Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models

Seid Muhie Yimam[1], **Hizkiel Mitiku Alemayehu**[3],
Abinew Ali Ayele[1,2] and **Chris Biemann**[1]
Universität Hamburg, Germany[1],
Bahir Dar University, Ethiopia[2],
Università di Bologna[3], Italy
yimam@informatik.uni-hamburg.de

Abstract

This paper presents the study of sentiment analysis for Amharic social media texts. As the number of social media users is ever-increasing, social media platforms would like to understand the latent meaning and sentiments of a text to enhance decision-making procedures. However, low-resource languages such as Amharic have received less attention due to several reasons such as lack of well-annotated datasets, unavailability of computing resources, and fewer or no expert researchers in the area. This research addresses three main research questions. We first explore the suitability of existing tools for the sentiment analysis task. Annotation tools are scarce to support large-scale annotation tasks in Amharic. Also, the existing crowdsourcing platforms do not support Amharic text annotation. Hence, we build a social-network-friendly annotation tool called 'ASAB' using the Telegram bot. We collect 9.4k tweets, where each tweet is annotated by three Telegram users. Moreover, we explore the suitability of machine learning approaches for Amharic sentiment analysis. The FLAIR deep learning text classifier, based on network embeddings that are computed from a distributional thesaurus, outperforms other supervised classifiers. We further investigate the challenges in building a sentiment analysis system for Amharic and we found that the widespread usage of sarcasm and figurative speech are the main issues in dealing with the problem. To advance the sentiment analysis research in Amharic and other related low-resource languages, we release the dataset, the annotation tool, source code, and models publicly under a permissive.

1 Introduction

Sentiment analysis is the task of detecting the orientation of someone's opinion and analyzing the emotions, feelings, and attitudes of a speaker or a writer in a piece of information concerning a certain situation, object, or event (Pandey and Govilkar, 2015). The most widely adopted approach in sentiment analysis to explore opinions is by employing very large datasets that target products and services, political, economical, social, and cultural feelings (Kauffmann et al., 2019; Caetano et al., 2018; Lennox et al., 2020). Understanding the sentiment of text content helps governments, organizations, and institutions to make correct, timely, and economical decisions (De Souza Bermejo et al., 2019).

Sentiment analysis has been researched intensively for resource-rich languages such as English and German (Liu, 2012; Feldman, 2013; Tymann et al., 2019; Akhtar et al., 2016; D'Andrea et al., 2015; Wojatzki et al., 2017). However, existing models and approaches for most resource-rich languages can not easily be adapted to Amharic due to context variations in language, culture, and technology, especially for social media communication (Gangula and Mamidi, 2018). The works by Gezmu et al. (2018) and Abate and Assabie (2014) indicate that natural language processing (NLP) components, such as part of speech tagging (POS), named entity recognition (NER), and sentiment analysis are nontrivial due to the morphological, syntactic, and semantic complexity of the language. The absence of well-annotated corpora and NLP resources like parsers and taggers make Amharic sentiment analysis still challenging (Gezmu et al., 2018; Pandey and Govilkar, 2015).

In addition to resource scarcity and morphological challenges for Amharic, the nature and structure of social media texts is also another bottleneck by itself (Nakov et al., 2013). On one hand, the statements in social media are noisy and do not usually follow proper rules, contain spelling errors, mixed scripts, and non-standard abbreviations (Badaro et al., 2019). On the other hand, punctuation marks, emoticons, emojis, and even other special symbols are very crucial to portray sentiment orientation in context. These issues make social media texts challenging for sentiment analysis tasks (Virmani et al., 2017; Badaro et al., 2019).

In general, sentiment analysis research for low-resource languages is under-researched. For Amharic, the first attempt of sentiment analysis is described by Philemon and Mulugeta (2014), who focused on the prediction of sentiment polarity. They have used a Naïve Bayes classifier based on unigram and bigram features on 600 tweets. While the datasets they have used are very small in size, it is not also publicly available for further investigation. Considering the importance of sentiment analysis tasks for several applications, it is essential to properly explore the challenges, develop readily usable models, and describe future challenges.

The main motivation of this work is to address sentiments analysis issues of Amharic comments, which become widely available on Facebook and Twitter. Another motivation stems from annotation tool challenges, where one of the major problems is the limited bandwidth in Ethiopia. Also, people favor smartphones over desktop applications. Hence classical web-based annotation tools will not be suitable. Moreover, the majority of crowdsourcing platforms, for example, MTurk, do not support workers and task requesters from Ethiopia.

The main foci of this work are, 1) to explore different annotation strategies and tools for low-resource languages, 2) to collect a large dataset, and 3) to build different machine learning models for Amharic sentiment analysis. We will also publicly release the collected datasets, annotation tools, pre-trained models, and the associated source codes to advance the sentiment analysis research in Amharic. In this work, we will address the following research questions: **RQ1**) How to identify appropriate data annotation tools and collect large-scale sentiment analysis corpus for Amharic? **RQ2**) Which machine learning model is most appropriate for Amharic sentiment analysis? **RQ3**) What are the main challenges in Amharic sentiment analysis?

The remainder of the paper is organized as follows. In Section 2, we will discuss the data acquisition, annotation strategy, annotation tools, and characteristics of the annotated data. In Section 3, the main approaches in the development of sentiment analysis tasks and related works in Amharic are presented. In Section 4, we have discussed the different experimental setups that are used to build different models. While Section 5 discusses the experimental results and analyses the different errors associated with the proposed models, Section 6 briefly summarises the main findings of the study.

