Learned Text Representation for Amharic Information Retrieval and Natural Language Processing

Tilahun Yeshambel, Josiane Mothe, Yaregal Assabie

▶ **To cite this version:**

Tilahun Yeshambel, Josiane Mothe, Yaregal Assabie. Learned Text Representation for Amharic Information Retrieval and Natural Language Processing. Information, 2023, 14 (3: Special Issue "Novel Methods and Applications in Natural Language Processing"), pp.1-23, article 195. 10.3390/info14030195 . hal-04229854

HAL Id: hal-04229854
https://hal.science/hal-04229854

Submitted on 5 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

information

MDPI

Article

Learned Text Representation for Amharic Information Retrieval and Natural Language Processing

Tilahun Yeshambel [1,*], Josiane Mothe [2] and Yaregal Assabie [3]

[1] IT Doctorial Program, Addis Ababa University, Addis Ababa P.O. Box 1176, Ethiopia
[2] Componsante INSPE, IRIT, UMR5505 CNRS, Université de Toulouse Jean-Jaurès, 118 Rte de Narbonne, F31400 Toulouse, France
[3] Department of Computer Science, Addis Ababa University, Addis Ababa P.O. Box 1176, Ethiopia
* Correspondence: tilahun.yeshambel@uog.edu.et; Tel.: +251-912990195

Abstract: Over the past few years, word embeddings and bidirectional encoder representations from transformers (BERT) models have brought better solutions to learning text representations for natural language processing (NLP) and other tasks. Many NLP applications rely on pre-trained text representations, leading to the development of a number of neural network language models for various languages. However, this is not the case for Amharic, which is known to be a morphologically complex and under-resourced language. Usable pre-trained models for automatic Amharic text processing are not available. This paper presents an investigation on the essence of learned text representation for information retrieval and NLP tasks using word embeddings and BERT language models. We explored the most commonly used methods for word embeddings, including word2vec, GloVe, and fastText, as well as the BERT model. We investigated the performance of query expansion using word embeddings. We also analyzed the use of a pre-trained Amharic BERT model for masked language modeling, next sentence prediction, and text classification tasks. Amharic ad hoc information retrieval test collections that contain word-based, stem-based, and root-based text representations were used for evaluation. We conducted a detailed empirical analysis on the usability of word embeddings and BERT models on word-based, stem-based, and root-based corpora. Experimental results show that word-based query expansion and language modeling perform better than stem-based and root-based text representations, and fastText outperforms other word embeddings on word-based corpus.

Keywords: word embeddings; BERT; pre-trained Amharic BERT model; query expansion; learning text representation; text classification; fine-tuning

check for updates

Citation: Yeshambel, T.; Mothe, J.; Assabie, Y. Learned Text Representation for Amharic Information Retrieval and Natural Language Processing. *Information* **2023**, *14*, 195. https://doi.org/10.3390/info14030195

Academic Editor: Kostas Stefanidis

Received: 15 January 2023
Revised: 17 February 2023
Accepted: 3 March 2023
Published: 20 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Information retrieval (IR) and NLP play key roles in dealing with text documents written in natural language. The use of NLP to analyze text and audio data is common in many languages. One of the techniques for processing a text is to represent it as a sequence of characters and one-hot encoding. However, this is not the most efficient and effective method for word representation [1]. Rather, context-based approaches represent a word by a real-valued vector without using any linguistic processing. Such a vector is then used in machine learning approaches for different applications. For example, semantic vector representation is used in IR [2], document classification [3], question answering [4], named entity recognition [5], and parsing [6]. The quality of results in such systems depends on the quality of word representations.

Neural models learn informative representations of words automatically from a large text corpus. They generate a vector representation of words, and thus a text can be further processed by algorithms. Neural network is a common approach for learning representations of sub-words, words, phrases, sentences, and documents [7]. A robust

text representation technique can capture the underlying meaning or syntax relationships between related texts or other language properties. Distributed representation is one way to represent words as continuous real-valued vectors [1]. This approach learns linguistic regularities, such as the syntax and semantics of a word in a text. Learning appropriate representations for text, such as words, phrases, sentences, and documents, is essential for IR, sentiment analysis, and other text-based tasks. In recent years, word embeddings and variants of BERT models have brought new solutions to learn better representations for NLP and IR tasks. Pre-trained word embeddings [8], paragraph embeddings [9], and sentence embeddings [10] have become essential elements in modern NLP systems.

Word embedding is a distributed dense representation of a text and refers to a set of language modeling and features of learning techniques in NLP. Recently, word embedding has gained more attention and has been successfully employed in various NLP and IR tasks [11,12]. It is considered to be one of the most significant breakthroughs of deep learning for solving challenges in NLP. It represents words in the form of real-valued vectors that make similar words to have similar vector representations [8]. Prominent word embedding methods include word2vec, global vector (GloVe), and fastText. These models learn semantic representation of a language based on the distributed hypothesis. As a result, they can capture the semantic relationships between words. Word2vec learns word representation based on a shallow neural network. It represents words as vectors by making use of self-supervised learning. The main idea behind word2vec is to train a model on the context of each word [8]. On the other hand, GloVe builds a model based on the advantage of global matrix factorization and local context windows [13]. Global matrix factorization minimizes a huge term frequency matrix by using the matrix factorization method from linear algebra. The global matrix indicates if words exist in a text. GloVe is trained using an accumulated global word–word co-occurrence matrix [13]. Unlike word2vec and GloVe, fastText is a sub-word embedding model. FastText learns more information about morphological details of words. It represents each word as an n-gram of characters rather than learning vectors for words directly. It uses sub-word information and the internal structure of a word for improving the quality of word embeddings [14]. The main advantage of using fastText is that it generates better word embedding for rare and unseen words during training. The n-gram character vectors generated by fastText are shared with other words. Word2vec and fastText are based on continuous bag-of-words (CBOW) and skip-gram architectures. The CBOW predicts a target word given its context, whereas the skip-gram predicts the context given a word. The major limitation of these standard language models is that words having multiple meanings in different contexts are represented by a single vector. BERT addresses this problem by incorporating context information from both directions [15]. It is transformer-based model that brought substantial advancement in NLP and obtained state-of-the-art results on a wide array of NLP tasks. Masked language modeling (MLM) and next sentence prediction (NSP) are the two most important pre-training tasks in BERT and its variants. The aim of MLM is to predict masked tokens from the input based on its context, and NSP determines if two sentences are consecutive. Since pre-trained language models have impacts on NLP downstream tasks, many BERT models have been released for various languages, such as for Arabic [16], Dutch [17], Italian [18], Portuguese [19], Russian [20], and French [21]. They can successfully improve many downstream NLP tasks [22]. However, such pre-trained models have not been yet built and made publicly accessible for the Amharic language.

Amharic is one of the Semitic languages spoken in Ethiopia. Research on Amharic computational linguistics and application development lags behind as a result of various challenges. One of the challenges is the morphological complexity of the language, which limits the progress on research and development of Amharic NLP tools, resources, and applications. Another challenge is unavailability of usable and publicly accessible linguistic resources, such as pre-trained language models and scientifically built huge text document collections. Some researchers conducted research to build Amharic pre-trained models. However, many of previously developed Amharic pre-trained models are inaccessible

publicly, and thus obtaining them is challenging for researchers [23]. Thus, our objective is to build a series of Amharic pre-trained language models and make them publicly available. This helps to build baselines for future studies in the area of Amharic IR and NLP. It also minimizes the repetitive training processes required for downstream NLP tasks. Accordingly, to investigate Amharic learned text representations and address challenges related to lack of Amharic-trained machine learning models, we have performed the following activities in this study: (i) We trained Amharic word embeddings, BERT WordPiece tokenizer, and BERT models, and we explore their effects on surface words, stems, and roots. (ii) We designed an Amharic IR system considering the morphology of the language and investigated the usability of the trained word embeddings on Amharic IR, following the TREC ad hoc retrieval experimental framework. (iii) We verified the effect of BERT models on Amharic language modeling and on some NLP tasks. For these, our pre-trained BERT models were fine-tuned using our testing datasets for text classification based on subject and relevance classifications of documents for some queries.

The rest of the paper is organized as follows. Section 2 presents related work on word embedding and BERT pre-trained models. Section 3 discusses the complexity of Amharic morphology, learning Amharic text representation, and its usability for Amharic IR and NLP. The construction of pre-trained Amharic word embedding and BERT models, query expansion using word embedding, and fine-tuning of Amharic pre-trained models for text classification tasks are presented in this section. Section 4 presents an experimental framework. Section 5 reports the results and discusses the findings, along with the analysis of the usability of word embeddings for query expansion and the effects of fine-tuning BERT models on Amharic text classification. Section 6 presents conclusions and suggestions for further work.

2. Related Work

Research on word embedding and pre-trained BERT models has been conducted for various languages. Word2vec models are employed in IR systems in order to improve retrieval results using query expansion. Semantically similar words to query terms can be selected from word2vec models and can be added to the original query in order to identify relevant documents accurately. For instance, if a user query contains the term "example", it can be expanded to "sample" and "instance". Searching using these three terms can be more effective than searching using only one of them. This approach has been used in [24–26], and promising results were obtained. For example, Diaz et al. [24] studied the use of word embeddings for expanding a user query in ad hoc IR. Word2vec and GloVe were employed to represent terms. The models were trained using documents from TREC ad hoc retrieval, robust, and Web collections, and then the effects of local and global embeddings were investigated. The numbers of documents in the TREC ad hoc retrieval collection, robust, and Web datasets are 469,949, 528,155, and 50,220,423, respectively. Each query term was expanded using the top 10 terms from word2vec and GloVe embeddings. Then, Indri retrieval tool was used for indexing and retrieval. They found that local embeddings outperformed global embeddings. Embedding-based query expansion outperformed classical based retrieval (i.e., without query expansion). In the case of global embedding (GloVe), the best performances achieved on TREC, robust, and Web collections were 0.545, 0.472, and 0.232, respectively. Whereas, in the case of local embedding (word2vec), the best performance obtained on TREC, robust, and Web collections were 0.563, 0.517, and 0.258. Aklouche et al. [25] designed query expansion methods based on NLP tools and word2vec embedding models. Both skip-gram and CBOW models were trained with a window size of 5, and the vectors dimension was 300. Words with a term frequency less than 5 were removed from the models. An experiment was conducted on TREC Washington Post Corpus. Semantically related words to TREC title words were selected using the word2vec model and were used to retrieve a set of top ranked 10,000 documents for each topic. Porter stemmer was used to process documents and query terms before query expansion. The retrieval result indicates that 80% of the topics are above the TREC median scores.

