papers (pp. 143–149). Berlin/Heidelberg: Springer.

Argaw, A. A., Asker, L., Cöster, R., Karlgren, J., & Sahlgren, M. (2006). Dictionary-based Amharic–French information retrieval. In C. Peters et al., (Eds.), *Accessing Multilingual Information Repositories: 6th Workshop of the Cross Language Evaluation Forum, CLEF 2005*. Vienna, Austria, September 21–23, 2005. Revised Selected Papers (pp. 83–92). Berlin/Heidelberg: Springer.

Argaw, A. A., Asker, L., & Eriksson, G. (2003). An empirical approach to building an Amharic treebank. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories* (pp. 205–208). Sweden: Växjö University.

Amine, A., Elberrichi, Z., Simonet, M., & Malki, M. (2008). Evaluation and comparison of concept based and n-grams based text clustering using SOM. *INFOCOMP Journal of Computer Science, 7*(1), 27–35.

Amsalu, S. (2001). *The application of information retrieval techniques to Amharic*. Master of Science Thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia.

Amsalu, S., & Gibbon, D. (2005). Finite state morphology of Amharic. In R. Mitkov (Ed.), *Proceedings of the 5th International Conference on Recent Advances in Natural Language Processing*, Borovets, Bulgaria (pp. 47–51).

Arampatzis, A. (2001). *Adaptive and temporally-dependent document filtering*. Doctor of Philosophy thesis, Department of Information Systems Sciences and Information Retrieval, Katholieke Universiteit Nijmegen, Nijmegen, The Netherlands.

Bayou, A. (2000). *Design and development of word parser for Amharic language*. Master of Science thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia.

Bayu, T. (2002). *Automatic morphological analyser: An experiment using unsupervised and autosegmental approach*. Master of Science thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia.

Bender, M. L., Head, S. W., & Cowley, R. (1976). The Ethiopian writing system. In M. Bender, J. Bowen, R. Cooper, & C. Ferguson (Eds.), *Language in Ethiopia* (pp. 120–129). London, England: Oxford University Press.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review, 37*(4), 573–595.

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

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Cai, L., & Hofmann, T. (2003). Text categorization by boosting automatically extracted concepts. In *Proceedings of the 26th International Conference on Research and Development in Information Retrieval*, (pp. 182–189). Toronto, Canada: ACM SIGIR.

CIA. (2008). *The world factbook—Ethiopia*. Washington, DC: The Central Intelligence Agency [Last updated 12 Feb, 2008].

Cowell, J., & Hussain, F. (2003). Amharic character recognition using a fast signature based algorithm. In *Proceedings of the 7th International Conference on Image Visualization* (pp. 384–389). England: IEEE, London.

Csiszár, I., & Tusnády, G. (1984). Information geometry and alternating minimization procedures. *Statistics and Decisions, 1*, 205–237.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Demeke, G. A., & Getachew, M. (2006). Manual annotation of Amharic news items with part-of-speech tags and its challenges. *ELRC Working Papers, 2*(1), 1–17.

Dumais, S. T. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods Instruments and Computers, 23*(2), 229–236.

Dumais, S. T. (1995). Using LSI for information filtering: TREC-3 experiments. In D. K. Harman (Ed.), *Proceedings of the 3rd Text Retrieval Conference* (pp. 219–230). Gaithersburg, MD: National Institute of Standards and Technology.

Firdyiwek, Y., & Yacob, D. (1993). The Ethiopian script in ASCII. *Journal of EthioSciences*, *3*(1). http://www.abyssiniacybergateway.net/fidel/sera.ps [Last updated 1 Jan 1997].

Fissaha, S., & Haller, J. (2003a). Amharic verb lexicon in the context of machine translation. In *Proceedings of the 10th Conference on Traitement Automatique des Langues Naturelles*, Batz-sur-Mer, France (Vol. 2, pp. 183–192).

Fissaha, S., & Haller, J. (2003b). Application of corpus-based techniques to Amharic texts. In *Proceedings of the 9th Machine Translation Summit, New Orleans, Louisiana. Workshop on Machine Translation for Semitic Languages: Issues and Approaches*. http://www.amtaweb.org/summit/WS2/Fissaya+Haller_paper.pdf.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Sciences, 55*(1), 119–139.

Furzey, J. (1996). *Enpowering socio-economic development in Africa utilizing information technology*. A country study for the United Nations Economic Commission for Africa, African Studies Center, University of Pennsylvania.

Gaustad, T., & Bouma, G. (2002). Accurate stemming of Dutch for text classification. In M. Theune, A. Nijholt, & H. Hondorp (Eds.), *Computational Linguistics in the Netherlands 2001: Selected Papers from the Twelfth CLIN Meeting*, Rodopi, Amsterdam, The Netherlands (pp. 104–117).

GebreMeskel, T. (2003). *Amharic text retrieval: An experiment using latent semantic indexing (LSI) with singular value decomposition (SVD)*. Master of Science thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia.

Gordon, R. G. Jr. (Ed.). (2005). *Ethnologue: languages of the world* (15th ed.). Dallas, TX: SIL International.