2 Data Collection and Analysis

In this section, we will briefly describe the data collection and sampling strategies we have followed to annotate the dataset. Furthermore, we discuss the limitations of the existing annotation tools and proposed a novel annotation tool that is appropriate for low-resource language data collection.

2.1 Data Acquisition and Dataset Characteristics

The data source for this study is the Ethiopic Twitter Dataset for Amharic (ETD-AM) that we have collected using the Twitter API and previously introduced in Yimam et al. (2019). The collection spans two months of data (December 2019 and January 2020) that are selected purposefully due to specific political and social events happening in Ethiopia. During those months: 1) The current Ethiopian Prime Minister Dr. Abiy Ahmed has received the 100^{th} Nobel peace prize. 2) Around 17 university students were kidnapped. 3) The ruling party EPRDF was resolved and transformed itself to 'prosperity party', and 4) many other socio-political changes such as religious conflicts were aggravated and were intensely discussed in mainstream and social media platforms. In total, we have collected more than 300k tweets.

Because of resource limitations for annotating 300k tweets, we have selected tweets based on an extended sentiment lexicon (435 positive and 660 negative lexicon entries) generated by Gebremeskel

Positive	English Translation	Lexicon	English Translation
እመርታ	success	ከንቱ	in vain
ሀሴት	happiness	ደባሪ	boring
ማራኪ	attractive	ወሬኛ	chatterbox
የሚደነቅ	amazing	አስጨናቂ	anxiety
ተስፋ	hope	ፈታኝ	challenging

Table 1: Positive and negative examples from the constructed sentiment lexicon.

(2010). Table 1 shows sample positive and negative examples with their English translation from the Amharic sentiment lexicon. For the final dataset annotation, we have considered tweets that contain at least one sentiment lexicon entry. Our final dataset is comprised of 9.4k tweets, where each tweet is annotated by three different users.

2.2 Annotation Strategy

Annotation is one of the most laborious and complex tasks in the process of developing machine learning components (Wang et al., 2013; Finlayson and Erjavec, 2017). In countries where technology is not fully exploited, people usually conduct surveys or annotations using manual labor, for instance, by answering to a printed version of questionnaires (Dickinson et al., 2019; Lupu and Michelitch, 2018). For the machine learning approach, this entails that the questionnaire should be transformed into a digital format that takes a lot of effort and could even introduce errors during encoding (Lupu and Michelitch, 2018).

To annotate a large number of datasets with minimum compensation, one has to use crowdsourcing platforms (Wang et al., 2013; Sabou et al., 2014). However, most of the crowdsourcing platforms require a complicated registration process, especially to incorporate transaction of payments for workers. For example, Amazon Mechanical Turk (MTurk) does not allow registration of task requesters from Ethiopia[1]. Moreover, we can not conduct our annotation using MTurk for Amharic sentiment analysis as there are no adequate online payment methods in Ethiopia and a lack of Amharic speakers on MTurk.

Hence, our first possible approach was to label our dataset using a spreadsheet application where we distribute our data in a tabular format (see Figure 1a). The users found that annotating the data using a spreadsheet is time-consuming, error-prone, and difficult to interact with. Therefore, we build a specific web-based annotation tool that can facilitate the annotation process. Figure 1b shows the annotation tool with the annotation instructions, the tweets to annotate, and the annotation types (sentiment classes). While this approach was more attractive and easy to use, the main challenge was that a lot of users in Ethiopia do not use desktop computers. Besides, most users have no experience in using web browsers with their mobile phones, as only 10% of smartphone users spent time in web browsers, and the rest 90% mostly spent time only in applications (Wurmser, 2018). This is a critical limitation to collect large-scale datasets from users with various backgrounds.

As the number of mobile device users is tremendously increasing (Sabou et al., 2014), social media-based annotation tools for the data collection could be an option. We choose chatbots as a great candidate (Fadhil and Villafiorita, 2017) to perform sentiment analysis annotations and extract emotional context from a text corpus. A lot of users adopt Telegram Messenger to channel and share data for their followers. We opt to build a Telegram Bot-based chatbot for our work due to its applicability for the task, availability of active users, and simplicity of use.

2.3 Design of the Annotation Tool: Amharic Sentiment Annotator Bot (ASAB)

Telegram Bot[2] supports a client-server architecture, where it enables users with Telegram accounts to directly communicate with 'Telegram Server'. We have implemented an application server that can communicate with the 'Telegram Server' using different endpoints. The '/start' endpoint initiates the communication with our application server while the '/instruction' endpoint displays the annotation in-

[1] https://blog.mturk.com/mturk-is-now-available-to-requesters-from-43-countries-77d16e6a164e
[2] https://core.telegram.org/bots/api

(a) Excel-sheet for annotation (b) Web annotator interface (c) ASAB interface

Figure 1: User interfaces of the different annotation tools we have used to annotate Amharic sentiments.

structions. The '/update' and '/end' endpoints enable us to receive responses (annotations) from the users and quit the communication with our server respectively. The user interface of ASAB as it can be seen on the user's mobile device is depicted in Figure 1c.

ASAB is designed to support rewards (in the form of mobile card vouchers) as soon as the user successfully annotated enough tweets. After conducting a pilot study, the number of tweets to annotate and get a reward was set at 50. When the worker completes the task, the voucher will be displayed instantly to the user.