Sentiment analysis is one possible application area for word embeddings. Deho et al. [27] developed a sentiment analysis classifier using word embeddings. Word vectors were generated by the skip-gram model and used to train the sentiment classification model. The random forest classifier was trained on tweet datasets and used to predict the sentiment polarities of the test dataset. The classifier achieved an overall accuracy of 81%. Acosta et al. [28] conducted an experiment to classify sentiment using the word2vec model. Various experiments were conducted on a publicly available dataset which contains 14,640 tweets. The highest accuracy obtained for sentiment classification of Twitter posts was 72%. They used both skip-gram and CBOW algorithms to generate word vectors. Gaussian naïve Bayes, Bernoulli naïve Bayes, support vector machines, and logistic regression were trained with both models of word2vec. The finding indicates that the performance of the support vector machine and logistic regression using the skip-gram model outperforms the naïve Bayes classifiers.

Question answering is another application area of word2vec. A word2vec model can be used to select a sentence from a text that has the highest word2vec embedding similarity with a given question, and can return it as the answer. Medved and Horák [29] presented automatic question answering system using word2vec and Doc2vec. While word2vec represents each word by a separate vector, the Doc2vec module was used to generate a vector for each sentence. An experiment was conducted on 3000 question-answer pairs extracted from the Czech Wikipedia. The highest performance was obtained by the combination of several similarity criteria, including term frequency–inverse document frequency and tree distance. Experimental results on the SQAD v1.0 and SQAD v1.1 datasets showed that the syntax-based approach using word2vec phrasal combinations outperformed the general Doc2vec model by 11.6%.

Devlin et al. [15] introduced the BERT model, which can be fine-tuned to create models for a range of NLP tasks. Sun et al. [30] proposed to use BERT NSP for prompt-learning tasks. A wide range of NLP tasks in the zero-shot scenario were performed using NSP-BERT and RoBERTa. The authors proposed a two-stage prompt method for word sense disambiguation tasks. Experiments were carried out on the Chinese benchmark FewCLUE dataset. The performance of NSP-BERT for word sense disambiguation was evaluated, and test results show that NSP-BERT has an improvement over generative pre-trained transformer (GPT) and pattern exploiting training (PET). The best accuracy NSP-BERT obtained was 69.7%. Shi and Demberg [31] also trained and used the NSP model for discourse relation classification. NSP was used to capture what events are expected to cause or follow each other based on semantic information. The result shows that the BERT-based NSP model [31] outperforms the current state-of-the-art [32] by 8% on Penn Discourse Tree Bank [33], which contains over 1 million words. Cui et al. [34] introduced a whole word masking strategy and MacBERT for Chinese BERT. They created a series of Chinese pre-trained language models such as BERT, RoBERTa, ELECTRA, and RBT. MacBERT is the modification of the MLM as a language correction task and solves the inconsistency of the pre-training and fine-tuning stages. A broad range of empirical studies were carried out to evaluate the performance of pre-trained models on various tasks. The number of words to train BERT, RoBERTa, ELECTRA, and RBT models were 0.4B, 5.4B, 5.4B, 5.4B, 5.4B, and 5.4B, respectively. Experimental results indicated that the proposed MacBERT method achieved better performance for many tasks. The best F1-score of the MacBERT-base and MacBERT-large case models were 93.8% and 95.5%, respectively. The whole word masking also addresses the problem of masking only a part of the whole word.

Aggarwal et al. [35] fine-tuned the BERT model for a fake news binary classification task. The proposed system achieved an accuracy of 97.02% on the NewsFN dataset that has 6335 items. It was reported that BERT performed better than XGBoost and LSTM approaches. Protasha et al. [36] examined the effect of word embeddings and fine-tuned BERT for sentiment classification. The authors integrated CNN-BiLSTM with BERT to address the problem of sentiment analysis for the Bangla language. They also compared word2vec, GloVe, fastText, and BERT. The experiment was conducted on 8952 samples from various sources. From these samples, 4325 samples were positive and the rest of the

samples were negative. The performance of word2vec and GloVe with LSTM classification is almost similar. However, fastText improves the performance by 4%, with a score of 88.35%, using the impact learning method. The combination of fine-tuned Bangla-BERT and LSTM leads to an accuracy of 94.15%.

Pre-trained language models have been generated for many languages. Although Amharic is the most widely spoken language in Ethiopia and a large volume of text is available on the Web, it is lagging behind in computational analysis including BERT and word embeddings. Some research has been conducted to create pre-trained Amharic models. To mention them: fastText [37,38], word2vec [26,38–40], and XLMR [41]. Some of these models were trained for cross-lingual purposes and are not usable for the needs of most NLP tasks. Moreover, most of them are not publicly accessible. Because of this, Amharic NLP tasks have been performed using classical text representations such as stems and roots [42,43], and the impact of learned text representations on roots, stems, and words to the development of various applications is not yet investigated. Thus, the construction of pre-trained Amharic models is a long sought resource for the research community. In view of this, the major contributions of this work are: (i) construction of pre-trained Amharic models and publicly sharing them to the research community; (ii) fine-tuning the pre-trained models for NLP and IR tasks; and (iii) investigation of the effects of roots, stems, and surface words on learned text representations.

3. Construction of Learned Text Representations for Amharic

3.1. Amharic Language

Amharic is the most widely spoken language in Ethiopia. It has its own alphabet and phonological and morphological features. Words are represented by phonemes/characters that are pronounced. Amharic has both constant and vowel characters where the base forms of consonants are modified by following the vowels.

Amharic has complex morphological features that possess complicated syntactic features. Many words can be generated from a base form through inflectional and derivational word formation processes performed using prefixes, suffixes, infixes, and circumfixes. For example, Amharic nouns and adjectives are marked for number, definiteness, gender, and case. Moreover, they are affixed with prepositions and are genitive. Amharic verbs have even more complex inflectional and derivational process than other word classes, and they comprise the majority of words in the language [44]. Verbal roots are the base of many verbal stems which, in turn, are the base of many verb surface forms. In Amharic, a verbal stem is not necessarily identical with the morphological root of the word. Morphology provides the templates for the combination of the root consonants with the theme vowels to derive basic stems, which may be actual or potential verbs [45]. For example, the stem ስበር /sɨbərɨ 'infinitive'/ is derived from the root ስ-ብ-ር /*s-b-r* 'break'/ using a ስ-ኧ-ብ-ኣ-ር pattern. The radicals are ስ /*s*/, ብ /*b*/ and ር /*r*/, whereas ኧ /*ə*/ and ኣ /*a*/ are vowels. The vowels change the shape of the immediately preceding radical in a given root during a stem or word formation. Morpho-syntactical properties of verbs are based on the arrangement of the consonants and vowels. A verbal root form uses different templates to derive different morpho-syntactic categories such as perfective, imperfective, imperative, jussive, gerundive, and verbal words. For example, the surface form ሰበረ /səbərə 'he broke'/ consists of the template form (CVCVCV), root consonants (ስ-ብ-ር /*s-b-r* 'break'/), and vowel pattern (ኧ /*ə*/). While the root ስ-ብ-ር /*s-b-r*/determines the basic meaning of the word, the template CVCVCV, along with the vowel pattern, provides additional morpho-syntactic information such as aspect and tense. Many affixes, such as person, gender, number, case, tense/aspect, subject, object, and mood markers, can be attached on verbal stems to form thousands of variants.

3.2. Pre-Training Process

In our study, documents from the training corpora pass through a preprocessing step. The preprocessing consists of removal of tags, punctuation markers using a punctuation

list, function words, and non-Amharic characters. We also normalize Amharic characters that have similar phonetic representations. The characters ሐ /hə/, ኀ /hə/, ኸ /hə/, and their modifications are normalized to their corresponding modifications of ሀ /hə/. The character ሠ /sə/ and its modifications are normalized to their corresponding modifications of ሰ /sə/. The character ፀ /ts'ə/ and its modifications are normalized to their corresponding modifications of ጸ /ts'ə/. The character ዐ /ʔə/ and its modifications are normalized to their corresponding modifications of አ /ʔə/. The fourth orders ሃ /ha/, ሓ /ha/, ኃ /ha/, and ኻ /ha/ are normalized to ሀ /hə/, whereas the fourth orders ኣ /ʔa/ and ዓ /ʔa/ are normalized to አ /ʔə/. This is followed by sentence segmentation using "።" and tokenization into fundamental units (i.e., words, tokens, or n-grams) using space. Since pre-trained Amharic BERT tokenizers are unavailable publicly, we built WordPiece tokenizer in order to encode inputs for the MLM, NSP, and fine-tuned models. Previous studies in Amharic IR have shown the importance of word-based, stem-based, and root-based corpora for raw text representation [40]. Here, we use the Amharic ad hoc information retrieval test collection (2AIRTC) corpus that also comes with the corresponding stem and root forms for learned text representation. The word-based, stem-based, and root-based documents were tokenized using the corresponding word-based, stem-based, and root-based WordPiece vocabularies. The tokenizers we used may segment words into multiple sub-words in order to more efficiently deal with out-of-vocabulary words and for better representation of complex words. For example, the sentence "የሰሜን ተራሮች ብሄራዊ ፓርክ እርዳታ እየጠየቀ ነው" and its corresponding stems and roots are tokenized into ['የሰ', '##ሜን', 'ተራ', '##ሮች', 'ብሄ', '##ራዊ', 'ፓርክ', 'እር', '##ዳ', "##ታ', 'እየ', '##ጠየ', '##ቀ','ነው'], ['ሰሜን', 'ተራራ', 'ብሄር', 'ፓርክ', 'ርድ', 'ጠየቅ'], and ['ሰሜን', 'ተራራ', 'ብሄር', 'ፓርክ', 'ር-ድ', 'ጥይ', '##ቅ'] using the word, stem, and root vocabularies, respectively. After tokenization, each token in the input sequence is converted into token IDs corresponding to the index in BERT's vocabulary.