Hoi, S. C. H., Jin, R., & Lyu, M. R. (2006). Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th International World Wide Web Conference*, Edinburgh, Scotland (pp. 633–642).

Honkela, T., Kaski, S., Lagus, K., & Kohonen, T. (1997). WEBSOM—Self-Organizing Maps of document collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*, Espoo, Finland (pp. 310–315).

Hudson, G. (1999). Linguistic analysis of the 1994 Ethiopian census. *Northeast African Studies, 6*(3), 89–107.

Hudson, G. (2006). 75 Ethiopian languages: 19 Cushitic, 20 Nilosaharan, 23 Omotic, 12 Semitic, and 1 unclassified. http://www.msu.edu/hudson/Ethlgslist.htm [Last updated 29 Dec, 2006].

Hulth, A. (2004). *Combining machine learning and natural language processing for automatic keyword extraction*. Doctor of Philosophy thesis, Stockholm University and the Royal Institute of Technology, Deparment of Computer and Systems Sciences, Stockholm, Sweden.

Karlgren, J., & Sahlgren, M. (2001). From words to understanding. In Y. Uesaka, P. Kanerva, & H. Asoh (Eds.), *Foundations of Real World Intelligence* (pp. 294–308). Stanford California: CSLI publications.

Kaski, S., Honkela, T., Lagus, K., & Kohonen, T. (1996). Creating an order in digital libraries with Self-Organizing Maps. In *Proceedings of the World Congress on Neural Networks*, San Diego, California (pp. 814–817).

Kohonen, T. (1999). *Self-organization and associative memory* (3rd ed.). Heidelberg, Germany: Springer.

Kohonen, T. (2001). *Self-Organizing Maps* (3rd ed.). Berlin, Germany: Springer.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., & Saarela, A. (2000). Self organization of a massive document collection. *IEEE Transactions on Neural Networks, 11*(3), 574–585.

Larkey, L. S. (2002). Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. In *Proceedings of the 25th International Conference on Research and Development in Information Retrieval* (pp. 275–282). Tampere, Finland: ACM SIGIR.

Li, F., & Yang, Y. (2003). A loss function analysis for classification methods in text categorization. In *Proceedings of the 20th International Conference on Machine Learning*, Washington, D.C. (pp. 472–479).

Lin, X., Soergel, D., & Marchionini, G. (1991). A self-organizing semantic map for information retrieval. In *Proceedings of the 14th International Conference on Research and Development in Information Retrieval* (pp. 262–269). Chicago, IL: ACM SIGIR.

Miniwatts Marketing Group. (2008). Internet world users by language. http://www.internetworldstats.com/languages.htm [Last updated 30 Jun, 2008].

Negga, W. (2008). *Wazéma System: an Ethiopian computer writing system for Windows NT/2000/XP/Vista Version 2.1*. Croydon, England. www.gzamargna.net.

Ng, H. T., Goh, W. B., & Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th International Conference on Research and Development in Information Retrieval* (pp. 67–73). Philadelphia, PA: ACM SIGIR.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Ruiz, M. E., & Srinivasan, P. (1999). Hierarchical neural networks for text categorization. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval* (pp. 281–282). Berkeley, CA: ACM SIGIR.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. Rumelhart, & J. McClelland (Eds.), *Parallel distributed processing* (Vol. 1, pp. 318–362). Cambridge, MA: MIT Press.

Salton, G., & McGill, M. (1983). *Introduction to modern information retrieval*. New York, NY: McGraw-Hill.

Schütze, H., Hull, D. A., & Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In E. A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th International Conference on Research and Development in Information Retrieval* (pp. 229–237). Seattle, WA: ACM SIGIR.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Sintayehu, Z. (2001). *Automatic classification of Amharic news items: The case of the Ethiopian News Agency*. Master of Science thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia.

Subramanya, A., & Bilmes, J. (2008). Soft-supervised learning for text classification. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (pp. 1090–1099). Honolulu, Hawaii: ACL.

Syiam, M. M., Fayed, Z. T., & Habib, M. B. (2006). An intelligent system for Arabic text classification. *International Journal of Intelligent Computing and Information Sciences, 6*(1), 1–19.

Tambouratzis, G., Hairetakis, N., Markantonatou, S., & Carayannis, G. (2003). Applying the SOM model to text classification according to register and stylistic content. *International Journal of Neural Systems, 13*(1), 1–11.

Xu, J., Fraser, A., & Weischedel, R. (2002). Empirical studies in strategies for Arabic retrieval. In *Proceedings of the 25th International Conference on Research and Development in Information Retrieval* (pp. 269–274). Tampere, Finland: ACM SIGIR.

Yacob, D. (1997). The system for Ethiopic representation in ASCII—1997 standard. http://www.abyssiniacybergateway.net/fidel/sera-97.html.

Yacob, D. (2005). Developments towards an electronic Amharic corpus. In *Proceedings of the 12th Conference on Traitement Automatique des Langues Naturelles, Dourdan, France. Workshop on Under-Resourced Languages*. http://yacob.org/papers/DanielYacob-TALN2005.pdf.