In general, controlling the quality of the annotated data by blocking bad workers or spammers is crucial on crowdsourcing platforms (Stenetorp et al., 2012; Hovy and Lavid, 2010). The chatbot-based annotation is much more restrictive, mainly designed with built-in control mechanisms to assure annotation quality. ASAB integrates a controlling strategy in the form of control questions. For every 6 tweets, we have included one control question with a known answer. Users who have made 3 consecutive mistakes will receive a warning message. If the user still keeps on randomly annotating the tweets, he/she will be blocked after the fourth attempt.

Another challenge in the crowdsourcing annotation framework is the preparation of concise annotation instructions that users can read and understand it instantly. However, it is very tough to display long instructions and annotation examples that can fit mobile devices. To mitigate this issue, we have published a separate web page that shows detailed instructions and annotation examples[3]. We have also prepared a YouTube video demonstrating the annotation steps[4]. Even if such elaborated instructions exist, users are tempted to start the annotation task immediately. We partly address this limitation by presenting minimal instructions and examples every time a user restarts the annotation task, before displaying the first tweet. Besides, since the texts are collected from Twitter and as we do not have direct control of the content, some of the tweets might not be appropriate for users. Thus, users are warned about such texts and they have to agree before proceeding to the main annotation task.

2.4 Analysis of Annotated Data

We have collected 9.4k tweets, a total of 143,848 words (total tokens), and 45,525 types (unique tokens) that are annotated by employing ASAB where each tweet is annotated by three users. In total, 92 Telegram users have visited ASAB while 53 users (58% of the total users) completed at least 50 tweets (rewarded one mobile card voucher). Furthermore, the system blocked 4 users who have made consecutive mistakes while annotating the control questions.

ASAB is the first of its kind to conduct surveys based on a specific reward scheme, which is mobile card vouchers. In the beginning, it was challenging to convince annotators regarding the reward. We have advertised the task on several social media networks such as Facebook and Twitter. We have also created a Telegram channel group to announce the release of new tasks and to answer questions raised by the annotators. Once the annotators obtain their first mobile card voucher as a reward, we have observed that the popularity of ASAB has increased and users rely on our system. We have conducted the annotation in batches. In the first batch, it took more than two days to complete the annotation. Whereas, in the second and third iterations, the annotation was completed in less than 6 hours for each iteration.

From the annotated dataset, we have observed that 2 or more annotators agreed on the 7,317 tweets

[3] https://annotation-wq.github.io/

[4] Annotation instruction of ASAB in the Amharic language is available here: https://annotation-wq.github.io/

(78%) while all the three annotators disagree on the remaining 2,072 tweets (22%). As an indicator of annotation quality, Fleiss' Kappa measurement metrics are used to evaluate the inter-annotator agreement (IAA) of the sentiment corpus. Fleiss' Kappa is chosen for its suitability to compute the inter-annotator agreement for multi-rater annotators (number of annotators larger than two) (Pustejovsky and Stubbs, 2012). A substantial level of inter-annotator agreement (0.785) is achieved on the four sentiment labels. This inter-annotator agreement value is relatively higher (very close to an almost perfect agreement, which spans from 0.81-1.00). The score indicates that human annotators can associate meanings and sentiments to texts despite the lack of background contexts for the tweets and the complex properties of social media texts such as incomplete phrases and sentences, mixed script, figurative and sarcastic speeches, and spelling and grammar errors. Even though the ASAB tool and the rewarding scheme are completely new to the annotators, the high IAA result indicates that ASAB is appropriate for sentiment analysis annotation tasks, which addresses our research question 1 (**RQ1**).

Types	**Example tweets**	**Literal Translations**
Sarcasm	ቅቤው ምናባቱ ሆነ ደሞ…	oh what happens to the butter again...
Sarcasm	እንኳን ደህና መጣህ! ከኤልፓ በፊት ማን ይቀበልህ ታድያ ??	Welcome! who should welcome you before EEPC ??
Sarcasm	?????? ድንጋይ ከመወርወር የተሻለ ነው መቼም} ??????	?????? It is better than throwing a stone ??????
Figurative	መላላጫ እያለ የምን እግር ነው	why the leg while rump is there
Figurative	የአዞ እንባ መሆኑ ነው???:: ድንቅ ነው	is it crocodile's tears??. it is amusing
Mixed-Script	ዩዝ & ስሮው ነው.	It is 'use-and-throw.'
Mixed-Script	hehehe የኔ ብቻ መስሎኝ ነበር ??	hehehe I thought it was only mine ??
Incomplete	አቤት ወያኔ …	Oh rebels ...
Incomplete	የሞላ የሞላ	already full already full

Table 2: Difficult tweets on which all the three annotators disagree on the labels.

We tried to analyze some of the tweets on which the three annotators fail to agree as can be seen in Table 2. For example, the tweets of the 'Sarcasm' group are difficult to interpret as they need extensive background knowledge. For instance, the first example might have a different interpretation depending on what "butter" refers to. Figurative speech is also very common in the Amharic language that is also present in abundance in the Twitter dataset. 'Mixed-Script' and 'Incomplete' tweets also need more background information as well as knowledge of other languages. For instance, in Table 2, the first 'Mixed-Script' example represents a tweet, which is an English sentence transliterated in 'Fidel' script.

3 Related Works

According to Liu (2012), opinion mining is a field of study that analyzes people's opinions, sentiments, evaluations, attitudes, and emotions from written language. It can be asserted that the expansion of digital technology and the volume of data made available by such technologies affects trends of sentiment analysis task. The work by Feldman (2013) differentiates sentiment analysis into four levels.

Document-Level Sentiment Analysis: This is the simplest form of sentiment analysis and it is assumed that the document contains an opinion on one main object expressed by the author of the document.