To compare the effect of learned text representations on Amharic words, stems, and roots, we compute different versions of word-based, stem-based, and root-based pre-trained Amharic models using the corresponding corpus. Figure 1 shows samples of preprocessed word-based, and its corresponding stem-based and root-based, documents. The distribution of words in each document for the entire corpus is illustrated in Figure 2.

Figure 1. Representation of a document using word-based (**top**), stem-based (**middle**), and root-based (**bottom**) items.

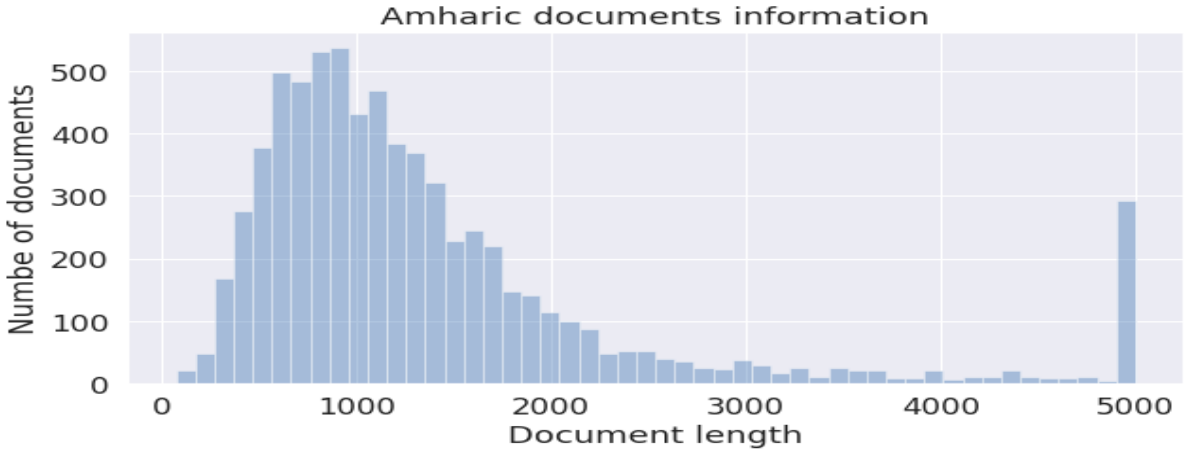

Figure 2. Distribution of words in the corpus.

3.3. Construction of Amharic Pre-Training Models

3.3.1. Amharic Pre-Trained Word Embeddings

We built three types of word embedding models, namely word2vec, GloVe, and fastText. The number of hidden layers between the input layer and the output layer is 1, as the word embeddings learn only one vector representation for each word. These embedding models learn each word representation considering the window size. For example, given the sentence "ጽሁፍ የተጻፈው የህክምና እርዳታ ለመስጠት አይደለም */ts'ihufi jətəts'afəwi jəhikimina ʔiridata ləməsit'əti ʔəjidələmi/*" along with its corresponding stem and root representations and the window size of 3, the word2vec model learns the representation of each word using the pair of input and output training datasets as presented in Table 1. The size of the window in the case of word2vec, GloVe, and fastText is presented in Table 2.

Table 1. Word-based, stem-based, and root-based Skip-gram training samples of word2vec. The first column contains the target/input in surface word form before morphological analysis, whereas the second column contains pairs of target and context/output words for word-based, stem-based, and root-based approaches of the given word.

Word	Training Sample	Type
ጽሁፍ	(ጽሁፍ, የተጻፈው), (ጽሁፍ, የህክምና), (ጽሁፍ, እርዳታ) (ጽሁፍ, ጻፍ), (ጽሁፍ, ህክም), (ጽሁፍ, ርድ) (ጽ-ህ-ፍ, ጽ-ህ-ፍ), (ጽ-ህ-ፍ, ህ-ክ-ም), (ጽ-ህ-ፍ, ር-ድ)	Word Stem Root
የተጻፈው	(የተጻፈው, ጽሁፍ), (የተጻፈው, የህክምና), (የተጻፈው, እርዳታ), (የተጻፈው, ለመስጠት) (ጻፍ, ጽሁፍ), (ጻፍ, ህክም), (ጻፍ, ርድ), (ጻፍ, ስጥ) (ጽ-ህ-ፍ, ጽ-ህ-ፍ), (ጽ-ህ-ፍ, ህ-ክ-ም), (ጽ-ህ-ፍ, ር-ድ), (ጽ-ህ-ፍ, ስ-ጥ)	Word Stem Root
የህክምና	(የህክምና, ጽሁፍ), (የህክምና, የተጻፈው), (የህክምና, እርዳታ), (የህክምና, ለመስጠት), (የህክምና, አይደለም) (ህክም, ጽሁፍ), (ህክም, ጻፍ), (ህክም, ርድ), (ህክም, ስጥ) (ህ-ክ-ም, ጽ-ህ-ፍ), (ህ-ክ-ም, ጽ-ህ-ፍ), (ህ-ክ-ም, ር-ድ), (ህ-ክ-ም, ስ-ጥ)	Word Stem Root
እርዳታ	(እርዳታ, ጽሁፍ), (እርዳታ, የተጻፈው), (እርዳታ,የህክምና), (እርዳታ,ለመስጠት), (እርዳታ, አይደለም) (ርድ, ጽሁፍ), (ርድ, ጻፍ), (ርድ, ህክም), (ርድ, ስጥ) (ር-ድ, ጽ-ህ-ፍ), (ር-ድ, ጽ-ህ-ፍ), (ር-ድ, ህ-ክ-ም), (ር-ድ, ስ-ጥ)	Word Stem Root
ለመስጠት	(ለመስጠት, የተጻፈው), (ለመስጠት, የህክምና), (ለመስጠት, እርዳታ), (ለመስጠት, አይደለም) (ስጥ, ጻፍ), (ስጥ, ህክም), (ስጥ, ርድ) (ስ-ጥ, ጽ-ህ-ፍ), (ስ-ጥ, ህ-ክ-ም), (ስ-ጥ, ር-ድ)	Word Stem Root

Table 2. Hyper-parameter setting for the different embedding models.

Model	Vector Size	Window Size	Epoch	Min_Count	Learning Rate
word2vec	300	5	20	3	0.05
GloVe	100	10	30	5	0.05
fastText	100	5	30	5	0.05

However, in the case of fastText, for a given word, n-grams are taken from 3 to 5 g and a word is represented by the sum of the vector representations of its n-grams. Each Amharic word is represented as a bag of sub-words and the word itself. For example, consider the Amharic word ለምስክርነት */ləmisikirinət'i* 'for witness'/ and the 3 g characters. Its sub-word representations in the case of word-based, stem-based, and root-based are presented as follows:

Using such self-supervised datasets, each vocabulary entry is represented by a real-valued vector which is learned from the corpora. The vector dimension of a word (or token) is different in the case of word2vec, GloVe, and fastText models.

Word-based: <ለም, ለምስ, ምስክ, ስክር, ክርነ, ርነት, ነት>, and ለምስክርነት
Stem-based: <ምስ, ምስክ, ስክር, ክር>, and ምስክር
Root-based: <ም-, ም-ስ,-ስ-, ስ-ክ, -ክ-, ክ-ር, -ር>, and ም-ስ-ክ-ር

3.3.2. Amharic Pre-Trained BERT Language Models

On top of word embeddings, we build pre-trained BERT models for masked language modeling and next sentence prediction. BERT models learn each token representation considering the context of a word in a sentence, and thus a word may have more than one vector representation [15]. The MLM and NSP BERT models have millions of parameters.

The MLM is BERT's pre-trained language model to predict a masked token. We train the Amharic MLM model by masking one or more words in a sentence using the embedding of a special symbol [MASK]. In order to train the Amharic pre-trained MLM model, 15% of the input tokens are randomly masked with [MASK], resulting in the masked input sequence. BERT special tokens are excluded from masking, and then the MLM model learns to predict those masked tokens back. For example, inputting the sentence መንግስት መግለጫ ሰጠ */mənigisiti məgiləʧ'a sət'ə* 'the government provides press release'/ and its corresponding stem-based and root-based representations by masking the word መግለጫ */məgiləʧ'a/* or its respective stem and root to the model, the MLM model predicts the masked word with the highest probability. If the input sentence is in surface form, the model predicts መግለጫ with the highest probability (see Figure 3). For its stem and root-based sentences, the model predicts the words ግለጥ */gilət'i/* and ግ-ል-ጥ */g-l-t'/*, respectively, with the highest probability.

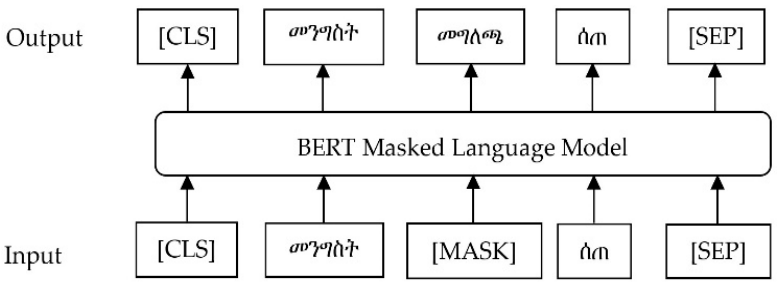

Figure 3. An example of masked word prediction.

During MLM training, there is a loss between the predicted probability and expected token for the masked word. The overall loss L of the input sequence is the average loss for all the k masked-out tokens in *xmasked*, and it is computed using Equation (1).

$$LMLM(\theta) = \frac{1}{k} \sum_{iem} - \log p(x_i|xmasked) \quad (1)$$

where x_i is the set of masked tokens in sentence x, k is number of masked tokens, and xmsked is masked token.

We trained the Amharic NSP model to learn the relationships between any sentence pairs. Given two input sentences A and B, the model is trained to predict whether sentence B is the next sentence after A. A total of 50% of the time, B is the next sentence after A (IsNext), and, for the other 50%, we used a random sentence from the corpus (NotNext). During the NSP model training, the loss value *LNSP* is computed using the loss function presented in Equation (2).

$$LNSP = -\log qM(n \mid x) \quad (2)$$

where n ∈ {IsNext, NotNext} and x is input sequence.

$$\mathrm{qM}(\mathrm{n_k}|\mathrm{x_i}) = \frac{\exp s(n_k|x_i{}^{(1)}, \ x_i{}^{(2)})}{\sum_n \exp s\left(n|x_i{}^{(1)}, x_i{}^{(2)}\right)} \quad (3)$$

where $x_i{}^{(1)}$ and $x_i{}^{(2)}$ denote sentence A and sentence B, respectively. s = Wnsp(tanh (Wh[CLS] + b))^2, where wh[CLS] is the hidden vector of [CLS] and *Wnsp* is a matrix learned by the NSP task.

Since our vocabularies were larger than the default vocabulary size of BERT-base-uncased models (i.e., 30,522), we resized the default BERT vocabulary size to the size of our word, stem, and root vocabularies. We pre-trained Amharic MLM and NSP with unlabeled large corpora, considering full length sentences. The resulting models understand the structure of the language but are not trained for a specific task. They can be further fine-tuned on various downstream tasks.

3.4. Usability of Pre-Trained Amharic Word Embeddings and BERT Models

The usability of our pre-trained models was investigated in this study. The pre-trained Amharic models can be used in many applications. Here, our aim was to explore their applications in Amharic IR and text classification tasks. The details of using the pre-trained models for these two tasks are presented below.

3.4.1. Pre-Trained Amharic Models for IR

We investigated the effect of learned text representation on Amharic IR. IR is the task of searching relevant documents to a user query. Various linguistic tools with good accuracy are integrated into IR systems to enhance the retrieval effectiveness. One of the main challenges in IR is term mismatch between query and document terms. Query expansion is a technique to address this problem by adding relevant terms to the original query. We evaluate the effects of learned real-valued text representations on Amharic IR. The impact of word embeddings on Amharic IR are investigated by integrating the created token vector space in the Amharic IR system from [46]. The system architecture is modified by inserting query expansion using our trained word/token embedding models (See Figure 4).

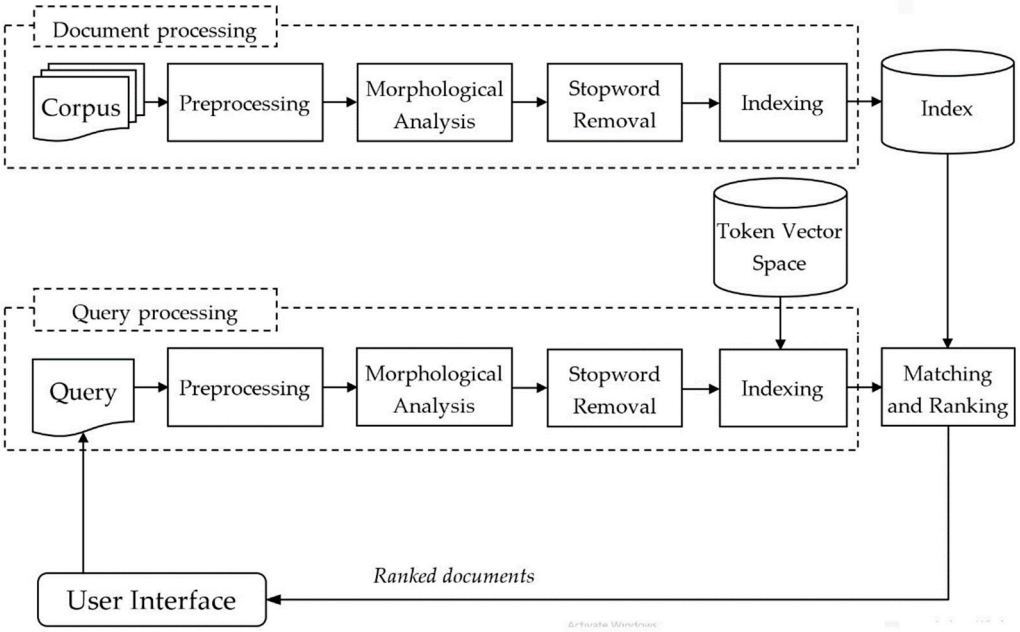

Figure 4. Query expansion architecture using word embeddings for Amharic IR.

As shown in Figure 4, documents in the corpus and a user query were first processed, considering stem-based and root-based stopwords lists [46]. For query expansion, we used the pre-trained Amharic word embeddings word2vec, GloVe, and fastText models as described in Section 3.3. The token vector space in Figure 4 refers to these word-embedding models. The word embeddings hold learned representations of each word, stem, and root in the corpora. The three models are used to find semantically related words/terms to each user's query term. The similarity between a query term (or word) x and each term y in the models is computed using the cosine similarity metrics presented in Equation (4).

$$\cos(\mathrm{x}, \mathrm{y}) = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} {x_i}^2 \sum_{i=1}^{n} {y_i}^2} \tag{4}$$

where x_i and y_i are vector values of two words, and n is size of the vectors.

Similar to the works of Diaz et al. [24] and Arora et al. [47], we found that the top 10 returned words were more similar to each query term. Therefore, for each query term (or word), 10 expanded terms are selected for retrieval. Some expanded terms occur more than once, and we remove redundancies from the last query term list. The last query set contains a list of the original query terms and expanded terms.

Searching for relevant documents is based on matching the original query terms and expanded terms with index terms. We used exact term matching. The retrieval probability of a relevant document for a query is different in the case of word-based, stem-based, and root-based retrieval. We used the Lemur toolkit for ranking (http://www.lemurproject.org accessed on 7 August 2022). For a given query Q and a collection of retrieved documents D, the Lemur toolkit ranks retrieval results based on their possible relevance. Language modeling is used for matching and ranking of retrieved documents. The similarity between a document D and a query Q is measured by the Kullback–Leibler (KL) divergence between the document model $D\theta$ and the query model $Q\theta$. The KL divergence ranking function captures the term occurrence distribution and is computed using Equation (5).

$$KL(Q\theta, D\theta) = \sum_{w \in V} p(w|Q\theta) \log \frac{p(w|Q\theta)}{p(w|D\theta)} \tag{5}$$

where w is word, v is word vector, $p(w \mid Q\theta)$ is estimated query term, and $p(w \mid D\theta)$ is the smoothed probability of a term seen in the document.

The retrieval effectiveness of the proposed system was evaluated using precision, recall, mean average precision (MAP), and normalized discounted cumulative gain (NDCG).

3.4.2. Pre-Trained Amharic Models for Text Classification

We also investigated the usability of our pre-trained BERT models for text classification. The common practice for using a pre-trained BERT model is to update (or retrain) the original output layer with a task-specific layer and fine-tune the whole model. Fine-tuning is the task of creating application on top of pre-trained models using labeled data. Fine-tuning BERT is one of the most popular and effective methods to tackle NLP tasks [48].

Here, we fine-tuned our pre-trained Amharic BERT models for text documents classification tasks. The pre-trained Amharic models were fine-tuned using manually labeled training datasets. The training datasets passed through the same data processing used for pre-training models. Furthermore, the same tokenizer and vocabulary were used. The pre-trained Amharic MLM model was retrained and evaluated using these datasets.

For the classification task, we added a classification head on top of the pre-trained Amharic BERT model and then trained the entire model on our training datasets. All parameters were fine-tuned from end-to-end. The softmax function was used to estimate the probability of the label c at output layer o using Equation (6). We fine-tuned all of the parameters from BERT and W.

$$P(c \mid o) = softmax(W) \tag{6}$$

where W is the task-specific parameter matrix.

Text classification is one of the core tasks of NLP, and it is commonly considered to show the effectiveness of a pre-trained BERT model. Our pre-trained Amharic models were fine-tuned for Amharic document classification based on subject and relevance to a query. For document classification based on subject, the documents in our datasets were classified as technology, politics, religion, sport, justice, entertainment, social, agriculture, economy, education, and health. Hence, the number of labels was 11. Moreover, we applied the pre-trained Amharic BERT models to check the relevance of a document for a query. Documents were classified as relevant or irrelevant to a query, and thus it was a binary text document classification. The fine-tuned relevance classification model was trained by inputting a pair of texts: query and document. Since the input of a BERT is a sequence, the two inputs are separated by a special token [SEP]. We transformed a query sQ and a document sD into the format compatible with BERT sequence-pair classification tasks as:

[CLS] sQ [SEP] sD [SEP].

where [CLS] and [SEP] are classification and separation tokens, respectively. Since the maximum input sequence length supported by the BERT model is 512 [48], the maximum sequence length of each of our training datasets was 512. The classifier was trained with both relevant and non-relevant documents for each query.

4. Experimental Framework

In this section, we illustrate how various experiments and existing pre-trained English language models are adapted for Amharic. Furthermore, the experimental results and detailed discussions are presented.

4.1. Implementation

We used Python for preprocessing texts and implementing the models. The pre-trained and fine-tuned models were trained on Google Colab's GPU and Kaggle's GPU. After training the pre-trained word embedding models, each query term was expanded using the models. Then, the Lemur toolkit was used to create word-based, stem-based, and root-based indexes of documents offline and run the corresponding word-based, stem-based, and root-based expanded terms with original query terms on the index files. Finally,

the retrieval effectiveness of our proposed Amharic IR system using our pre-trained models was evaluated using the trec_eval tool [49]. The word embedding models word2vec, GloVe, and fastText were trained using Python Gensim, GloVe, and fastText libraries. Furthermore, we used a transformers library for developing and training BERT models. Transformers provide a set of pre-trained deep learning models for a broad range of NLP tasks, such as text classification, question answering, machine translation, etc. We trained the MLM, NSP, and classification BERT models using BertForMaskedLM, BertForNextSentencePrediction, and BertForSequenceClassfication, respectively, using Hugging Face libraries. Hugging Face provides a means for using pre-trained BERT models specialized for various NLP tasks.