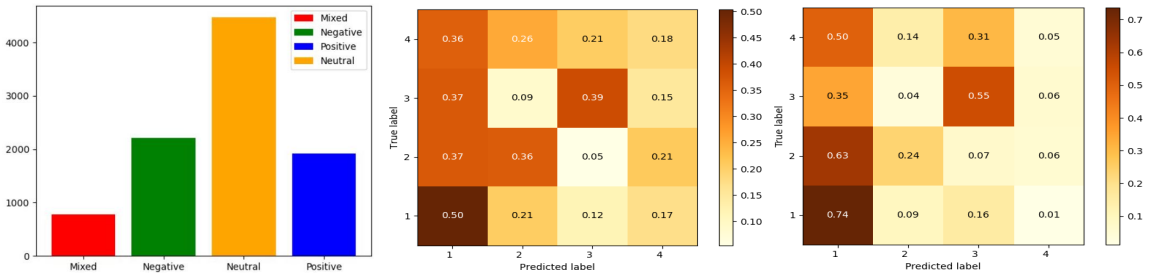
Sentence-Level Sentiment Analysis: A single document may contain multiple opinions even about the same entities. When we want to have a more generalized view of the different opinions expressed in the document about the entities, we must move to the sentence level.

Aspect-Based Sentiment Analysis: The above methods work when whether the whole document or each sentence is discussing a single concept. However, in many cases, people talk about entities that have many aspects (attributes) and they have a different opinion about each of the aspects.

Comparative Sentiment Analysis: Usually, people do not give opinions about a product directly instead they give comparable opinions. Sentiment analysis systems for such kind of situation identify sentences that contain the comparative opinion and extract the preferred entity(-ies) in each opinion.

3.1 Sentiment Classification Approaches

The work by Anitha et al. (2013) defines sentiment classification as a task of categorizing sentimental text in a specific document into 'positive' or 'negative' classes. This approach completely ignores the



(a) Frequency of annotation classes (b) Confusion matrix - NearestCentroid (c) Confusion matrix - role2Vec

Figure 2: Annotation statistics and confusion matrices using the 'NearestCentroid' and 'role2Vec' models. In the confusion matrices, 1, 2, 3, and 4 represents Neutral, Negative, Positive, and Mixed resp.

'neutral' classes from the categories. However, other researchers argue that learning from negative and positive examples alone will not permit accurate classification of neutral examples (Koppel and Schler, 2005). Moreover, the use of neutral training examples in the machine learning approach favors a better distinction between positive and negative instances. Furthermore, the work by De Souza Bermejo et al. (2019) incorporates another class called 'mixed' that can help to capture opinions expressed with both positive and negative sentiment.

We adopt the sentiment classes defined by De Souza Bermejo et al. (2019) to explore, which classes mostly fit the tweets in the Amharic language. However, as depicted in Figure 2a, the number of mixed examples is much lower than the rest. The type of tweets labeled as mixed by the annotators and their significance during the model evaluation will be discussed in Section 5.2.

3.2 Sentiment Analysis for Amharic

The task of sentiment analysis for low-resource languages like Amharic remains challenging due to the lack of publicly available datasets and the unavailability of required NLP tools. Moreover, there are no attempts of analyzing the complexities of sentiment analysis on social media texts (e.g. Twitter dataset), as the intended meaning is highly context-dependent and influenced by the user experience (Gangula and Mamidi, 2018). Some of the existing works in Amharic either targeted the generation of sentiment lexicon or limited to the manual analysis of very small social media texts.

The work of Alemneh et al. (2019) focuses on the generation of Amharic sentiment lexicon using the English sentiment lexicon. The English lexicon is translated to Amharic using a bilingual English-Amharic dictionary. The work proposed by Gebremeskel (2010) describes a rule-based sentiment polarity classification system. Using movie reviews, 955 sentiment lexicon entries are generated. The system then tries to detect the presence and absence of the positive and negative terms from the lexicon to classify the polarity of the document. We have extended this sentiment lexicon (to a total of 1194 entries) to select tweets for further annotation. We have found that even filtering tweets with the sentiment lexicon, the majority of the tweets are annotated as 'neutral'. This indicates that the sole use of sentiment lexicon is not enough to build a proper sentiment classification model.

4 Experimental Setup

For the sentiment classification task, we follow the document classification approach as it is mainly addressed in the literature (Prabowo and Thelwall, 2009). Our tweets mainly constitute of one or two sentences, which are limited to the maximum length of texts allowed by Twitter. We explore both classical supervised machine learning and deep learning approaches for the classification task. Instead of manually crafted features, we have used automatic text representation techniques such as Term-Frequency Inverse Document Frequency (TF-IDF) and word representations (embeddings). The TF-IDF document representation is produced using the scikit-learn (Pedregosa et al., 2011) *CountVectorizer* and *TF-IDFTransformer* built-in methods. For the TF-IDF computation, each tweet is considered as a document. To build word2vec-based word representations, we have collected around 15 Million sentences from dif-

ferent sources, such as a News dataset using the Scrapy Python API[5], YouTube comments using the YouTube Data API[6], and a Twitter dataset using the Twitter API[7]. We collect the datasets every day and store relevant metadata such as the date, title, and language of the dataset.

As we have discussed in Section 2 and Section 3.1, each tweet is annotated with 'Positive', 'Negative', 'Neutral', and 'Mixed' sentiment classes. The 9.4k annotated tweets are further split into training, development, and test instances using an 80:10:10 split. We have used the development dataset to optimize the learning algorithms. All the results reported in the remaining sections are based on the test dataset instances.