4.2. Datasets

To train the word embeddings (word2vec, GloVe and fastText), BERT models (NSP and MLM), and fine-tuned models, we used three datasets: word-based, prepared in 2AIRTC [50], stem-based, and root-based corpora built by [51]. Each of these datasets has 6069 documents and 240 queries. All documents were used to train the pre-trained word embedding and BERT models. The documents consisted of 72,814 sentences constructed from 1,592,351 words. Furthermore, the retrieval effectiveness of Amharic IR system after query expansion using word embedding models was tested using these test collections. The test collections had corpus, topic set, and relevance judgment. The datasets are publicly available freely at https://www.irit.fr/AmharicResources/, accessed on 7 August 2022. For relevance and non-relevance classification, we used only 50 word-based queries and 5514 documents, as well as their relevance judgment from the 2AIRTC collection. We prepared 1189 labeled word-based text documents for fine-tuning the BERT model for document classification based on subject. These documents were collected from various sources, including news agencies, the web, and Amharic Wikipedia.

4.3. BERT Vocabulary

We built three WordPiece vocabularies on the three corpora: word-based, stem-based, and root-based vocabularies, using the BertWordPieceTokenizer model. The vocabulary sizes of word-based, stem-based, and root-based vocabularies were 252,605, 49,817, and 46,995, respectively. The three vocabularies were created after character normalization and were used to tokenize words in the corresponding documents.

4.4. Pre-trained Models and Training Configurations

The word2vec, GloVe, and fastText models were trained using the configurations presented in Table 2.

The MLM model was trained to predict randomly masked tokens, whereas the NSP model learned to predict whether sentence B is the next sentence of A, given a pair of input sentences A and B. We used the standard BERT model architecture. The two BERT models' trainings were performed using sentences as input. The maximum number of tokens of each input sentence for each model was 512. The two models were trained using the same hyper-parameters over the three datasets using epochs 5 and 10 (see Table 3). We used Adam optimizer [52] with a learning rate of 10^{-5} to optimize the objective functions of MLM, NSP, and classification models.

Table 3. Hyper-parameter settings for NSP and MLM models.

Model	Case	Layer Size	Batch Size	Max Length	Hidden	Attention	Total Parameters	Learning Rate
NSP	base	12	16	512	768	12	110 M	10^{-5}
MLM	base	12	16	512	768	12	110 M	10^{-5}

4.5. Evaluation Metrics

The performance of the constructed Amharic BERT models was evaluated based on loss, F-score, and accuracy metrics. The performance of the NSP-BERT model was evaluated using 50 similar pairs of sentences but in word-based, stem-based, and root-based forms. These 50 pairs of sentences were selected randomly from the three datasets; 25 pairs were adjacent and the remaining 25 pairs of sentences were non-adjacent sentences. The accuracy of the NSP model is the percentage of corrected predicted pairs of sentences over the total sentence pairs, and it is computed using Equation (7).

$$\text{Accuracy} = \frac{\text{TP} + \text{FP}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{7}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

The retrieval effectiveness of our proposed IR system using the pre-trained word embedding models was evaluated using precision, recall, MAP, bpref, and normalized discounted cumulative gain (NDCG).

Precision: Precision is defined as the fraction of relevant documents in ranked list *R*, and it is computed using Equation (8).

$$\text{Precision}(\text{R}, \text{q}) = \frac{\sum_{(i,d)\epsilon R} rel(q,d)}{|R|} \tag{8}$$

where *rel(q,d)* indicates if the document *d* is relevant to the query *q*. In the case of binary relevance, *rel*(*q*,*d*) = 1 for relevant documents, and 0 otherwise. Often, precision is evaluated at a cutoff *k*, denoted as Precision@k. If the cutoff is defined in terms of the number of relevant documents for a particular topic (i.e., a topic-specific cutoff), the metric is known as R-precision [53].

Recall: Recall is the set of fractions of relevant documents from the entire collection *C* for the query *q* that are retrieved in ranked list *R*. It is computed using Equation (9).

$$\text{Recall}(\text{R},\ \text{q}) = \frac{\sum_{(\text{i,d})\epsilon \text{R}}\ \text{rel}(\text{q},\text{d})}{\sum_{\text{d}\epsilon \text{C}}\ \text{rel}(\text{q},\text{d})} \tag{9}$$

where *rel(q,d)* indicates whether document *d* is relevant to query *q*.

Mean Average Precision (MAP): MAP specifies the number of relevant documents for a given query from the set of retrieved documents. For the set of all relevant documents *D* = {*d*, ... , *dmj*} and for a query *qj* in *Q*, the MAP for the overall retrieval effectiveness is computed using Equation (10).

$$\text{MAP} = \frac{1}{\text{Q}}\sum_{\text{j}=1}^{|\text{Q}|}\frac{1}{\text{mj}}\sum_{\text{k}=1}^{\text{mj}}\text{p}\left(R_{jk}\right) \tag{10}$$

where *Rjk* represents the set of the top *k* ranked retrieval results.

Normalized Discounted Cumulative Gain (NDCG): Discounted cumulative gain (DCG) predicts the relevance of a document based on its rank among the retrieved documents. The relevance measure is accumulated from top to bottom, discounting the value of documents at lower ranks. NDCG at *k* measures DCG for the top *k* documents, normalizing by the highest possible value for a query. The NDCG values for all queries can be averaged to reliably evaluate the effectiveness of a ranking algorithm for various information needs across a collection. Given a set of queries $\text{q}_\text{j}\ \epsilon\ \text{Q}$ and relevance judgments Rd_j for a document *d*, the NDCG is computed using Equation (11).

$$\text{NDCG}(\text{Q},\ \text{k}) = \frac{1}{|\text{Q}|}\sum_{\text{j}=1}^{|\text{Q}|}\text{Z}_{\text{kj}}\ \sum_{\text{m}=1}^{\text{k}}\frac{2^{\text{R}_{\text{jm}}}\ -\ 1}{\log_2(1+\ \text{m})} \tag{11}$$

The performance of the fine-tuned models is measured using F-score metrics, which are computed using Equation (12).

$$\text{F1} - \text{score} = 2 \times \frac{\text{R} \times \text{P}}{\text{R} + \text{P}} \tag{12}$$

where Recall (R) = $\frac{\text{TP}}{\text{TP+FN}}$ and Precision (P) = $\frac{\text{TP}}{\text{TP+FP}}$

Binary preference (bpref): It measures the number of judged non-relevant documents that are located above each relevant document in the ranked list. It is computed using Equation (13).

$$\text{bref} = \frac{1}{R} \sum_{\text{D}} (1 - \frac{|D_R \text{ ranked higher than D}|}{\min(|\text{R}|, |\text{N}|)} \tag{13}$$

where R is the number of judged relevant documents, N is number of judged non-relevant documents, D is a relevant retrieved document, and D_R is a member of the first R judged non-relevant documents retrieved.

5. Results and Discussion

We carried out various experiments to create Amharic pre-trained word embedding and BERT models, and evaluated them in NLP and retrieval tasks. In this section, we demonstrate the essence of BERT on Amharic language by fine-tuning the pre-trained models and investigating the retrieval effect of query expansion using word embeddings. Therefore, we present the intrinsic and extrinsic evaluations with detailed analysis in the following sections.

5.1. Result

5.1.1. Pre-Trained BERT Models

The Amharic pre-trained MLM model was trained to predict masked tokens. The training loss values decreased from epoch 0 to 5. The best training loss value was achieved at epoch 5. The aggregate loss value during training is presented in Table 4. As shown in the table, the Amharic word-based pre-trained BERT MLM model provides less training loss than others on epoch 5.

Table 4. Loss values of MLM model.

Dataset	Training Loss	
	Epoch 5	**Epoch 10**
Word-based	0.253	0.480
Stem-based	0.458	0.497
Root-based	0.507	0.614

Three NSP-BERT models were trained using word-based, stem-based, and root-based corpora to predict whether two sentences appear consecutively within a document or not. After training, we manually selected 50 pairs of sentences from each corpus in order to evaluate the accuracy of the models. Each pair of sentences was tokenized using the same WordPiece tokenizer, which is also used for the corpora. The training loss, validation loss and accuracy are presented in Table 5. As shown in the table, the testing loss and accuracy of the word-based NSP model was better than stem-based and root-based datasets on the same evaluation dataset at epoch 5. However, the model selects pairs of sentences randomly during the training phase. As a result of this, the training loss values of word-based, stem-based, and root-based were based on different samples, making it difficult to directly compare training loss values.

Table 5. Loss and accuracy of NSP models (best performance is in bold font).

Dataset	Model	Training Loss		Testing Loss		Accuracy	
		Epoch 5	Epoch 10	Epoch 5	Epoch 10	Epoch 5	Epoch 10
Word	base	0.211	2.84×10^{-5}	0.307	1.443	**0.68**	0.52
Stem	base	0.343	0.051	0.652	1.74	**0.66**	0.64
Root	base	0.432	0.0057	0.654	1.663	**0.64**	0.60

5.1.2. Fine Tuning

The Amharic pre-trained MLM model was fine-tuned for document classification tasks. Our pre-trained MLM was retrained using labeled datasets to create two fine-tuned models: a fine-tuned model to classify documents as relevant or non-relevant for 50 queries, and a fine-tuned model to classify 1189 labeled documents based on their subjects. These two fine-tuned models were created on top of the pre-trained word-based MLM model. The labeled datasets included both training and validating datasets used for training and evaluating the fine-tuned models. The labeled training and validating documents in the datasets were preprocessed using the same technique used for document processing during training MLM and NSP models. Our labeled datasets had two columns: the document and label columns. In the case of document relevance identification, the values of the label column were relevant and non-relevant, whereas in the case of document classification based on subject, the values of the label columns were education, religion, politics, sport, etc. Each of the queries were trained with both relevant and non-relevant documents. Since there was no usable Amharic stemmer and morphological analyzer to extract stems and roots from labeled dataset, we could not evaluate the stem-based and root-based versions of the pre-trained MLM models. Since the fine-tuned models have consistent training and validation loss values after epoch 14, we trained them for the first 17 epochs and reported the results. Their training and validation losses were decreased from epoch 0 to 17 (see Figure 5). Their F1-score and accuracy values were improved from epoch 0 to 17. On the graph legend, R represents relevance, and S represents subject of a document.