After error analysis, we found out that tweets annotated with the 'Mixed' class are noises, which can be regarded as 'Positive', 'Negative', or 'Neutral'. Hence, we further cleanse the dataset and exclude tweets labeled as 'Mixed', which leads to a final dataset of size 8.6k tweets. We follow the same split and report the results accordingly (see column 'Cleaned' in Table 3).

4.1 Baseline Methods

We build the baseline models using the scikit-learn Python machine learning framework. The 'DummyClassifier' includes the following strategies to build the baseline models: 1) **Stratified**: Generates predictions by respecting the training set's class distribution. 2) **Uniform**: Generates predictions uniformly at random. 3) **Most frequent**: Always predicts the most frequent label in the training set.

4.2 Supervised Approach

In this work, we do not consider handcrafted features, such as N-gram features, lexical and syntactic features, word frequencies, and sentiment lexicon entries to train a supervised machine learning model. Instead, we rely on word representations that are obtained using different approaches. For the supervised machine learning approach, we have used TF-IDF and word embeddings. We have used the following machine learning algorithms from scikit-learn based on the TF-IDF feature vectors.

Support Vector Machine (SVM): It is a machine learning algorithm for two-group classification problems. The 'SGDClassifier' in sci-kit learn supports multiclass probability estimation, which is derived from the binary 'one-versus-rest' estimates by simpler normalization (Cortes and Vapnik, 1995; Zadrozny and Elkan, 2002). The hyperparameters used for final model include (loss='modified_huber',penalty='l2',alpha=1e-3,and max_iter=100).

K-Nearest Neighbor (KNN): KNN works by determining the nearest neighbors to a given query and use those classes to predict the right class of the query (Cunningham and Delany, 2020). We use n_neighbors=10 and weights='distance' as a hyperparameter to build the model.

Logistic Regression: It is a common supervised learning technique that classifies a text into two or more classes. This technique employs a discriminative classification approach (Jurafsky and Martin, 2019). The model is tunned with the following parameters: solver='newtoncg', multi_class='multinomial', and max_iter=100.

Nearestcentroid: It is a simple machine learning approach that achieves classification by assuming the locally constant class conditional probability. It calculates the mean of a given observation and assigns it to to the class with the nearest centroid (Pedregosa et al., 2011). The default parameter settings are used to build the final model.

4.3 Deep Learning Approach

Over the previous couple of years, many NLP applications start employing deep learning approaches for their automation components (Young et al., 2018). Unlike the approaches in classical supervised machine learning, the use of deep learning methods avoids the unnecessary hand-crafted feature engineering pre-process steps (Minaee et al., 2020). The effectiveness of deep learning models is improving over time as newer algorithms, better hardware infrastructure, and above all, a substantially large amount of free texts are being generated (Torfi et al., 2020). Unlike high-resource languages such as English

[5] https://scrapy.org/
[6] https://developers.google.com/youtube/v3
[7] https://developer.twitter.com/en

and German, the impacts, limitations, and perspectives of using deep learning models in sentiment analysis for low-resource languages, particularly for Amharic, is not yet exploited. In this work, three types of embeddings, namely static (Mikolov et al., 2013), contextualized (Devlin et al., 2019), and network (Hamilton et al., 2017) embeddings are considered to build different deep learning models for the sentiment classification.

Word2Vec: Word2vec helps to learn word representations (word embeddings) that employ a two-layer neural network architecture (Mikolov et al., 2013). Embeddings can be computed using a large set of texts as an input to the neural network architecture. We have used the Gensim Python Library (Řehůřek and Sojka, 2011) to train the embeddings using the default parameters.

Network embeddings: Network embeddings allow representing nodes in a graph in the form of low dimensional representation (embeddings) to maintain the relationship of nodes (Hamilton et al., 2017; Sevgili et al., 2019; Cai et al., 2018). In this paper, we first compute the network-based distributional thesaurus (DT) (Ruppert et al., 2015) and later convert the DT to a network embeddings following the approach by Jana and Goyal (2018).

Contextual embeddings: With the release of Google's Bidirectional Encoder Representations from Transformer (BERT) (Devlin et al., 2019), word representation strategies have shifted from the traditional static embeddings to a contextualized embedding representation. BERT-like models have an advantage over static embeddings as they can accommodate different embedding representation for the same word based on its context. In this task, we have used RoBERTa (A Robustly Optimized BERT Pre-training Approach), which is a replication of BERT developed by Facebook (Liu et al., 2019). Unlike BERT, RoBERTa removed the 'next sentence prediction' functionality, allowing training on longer sequences, and dynamically changing the masking patterns. We also train and fine-tune contextual embedding models using the FLAIR framework (Akbik et al., 2018; Akbik et al., 2019)

Document Embedding: Unlike word embeddings, document embeddings provide a single embedding for the entire text (sentence, paragraph, or the entire document). The FLAIR framework has document embeddings implementations, such as 'DocumentPoolEmbeddings, which produces document embeddings from pooled word embeddings, and 'DocumentLSTMEmbeddings, which provides document embeddings from LSTM based on word embeddings (Akbik et al., 2019).