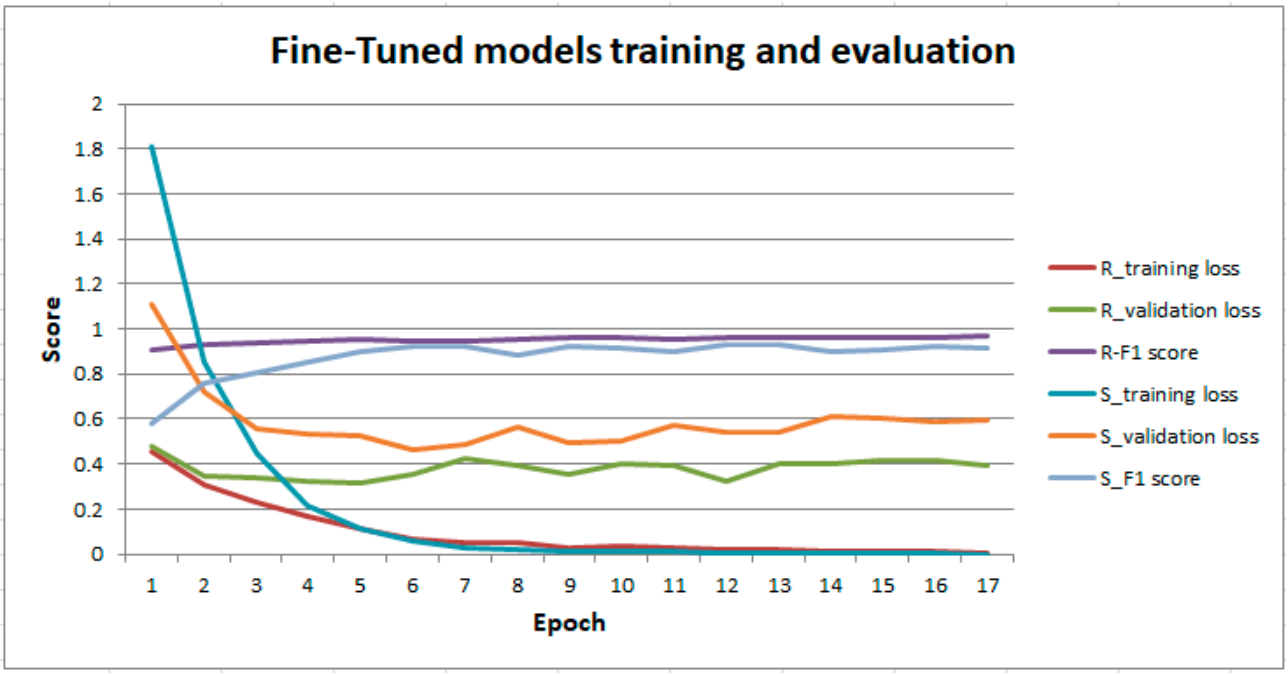

Figure 5. Training and evaluation of BERT word-based fine-tuned models for document classification based on relevance and subject.

As depicted in Figure 5, after epoch 14, the training, validation, and F1 scores of the two fine-tuned models did not show significant change. The optimal loss, F1-score (weighted), and accuracy of the fine-tuned models are reported in Table 6. As shown in Table 6, the performance of document classification models was high because the training dataset was of a moderate size and balanced, and the size of almost all training and testing documents was less than 512 tokens, which is supported by BERT (see Figure 6).

Table 6. Fine-tuning results of documents classification.

Task	Training Loss	Validation Loss	F1-Score	Accuracy
Document classification based on subject	0.03	0.58	0.91	0.89
Relevance classification	0.02	0.38	0.97	0.95

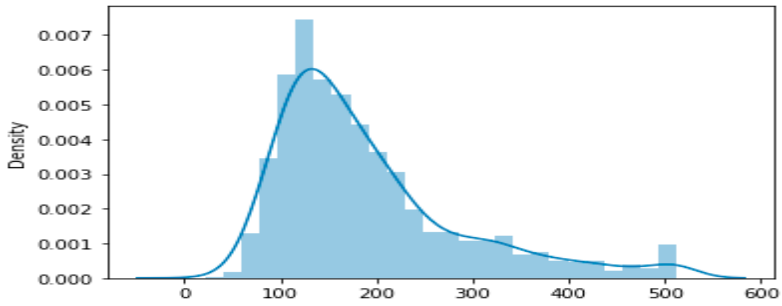

Figure 6. Number of tokens in document classification training and testing datasets.

5.1.3. Query Expansion Using Word Embedding

The retrieval effectiveness of query expansion using the three word embedding models was evaluated. We ran 240 queries on word-based, stem-based, and root-based index files. In the case of stem-based and root-based retrieval, documents and user information needs were morphologically processed before indexing and query expansion. The retrieval results of each query were processed to convert them into TREC format, which can be used by Trec-Eval toolkit. The retrieval effectiveness of query expansion using word2vec, GloVe, and fastText is presented in Tables 7–9, respectively.

Table 7. Retrieval effectiveness of query expansion using word2vec.

Dataset	Retrieval Effectiveness									
	CBOW					Skip-gram				
	P@5	P@10	ndcg	R-prec	bpref	P@5	P@10	ndcg	R-prec	bpref
Word-based	0.53	0.51	0.70	0.45	0.43	0.53	0.51	0.70	0.45	0.43
Stem-based	0.40	0.35	0.55	0.32	0.30	0.40	0.35	0.55	0.32	0.30
Root-based	0.45	0.40	0.66	0.38	0.36	0.44	0.38	0.66	0.36	0.35

Table 8. Retrieval effectiveness of query expansion using GloVe.

Datasets	Retrieval Effectiveness				
	P@5	P@10	ndcg	R-prec	bpref
Word-based	0.54	0.50	0.70	0.45	0.43
Stem-based	0.36	0.34	0.55	0.32	0.29
Root-based	0.41	0.38	0.66	0.37	0.34

Table 9. Query expansion retrieval effectiveness using fastText.

Dataset	Retrieval Effectiveness									
	CBOW					Skip-Gram				
	P@5	**P@10**	**ndcg**	**R-Prec**	**bpref**	**P@5**	**P@10**	**ndcg**	**R-Prec**	**bpref**
Word-based	0.50	0.44	0.66	0.37	0.34	0.60	0.56	0.75	0.49	0.47
Stem-based	0.30	0.27	0.51	0.26	0.22	0.32	0.30	0.52	0.26	0.23
Root-based	0.34	0.30	0.62	0.32	0.29	0.37	0.33	0.62	0.32	0.29

As indicated in Table 7, the retrieval performance of the word-based approach was better than the stem-based and root-base ones. On the three datasets, the CBOW and skip-gram models performed equally or closely on the same dataset. As presented in Table 8, word-based retrieval outperformed the stem-based and root-based, with a large difference. As shown in Table 9, as with word2vec and GloVe, retrieving using a word-based approach was better than the others. The skip-gram model outperformed CBOW. This indicates that skip-gram is more suitable for obtaining or grouping similar Amharic words.

For the three word embeddings, the retrieval effectiveness of the word-based approach was better than the root-based approach, which, in turn, was better than the stem-based one. The retrieval effectiveness of the three word embedding models also decreased from P@5 to P@10 (See Tables 7–9). The main reason is that an insufficient relevant number of documents existed in the test collection.

5.2. Discussion

Test results show that the retrieval effectiveness of query expansion using word-base embedding was found to be better than that of stem-based and root-based embedding. The reason is that word-based query expansion returns many more variants of query terms than the stem-based and root-based learned text representations. The stem-based learned text representation returned fewer variants of query terms. However, in the case of the root-based representation, none of the expanded terms were variants to one of the query terms. For example, the top ten expanded terms for the query terms በሽታ */bəʃita* 'disease'/, የምርመራ */jəmiriməra* 'of diagnosis'/, አገልግሎት */ʔəgəligiloti* 'service'/, መቆጣጠሪያ */mək'ot'at'ərija* 'controlling'/, and አክራሪነት */ʔəkirarinəti* 'extremism'/ are presented in Table 10.

Furthermore, word-based learned text representation returned many variants which were semantically related to a query word. For example, for the query term የፖለቲካ */jəpolətika* 'of politics'/, the expanded terms were የፓርቲዎች */jəparitiwotʃi* 'of parties'/, ፓርቲዎች the parties'/, */paritiwotʃi* 'the parties'/, ፓርቲዎችን */paritiwotʃini* 'the parties *[acc]*'/, በፓርቲዎች */bəparitiwotʃi* 'by the parties'/, ፓርቲዎችና */paritiwotʃina* 'parties and'/, የፓርቲዎችን */jəparitiwotʃini* 'of the parties'/, የፖለቲካና */jəpolətikana* 'of politics and'/, የፖለቲካ */jəpolətika* 'of the politics'/, ፖለቲካ */polətika* 'politics'/, and ፓርቲነት */paritinəti* 'being party'/. Out of these words, የፖለቲካና */jəpolətikana/*, የፖለቲካ */jəpolətika/*, and ፖለቲካ */polətika/* are variants of the query word የፖለቲካ */jəpolətika/*, whereas the remaining expanded terms are variants of ፓርቲ */pariti* 'party'/, which is semantically related to the query term. As shown in Tables 7–9, fastText word embedding was more suitable to Amharic IR than word2vec and GloVe word embeddings, for two reasons.