5 Results and Discussions

Table 3 shows the experimental results based on the baseline, supervised, and deep learning models. As we can see in the table, both the supervised and deep learning approaches outperform the three baseline systems. We have followed the suggestions by De Souza Bermejo et al. (2019) to categorize sentiment classes into 'positive', 'negative', 'neutral', and 'mixed'. However, we have observed that the number of tweets annotated as 'mixed' are substantially smaller than the other classes, see Figure 2a. Due to this reason, we have conducted two variant experiments, 1) with the whole dataset ('All' column in Table 3) and 2) removing all the 'mixed' instances from the training ('Cleaned' column). For both datasets ('All' and 'Cleaned'), the models based on the deep learning approach (models with prefix **F-**) performs better than the supervised machine learning approaches. Moreover, the models perform better when the 'mixed' sentiment classes are removed from the dataset, hence, 'mixed' sentiment classes can be considered as noise for the model.

5.1 Discussion

Concerning the dataset, we have observed that with proper control questions integration and concise and clear instructions, the Telegram bot-based annotation strategy is viable for low-resource language data collection. The reward technique in ASAB can be enhanced with a more general vouchering system such as incorporating Amazon vouchers or even direct monetary rewards that will be more attractive. The significant contributions of the tool are 1) ASAB, a Telegram bot-based annotation tool does not require installation of extra applications. 2) Verification and authentication of users will be managed directly by the Telegram server. 3) It is very convenient to directly communicate with the users in case errors or problems. 4) The annotation can be conducted even with a very slow internet connection.

Model	All				Cleaned			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
	TF-IDF representation							
Logistic-Regression	53.04	34.76	39.99	37.19	58.42	46.80	60.88	52.92
Random-Forest	50.48	33.97	37.75	35.76	54.36	44.59	52.17	48.09
K-Nearest-Neighbor	46.75	34.99	33.53	34.25	51.68	49.85	50.22	50.03
NearestCentroid	42.07	35.96	37.61	36.77	51.92	47.36	49.20	48.26
Support-vector-machine	52.40	33.54	41.81	37.22	53.19	35.44	46.42	40.20
	Word Embeddings							
fastText	49.63	36.26	35.08	35.66	54.12	48.35	51.33	49.79
F-Word2Vec	53.57	**41.76**	43.01	**42.38**	55.40	55.54	54.91	55.22
F-fastText	51.22	31.96	51.80	39.53	56.68	43.50	67.81	53.00
	Contextual Embeddings							
F-AmharicFlair	53.04	38.66	45.85	41.95	58.65	53.24	59.25	56.09
F-Multi-flair	53.67	36.41	41.89	38.96	55.75	46.96	55.05	50.68
F-Multi-flair-Finetuned	54.42	38.10	**45.87**	41.63	59.81	54.49	59.58	56.92
Amharic-Roberta	53.89	40.05	39.71	39.88	56.33	46.62	56.39	51.04
	Graph Embeddings							
F-DeepWalk	**54.53**	38.49	41.08	39.74	58.65	55.89	57.71	56.78
F-Role2Vec	52.40	39.45	41.16	40.29	**60.51**	**56.26**	**60.89**	**58.48**
	Baselines							
Stratified	34.72	26.59	26.71	26.65	39.02	33.79	33.80	33.80
Uniform	23.54	24.19	23.69	23.94	30.89	29.72	30.96	30.33
Mostfrequent	47.60	25.00	11.90	16.13	51.92	33.33	17.31	22.78

Table 3: Experimental results using the test set. The 'All' column shows performance on the four sentiment classes and the 'Cleaned' column shows the performance on three classes ('mixed' class removed').

Regarding the developed classifier models, we have observed that: 1) Deep learning models generally outperform the classical supervised models. 2) Fine-tuning pre-trained models perform very well (compare the results of 'F-Multi-flair' and 'F-Multi-Flair-Finetuned' models). 3) Most interesting, models based on network embeddings perform better than Word2Vec embeddings (see 'F-DeepWalk' and 'F-Role2Vec' models). Hence, using network embeddings in a deep learning setup works better for the Amharic sentiment analysis task, which addresses our research question 2 (**RQ2**).

As it can be seen from Figure 2b and Figure 2c, the supervised machine learning model based on 'NearestCentroid' predict more 'mixed' sentiment classes than the deep learning model based on 'role2Vec' document embeddings. However, 'NearestCentroid' fails to correctly classify the 'neutral' and 'positive' sentiment classes. Hence, we suggest that it is better to use ensemble methods based on supervised and deep learning models to improve classification performance.

5.2 Error Analysis

We have randomly selected tweets where the model prediction (using the 'role2Vec' model) and the user annotations differ. As can be seen from Table 4, for the errors under 'a)', the machine was able to correctly classify these tweets while the users wrongly annotate them to different classes. We suppose that such annotation errors by the users occur due to 1) users press the wrong button by mistake, 2) some users might not understand the tweet, or 3) due to slow internet connection, some users reported that there was a delay between the first and the second tweet. At this interval, it is possible to click the button for the second tweet even before the actual tweet is displayed to the user. The tweets under 'b)' are wrong predictions of the model. These tweets are mostly figurative speech, for example, the first tweet in the category contains the phrase **ሆድ ይፍጀው** - with a literal meaning of each word as "abdomen" and "burn" while the correct meaning is "Let it go, I can't say anything". The model seems to consider the