First, fastText can handle most of the morphological variants of a word in a better way than other word embeddings. Most expanded terms of fastText are morphological variants. FastText expands many query terms by more variants than word2vec and GloVe models. For example, the expanded terms for the query term አቅርቦትና */ʔək'iribotina* 'supply and'/ are presented in Table 11.

Table 10. Sample expanded terms using fastText model.

Model	Word	Top 10 Most Similar Words
Word	በሸታ	በበሸታ, በሸታው, በበሸታው, በሸታወች, በሸታን, በሸ, በቫይረሱ, ቲቢ, ቫይረሶቹ, በኤችአይቪ
	የምርመራ	የምርመራው, የምርመራና, የምርመራውን, የሙከራ, የቅስቀሳ, የጨረራ, የምርቃት, የቆጠራ, ለምርመራ, የሰመራ
	አገልግሎት	አገልግሎ, አገልግሎት, አገልግሎትም, አገልግሎቱ, አገልግሎቱም, አገልግሎትና, አገልግሎቱንና, አገልገሎት, አግልግሎት, አገልግሎታቸው
	መቆጣጠሪያ	መቆጣጠሪያና, መከታተያ, መቁጠሪያ, መቆያ, መቋቋሚያ, መከራከሪያ, መደገፊያ, ማጥፊያ, ማቆምያ, መደርደሪያ
	አከራሪነት	አከራሪነት, አከራሪነትና, አከራሪነትንና, አከራሪ, አከራሪና, አሸባሪነት, አከራሪነትን, አሸባሪነትና, አርበኝነት, አሳቢነት
Stem	በሸታ	ኤችፒቪ, ቫይረስ, ካንሰር, ሽታ, በሽ, በሸት, ህመም, ህክም, ክትብ, አይቪ
	ምርመር	መርመር, አጣር, ቆጣር, ህክም, ጣር, ጥርጣር, ምረመር, አዲት, ቆጣጣር, ከምር
	ገልግል	ገልግል, ገል, ገልጥ, ገልል, ግልግል, ገልበጥ, ገለገል, ገልገል, ገላገል, ገልጋል
	ቆጣጠር	ቆጣር, ቆጣጥር, ነጠር, ቆጣጣር, ከሰር, ከላካል, ፈንጠር, ከላከል, ቆጣ, ቁጠር
	ከራር	ከራር, ከራክ, ከራሪ, ጽንፍ, አከራር, አከራ, ክፍ, ሽብር, ጸረሽብር, ሙስሊም
Root	በሸታ	በሸታ, ሽታ,ቫይረስ, በሸ, ሸት, ካንሰር, ወባ, ቲቢ, ቫይርስ, ኤችፒቪ
	ም-ር-ም-ር	ክ-ም-ር, ም-ር-ጥ, ም-ም-ር, ም-ር-ር, ድ-ም-ር, እ-ም-ር, ም-ር-ዝ, ም-ር-ክ-ዝ, ም-ር-ቅ, ት-ር-ም-ስ
	ግ-ል-ግ-ል	ግ-ግ-ል, ግ-ል, እ-ግ-ል, ግ-ል-ት, ድ-ግ-ል, ግ-ል-ል, ግ-ል-ብ, ግ-ው-ል-ም-ስ, ን-ግ-ል-ት, ግ-ል-ጥ
	ቅ-ጥ-ጥ-ር	ን-ጥ-ጥ-ር, ቅ-ጥ-ር, ብ-ጥ-ር, ን-ጥ-ር, ቅ-ጥ-ቅ-ጥ, ስ-ቅ-ጥ-ጥ, ጥ-ር-ጥ-ር, ስ-ብ-ጥ-ር, ው-ጥ-ር, ቅ-ጥ-ፍ
	ክ-ር-ር	ክ-ር-ር, ክ-ር-ክ-ር, ክ-ር, ክ-ር-ይ, ጽ-ር-ር, ክ-ር-ም, ክ-ር-ፍ, ክ-ር-ምት, ብ-ር-ር, ክ-ን-ን

Table 11. Top four expanded terms of the query word አቅርቦትና */ʔək'iribotina*. Roots of verbs are represented by consonant separated by hyphen (-).

Model	Conventional Text Representation		
	Word-Based	**Stem-Based**	**Root-Based**
word2vec	አቅርቦት, የግብአት, የግብርና, በሲሚንቶ	ቅረብ, ቀርብ, ቀረብ, ኤሌክትሪክ	ድ-ር-ግ, አቃቢ, ፍ-ል-ግ, እ-ጥ-ር
GloVe	አቅርቦትን, የግብአት, አቅርቦት, ፍላጐትና	ዋል, ምስጋን, ጥር, ወልድም	ም-ስ-ግ-ን, ጥ-ር, ሪፖርት, ጥ-ይ-ቅ
fastText	የአቅርቦትና, በአቅርቦትና, አቅርቦት, የአቅርቦት	ምስግን, ዋል, ቀርብ, ምስጋናው	ጥ-ር, ቅርስ, ም-ል-ክ-ት, ክ-ር-ክ-ር

Second, the fastText algorithm addresses the out-of-vocabulary problem, which is the main challenge of word2vec and GloVe. Some query terms are not found in vocabulary/dictionary of the word2vec and GloVe models. As a result, such words are not expanded in the case of the word2vec and GloVe models. However, the fastText model returns expanded terms even though such words do not exist in its vocabulary. If a word is unseen during training, fastText segments a word into n-grams and generates its embedding. As a result, it helps to embed rare words, misspelled words, and words that do not exist in corpora but are found in the topic set. For example, the query term አገልግሎት/*ʔəgəligiloti* 'services'/ is not found in the corpora, and thus the word2vec and GloVe models do not return any expanded terms. However, fastText returns the words አገልግሎ */ʔəgəligilo/*, አገልግሎትም */ʔəgəligilotimi/*, አገልግሎትን */ʔəgəligilotini/*, አገልግሎትና */ʔəgəligilotina/*, አገልገሎት */ʔəgəligəloti/*, አገልግሎቱንና */ʔəgəligilotunina/*, አገልገሎቱን */ʔəgəligəlotuni/*, አገልግሎቱን */ʔəgəligilotuni/*, and አገልግሎቱም */ʔəgəligilotumi/*, which are variants referring to the concept "serve". Furthermore, fastText based on skip-gram outperforms the baseline retrieval performance reported in [46]. The effectiveness of the Amharic retrieval system without and with query expansion using fastText is presented in Table 12.

Table 12. Effectiveness of Amharic retrieval with and without query expansion.

Technique	Precision					ndgc	R-Prec
	P@5	P@10	P@15	P@20	MAP		
Conventional	0.56	0.49	0.44	0.40	0.43	0.65	0.43
With query expansion	0.60	0.56	0.50	0.47	0.51	0.75	0.49

Like word embeddings, the pre-trained and fine-tuned Amharic BERT models performed better on the word-based form over the other forms (see Tables 4–6). As depicted in Table 6, most fine-tuned BERT models show promising results for Amharic text classification and IR tasks. On Amharic IR, BERT significantly outperformed all embedding algorithms. For example, the F1-score and accuracy values of fine-tuned BERT model to classify documents as relevant and non-relevant for the IR system were higher than word embedding models, as the BERT model represents each query and document word considering contextual information of a word in a sentence or a phrase. From learned text representation techniques, fastText and BERT models are better at understanding and handling the specificity of Amharic, such as morphology, syntax, and semantics.

As shown in Tables 4 and 5, word-based BERT models outperformed both stem-based and root-based approaches. Even though stems and roots are the fundamental units for many text processing tasks, the WordPiece tokenizer splits a stem-based and root-based token into sub-units if it does not recognize a token. As a result, the performances of stem-based and root-based approaches are degraded compared to that of the word-based approach. The accuracy of the pre-trained Amharic BERT models decreases when the number of epochs is greater than five. The best training performance is achieved at epoch 5. We make the models that produce the best performance accessible to the research community at https://www.irit.fr/AmharicResources/amharic-pre-trained-language-models/, accessed on 7 August 2022. The resources we share on this link are:

i. Three BERT WordPiece tokenizers (word-based, stem-based, and root-based);
ii. Three BERT byte-pair encoding (BPE) tokenizers (word-based, stem-based, and root-based);
iii. Three BERT masked language models (word-based, stem-based, and root-based);
iv. Three BERT next sentence prediction models (word-based, stem-based, and root-based);
v. Nine word embedding models (word2vec, GloVe and fastText; each has word-based, stem-based and, root-based versions).

The constructed BERT model achieves remarkable results on verified Amharic NLP and IR down streaming tasks. However, the effect of word embedding to the improve retrieval effectiveness of IR systems in many languages is low. The performance of our models can be compared to some other languages' models using Tables 6 and 13.

Table 13. Performance of word embeddings and BERT models on some tasks on some languages.

Language	Down Streaming Task	Model	Performance
English [54]	Document classification	BERT	0.96 (F1-score)
Chinese [55]	Document classification	BERT	0.97 (accuracy)
English [25]	*Ad hoc* retrieval	word2vec	0.48 (NDCG)
English [56]	Query expansion	word2vec	0.086 (precision)
		GloVe	0.087 (precision)
		fastText	0.087 (precision)

In this study, an investigation was made to build Amharic word embedding and BERT models and verify their use for Amharic NLP and IR tasks. Due to some challenges, such

as experimental resource and computational machine scarcity, we have not explored the following things in our experiments.

i. Since our datasets are small relation to the requirement of BERT model, we tested only the effect of BERT at the base case. However, recently, BERT, at a large scale, has performed better than the base case on huge datasets. Therefore, the performance of the BERT model can be further enhanced with the use of a large corpus.
ii. Recently, the use of the BERT model for ranking retrieval results has provided better performance. In our study, we evaluated the performance of the BERT model to identify relevant and non-relevant documents for a query rather than ranking. Thus, our IR system can be extended to include ranking of retrieval results using the BERT model.