tweets	Annotator	Model prediction
a) Incorrectly labeled tweets that the model correctly classifies		
ባጭሩ ዘሩን የማያቅ ሰው ሁሉ በክልል መታጎሩ ኢፊታዊ ነው። In short, strangling someone who did not know his ethnics origin by region is an injustice.	Neutral	Negative
የሰይፉ ቀልድ አላዋቂነት እና በሁሉም ሰው መቀለድ እንዳለ ሆኖ በሰብአዊነት ለመርዳት ስላደረጋቸው ተግባሮቹ ግን ምስጋና እንጂ እንድህ ማንጓጠጥ አይገባውም:: Besides Seifus' ignorance of humor and the fact that he likes to make fun of everyone, we need to appreciate his humanitarian activities instead of belittling him.	Negative	Mixed
የት ነው ሂሳብ የተማረው ደደቢት Where did he study math, Dedebit	Neutral	Mixed
b) Wrong model predictions		
ህምም በሱ፡ ዘመን ያየነውን ሆድ ይፍጀው። hm. Lets not mention the misery we faced during his time	Negative	Positive
ድሮም ጠርጥሬ ነበር ቀልቃላ ቢጤ ነች። I already suspect, she is quasi impetuous.	Negative	Neutral
ምን ባለሙያ አላችሁ ብለህ ነው። ዲኩላ You think you had a professional. Antelope	Negative	Neutral

Table 4: Error Analysis: Difference of class labels between the annotators and the model predictions.

literal meaning and classify it as 'neutral'. We found out that sarcasm, figurative speech, mixed scripts, incomplete phrases and sentences, and spelling and grammar errors in social media texts are the main challenges for Amharic sentiment analysis (answering our research question 3 - **RQ3**).

6 Conclusion and Future Directions

In this paper, we have presented the first work of sentiment analysis for the Amharic language based on the Twitter dataset. The source dataset is collected using the Twitter API for two months, targeting only tweets written in the 'Fidel' or 'Ethiopic' script. Non-Amharic texts in languages such as Geez and Tigryinga are removed. Using extended sentiment lexicon, a total of 9.4k tweets are processed for sentiment class annotation. As there is no well-established annotation framework to conduct an annotation task for the Amharic sentiment classification, we developed a mobile and social network-based annotation tool called ASAB using the Telegram bot chatbot framework. ASAB incorporated the following: 1) Support of parallel annotation for a large number of users. 2) Integrate controlling mechanisms to block spammers or users with repetitive wrong annotations. 3) Employ an automatic rewarding scheme in the form of mobile card vouchers, which can be extended for various vouchering systems. 4) Allow seamless communication between the users and the annotation task managers in the form of Telegram and email messages. ASAB demonstrated a success story for tackling low-resource language data annotation problems, which resembles the existing crowdsourcing annotation platforms.

We further developed different classical supervised machine learning and deep learning models that are trained on the collected dataset. While the supervised models performed significantly better than the baseline systems, the deep learning models demonstrated superior performance over the classical supervised approaches. The dataset, the extended sentiment lexicon, the best performing models, and associated source codes are released under a permissive license[8].

In the future, we plan to integrate different rewarding mechanisms to the ASAB tool to increase its usability. ASAB can also be extended for different annotation problems such as named entity recognition, relation extraction, machine translation, and so on. Moreover, we will improve the classification models by employing an ensemble approach using batteries of supervised and deep learning models, to lift automatic sentiment labels to a usable level in applications.

References

Mesfin Abate and Yaregal Assabie. 2014. Development of Amharic Morphological Analyzer Using Memory-Based Learning. In *International Conference on Natural Language Processing, NLP 2014*, pages 1–13, Warsaw, Poland.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In

[8] https://github.com/uhh-lt/ASAB

Proceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649, Santa Fe, NM, USA.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, MN, USA.

Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. Aspect based sentiment analysis in Hindi: resource creation and evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2703–2709, Portorož, Slovenia.

Girma Neshir Alemneh, Andreas Rauber, and Solomon Atnafu. 2019. Dictionary Based Amharic Sentiment Lexicon Generation. In *International Conference on Information and Communication Technology for Development for Africa*, pages 311–326, Bahir Dar, Ethiopia.

N. Anitha, B. Anitha, and S. Pradeepa. 2013. Sentiment classification approaches. *International Journal of Innovations in Engineering and Technology (IJIET)*, 3(1):22–31.

Gilbert Badaro, Ramy Baly, Hazem Hajj, Wassim El-Hajj, Khaled Bashir Shaban, Nizar Habash, Ahmad Al-Sallab, and Ali Hamdi. 2019. A survey of opinion mining in arabic: a comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3):1–52.

Josemar A. Caetano, Hélder S. Lima, Mateus F. Santos, and Humberto T. Marques-Neto. 2018. Using sentiment analysis to define Twitter political users' classes and their homophily during the 2016 American presidential election. *Journal of Internet Services and Applications*, 9(1):1–15.

Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.

Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Machine learning*, 22(11):273–297.

Padraig Cunningham and Sarah Jane Delany. 2020. k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples). eprint: 2004.04523, archivePrefix: arXiv, primaryClass:cs.CL.

Alessia D'Andrea, Fernando Ferri, Patrizia Grifoni, and Tiziana Guzzo. 2015. Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, 125(3):26–33.

Paulo De Souza Bermejo, José Pereira, Daniely Barbosa, and Daniel Oliveira. 2019. The application of the sentiment analysis technique in social media as a tool for social management practices at the governmental level. *Revista de Administracao Publica*, 53:235–251.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN, USA.

Fiona M. Dickinson, Mary McCauley, Barbara Madaj, and Nynke van den Broek. 2019. Using electronic tablets for data collection for healthcare service and maternal health assessments in low resource settings: lessons learnt. *BMC Health Services Research*, 19(1):336.

Ahmed Fadhil and Adolfo Villafiorita. 2017. An adaptive learning with gamification & conversational UIs: The rise of CiboPoliBot. In *Adjunct publication of the 25th conference on user modeling, adaptation and personalization*, pages 408–412, Bratislava, Slovakia.

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Communications of the Association for computing machinery*, 56(4):82–89.

Mark A Finlayson and Tomaž Erjavec. 2017. Overview of Annotation Creation: Processes and Tools. In *Handbook of Linguistic Annotation*, pages 167–191. Springer.

Rama Rohit Reddy Gangula and Radhika Mamidi. 2018. Resource Creation Towards Automated Sentiment Analysis in Telugu (a Low Resource Language) and Integrating Multiple Domain Sources to Enhance Sentiment Prediction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 627–634, Miyazaki, Japan.

Selama Gebremeskel. 2010. Sentiment Mining Model for Opinionated Amharic Texts. unpublished, Online: http://etd.aau.edu.et/handle/123456789/3029.

Andargachew Mekonnen Gezmu, Binyam Ephrem Seyoum, Michael Gasser, and Andreas Nürnberger. 2018. Contemporary Amharic Corpus: Automatically Morpho-Syntactically Tagged Amharic Corpus. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 65–70, Alexandria, Egypt.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation Learning on Graphs: Methods and Applications. *IEEE Data Engineering Bulletin*, pages 1–24.

Eduard Hovy and Julia Lavid. 2010. Towards a 'science'of corpus annotation: a new methodological challenge for corpus linguistics. *International journal of translation*, 22(1):13–36.

Abhik Jana and Pawan Goyal. 2018. Can network embedding of distributional thesaurus be combined with word vectors for better representation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 463–473, New Orleans, LA, USA.

Dan Jurafsky and James H. Martin. 2019. Speech and language processing (3rd ed. draft). Online version: https://web.stanford.edu/~jurafsky/slp3/.

Erick Kauffmann, Jesús Peral, David Gil, Antonio Ferrández, Ricardo Sellers-Rubio, and Higinio Mora. 2019. Managing marketing decision-making with sentiment analysis: An evaluation of the main product features using text data mining. *Sustainability*, 11(15):4235.

Moshe Koppel and Jonathan Schler. 2005. The importance of neutral examples for learning sentiment. In *Workshop on the Analysis of Informal and Formal Information Exchange During Negotiations (FINEXIN)*, pages 100–109, Ottawa, ON, Canada.

Robert J. Lennox, Diogo Veríssimo, William M. Twardek, Colin R. Davis, and Ivan Jarić. 2020. Sentiment analysis as a measure of conservation culture in scientific literature. *Conservation Biology*, 34(2):462–471.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*. eprint: 1907.11692, primaryClass:cs.CL.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Noam Lupu and Kristin Michelitch. 2018. Advances in survey methods for the developing world. *Annual Review of Political Science*, 21:195–214.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2020. Deep Learning Based Text Classification: A Comprehensive Review. eprint: 2004.03705, archivePrefix: arXiv, primaryClass:cs.CL.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, GE, USA.

Pooja Pandey and Sharvari Govilkar. 2015. A framework for sentiment analysis in Hindi using HSWN. *International Journal of Computer Applications*, 119(19).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Wondwossen Philemon and Wondwossen Mulugeta. 2014. A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts. *HiLCoE Journal of Computer Science and Technology*, 2(2):8.

Rudy Prabowo and Mike Thelwall. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143 – 157.

James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. O'Reilly Media, Inc.

Eugen Ruppert, Manuel Kaufmann, Martin Riedl, and Chris Biemann. 2015. JoBimViz: A web-based visualization for graph-based distributional semantic models. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 103–108, Beijing, China.

Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 859–866, Reykjavik, Iceland.

Özge Sevgili, Alexander Panchenko, and Chris Biemann. 2019. Improving neural entity disambiguation with graph embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 315–322, Florence, Italy.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France.

Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavvaf, and Edward A. Fox. 2020. Natural language processing advancements by deep learning: A survey. eprint: 2003.01200, archivePrefix: arXiv, primaryClass:cs.CL.

Karsten Tymann, Matthias Lutz, Patrick Palsbröker, and Carsten Gips. 2019. GerVADER -A German adaptation of the VADER sentiment analysis tool for social media texts. In *"Lernen, Wissen, Daten, Analysen - LWDA2019*, pages 178–189, Berlin, Germany.

Charu Virmani, Anuradha Pillai, and Dimple Juneja. 2017. Extracting Information from Social Network using NLP. *International Journal of Computational Intelligence Research*, 13(4):621–630.

Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. 2013. Perspectives on crowdsourcing annotations for natural language processing. *Language resources and evaluation*, 47(1):9–31.

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. GermEval 2017: Shared Task on Aspect-based Sentiment in Social Media Customer Feedback. In *Proceedings of the GSCL GermEval Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, pages 1–12, Berlin, Germany.

Yoram Wurmser. 2018. Mobile Time Spent 2018. Will Smartphones Remain Ascendant? Technical report, emarketer.com. Online: https://www.emarketer.com/content/mobile-time-spent-2018.

Seid Muhie Yimam, Abinew Ali Ayele, and Chris Biemann. 2019. Analysis of the Ethiopic Twitter Dataset for Abusive Speech in Amharic. In *In Proceedings of International Conference On Language Technologies For All: Enabling Linguistic Diversity And Multilingualism Worldwide (LT4ALL 2019)*, pages 1–5, Paris, France.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13:55–75.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 694–699, New York, NY, USA.

Radim Řehůřek and Petr Sojka. 2011. Gensim – Statistical Semantics in Python. In *EuroScipy 2011*, Paris, France.