5.3. Application of the Pre-Trained Word Embedding and BERT Models

One of the aims of this research was to create pre-trained Amharic word embedding and BERT models and make them publicly accessible. We trained and released our pre-trained Amharic word2vec, GloVe, fastText, and BERT models, and Amharic BERT tokenizers, which can motivate the research community to carry out experiments on the pre-trained Amharic language models. Pre-training is fairly expensive, but it is a one-time procedure. Our pre-trained models can be used to reduce the need for many heavily-engineered tasks. The extracted features can be fed into a machine learning model so as to work with text data and preserve the semantic and syntactic information. The constructed Amharic models can be used in different applications. For example, our word embedding vectors can be used for word analogy tasks, namely entity recognition and chunking. The pre-trained Amharic BERT models can also be fine-tuned with some labeled data for a range of specific tasks, such as text classification, ranking, sentiment analysis, question answering, named entity recognition, natural language inference, co-reference, information extraction, semantic parsing, etc.

6. Conclusions

Amharic exhibits complex morphological characteristics. It is one of the under-resourced languages. There is lack of usable and publicly accessible pre-trained language models for this language. Because of this, the effect of learned text representation on NLP tasks has not yet been investigated. In this study, we explored the impact of word embedding (word2vec, GloVe, fastText) and BERT models on Amharic text. Word-based, stem-based, and root-based corpora were used to investigate the effect of word embedding models on query expansion. Moreover, we built Amharic pre-trained BERT models and fine-tuned them for text classification. The usability of our pre-trained BERT models were evaluated in IR and NLP tasks. The training and validating losses of the fine-tuned BERT model decreased as number of epochs increased. In contrast, their F1-score and accuracy improved from the first epoch to the next. We also made the pre-trained word embeddings, BERT models, and BERT tokenizers publicly accessible at https://www.irit.fr/AmharicResources/amharic-pre-trained-language-models/, (accessed on 19 May 2022). From the experiments we performed on the constructed models, we found that learned text representation using a word-based approach outperformed stem-based and root-based approaches across all tasks considered in this research. Promising results were obtained from the BERT and fastText models. As we created pre-trained models at base scale, our work can be further enhanced by increasing the size of the corpus. Therefore, this work can be expanded with the aim of investigating the significance of model construction at large scale using a huge dataset. Furthermore, our pre-trained models can be fine-tuned for different NLP applications by preparing more validation and testing datasets.

Author Contributions: Conceptualization, T.Y., J.M. and Y.A.; formal analysis, T.Y. and Y.A.; funding acquisition, J.M.; investigation, T.Y.; methodology, J.M.; resources, T.Y.; software, T.Y.; supervision,

J.M. and Y.A.; validation, T.Y., J.M. and Y.A.; writing—original draft, T.Y.; writing—review and editing, J.M. and Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Experiment dataset and resources are available online at https://www.irit.fr/AmharicResources/ (accessed on 19 May 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Z.; Lin, Y.; Sun, M. *Representation Learning for Natural Language Processing*; Open access e-book; Springer: Berlin/Heidelberg, Germany, 2020; ISBN 978-981-15-5572-5. Available online: https://link.springer.com/book/10.1007/978-981-15-5573-2 (accessed on 15 May 2022).
2. Manning, C.; Raghavan, P.; Schutze, H. *Introduction to Information Retrieval*; Draft e-book; Cambridge University Press: Cambridge, UK, 2010; Available online: https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf (accessed on 22 June 2022).
3. Sebastiani, F. Machine learning in automated text categorization. *ACM Comput. Surv.* **2002**, *34*, 1–47. [CrossRef]
4. Tellex, S.; Katz, B.; Lin, J.; Fernandes, A.; Marton, G. Quantitative evaluation of passage retrieval algorithms for question answering. In Proceedings of the 26th Annual international ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, ON, Canada, 28 July–1 August 2003; pp. 41–47.
5. Turian, J.; Ratinov, L.; Yoshua, B. Word representations: A simple and general method for semi-supervised learning. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; pp. 384–394.
6. Socher, R.; Bauer, J.; Manning, C.; Ng, A.Y. Parsing with compositional vector grammars. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; pp. 455–465.
7. Babic, K.; Martinčić-Ipšić, S.; Meštrovi'c, A. Survey of neural text representation models. *Information* **2020**, *11*, 511. [CrossRef]
8. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
9. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
10. Logeswaran, L.; Lee, H. An efficient framework for learning sentence representations. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–16.
11. Kusner, M.; Sun, Y.; Kolkin, N.; Weinberger, K. From word embeddings to document distances. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 957–966.
12. Zhou, G.; He, T.; Zhao, J.; Hu, P. Learning continuous word embedding with metadata for question retrieval in community question answering. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 27–29 July 2015; pp. 250–259.
13. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
14. Athiwaratkun, B.; Gordon, A.; Anandkumar, A. Probabilistic fastText for multi-sense word embeddings. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 1–11.
15. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. 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, Minneapolis, Minnesota, 3–5 June 2019; pp. 4171–4186.
16. Antoun, W.; Bal, F.; Hajj, H. Arabert: Transformer-based model for Arabic language under-standing. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 12 May 2020; pp. 9–15.
17. Delobelle, P.; Winters, T.; Berendt, B. RobBERT: A Dutch RoBERTa-based language model. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–18 November 2020; pp. 3255–3265.
18. Polignano, M.; Basile, P.; Gemmis, M.; Semeraro, G.; Basile, V. AlBERTo: Italian BERT language understanding model for NLP challenging tasks based on tweets. In Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-It 2019), Bari, Italy, 13–15 November 2019; Volume 2481.
19. Terumi, E.; Vitor, J.; Knafou, J.; Copara, J.; Oliveira, L.; Gumiel, Y.; Oliveira, L.; Teodoro, D.; Cabrera, E.; Moro, C. BioBERTpt-A Portuguese neural language model for clinical named entity recognition. In Proceedings of the 3rd Clinical Natural Language Processing Workshop, Online, 19 November 2020; pp. 65–72.
20. Kuratov, Y.; Arkhipov, M. Adaptation of deep bidirectional multilingual transformers for Russian language. In Proceedings of the Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2019", Newral Networks and Deep Learning Lab, Moscow, Russia, 29 May–1 June 2019; pp. 1–7.

21. Martin, L.; Muller, B.; Javier, P.; Suárez, O.; Dupont, Y.; Romary, L.; Villemonte, É.; Clergerie, D.; Seddah, D.; Sagot, B. CamemBERT: A Tasty French language model. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7203–7219.
22. Han, X.; Zhang, Z.; Ding, N.; Gu, Y.; Liu, X.; Huo, Y.; Qiu, J.; Yao, Y.; Zhang, A.; Zhang, L.; et al. Pre-trained models: Past, present and future. *AI Open* **2022**, *2*, 225–250. [CrossRef]
23. Yeshambel, T.; Mothe, J.; Assabie, Y. Evaluation of corpora, resources and tools for Amharic information retrieval. In Proceedings of the ICAST2020, Bahir Dar, Ethiopia, 2–4 October 2020; pp. 470–483.
24. Diaz, F.; Mitra, B.; Craswell, N. Query expansion with locally-trained word embeddings. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 367–377.
25. Aklouche, B.; Bounhas, I.; Slimani, Y. Query expansion based on NLP and word embeddings. In Proceedings of the 27th Text REtrieval Conference (TREC 2018), Gaithersburg, ML, USA, 14–16 November 2018; pp. 1–7.
26. Getnet, B.; Assabie, Y. Amharic information retrieval based on query expansion using semantic vocabulary. In Proceedings of the 8th EAI International Conference on Advancements of Science and Technology, Bahir Dar, Ethiopia, 2–4 October 2021; pp. 407–416.
27. Deho, O.; Agangiba, W.; Aryeh, F.; Ansah, J. Sentiment analysis with word embedding. In Proceedings of the 2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST), Accra, Ghana, 22–24 August 2018; pp. 1–4.
28. Acosta, J.; Norissa, L.; Mingxiao, L.; Ezra, F.; Andreea, C. Sentiment analysis of twitter messages using word2Vec. In Proceedings of the Student Faculty Research Day, CSIS, New York, NY, USA, 5 May 2017; pp. 1–7.
29. Medved, M.; Horák, A. Sentence and Word embedding employed in Open question-Answering. In Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018), Funchal, Portugal, 16–18 January 2018; pp. 486–492.
30. Sun, Y.; Zheng, Y.; Hao, C.; Qiu, H. NSP-BERT: A Prompt-based few-shot learner through an original pre-training task—Next sentence prediction. In Proceedings of the 29th International Conference on Computational Linguistics, Yeongju, Republic of Korea, 12–17 October 2022; pp. 3233–3250.
31. Shi, W.; Demberg, V. Next sentence prediction helps implicit discourse relation classification within and across domains. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 5790–5796.
32. Bai, H.; Zhao, H. Deep enhanced representation for implicit discourse relation recognition. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 571–583.
33. Prasad, R.; Dinesh, N.; Lee, A.; Miltsakaki, E.; Robaldo, L.; Joshi, A.; Webber, B. The Penn Discourse TreeBank 2.0. In Proceedings of the Sixth Conference on International Language Resources and Evaluation (LREC-2008), Marrakech, Morocco, 28–30 May 2008; pp. 2961–2968.
34. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z. Pre-training with whole word masking for Chinese BERT. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3504–3514. [CrossRef]
35. Aggarwal, A.; Chauhan, A.; Kumar, D.; Mittal, M.; Verma, S. Classification of fake news by fine-tuning deep bidirectional transformers based language model. *EAI Endorsed Trans. Scalable Inf. Syst.* **2020**, *27*, e10. [CrossRef]
36. Protasha, N.; Sam, A.; Kowsher, M.; Murad, S.; Bairagi, A.; Masud, M.; Baz, M. Transfer learning for sentiment analysis using BERT based supervised fine-tuning. *Sensors* **2022**, *22*, 4157. [CrossRef] [PubMed]
37. Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; Mikolov, T. Learning word vectors for 157 languages. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; pp. 3483–3487.
38. Eshetu, A.; Teshome, G.; Abebe, T. Learning word and sub-word vectors for Amharic (Less Resourced Language). *Int. J. Adv. Eng. Res. Sci. (IJAERS)* **2020**, *7*, 358–366. [CrossRef]
39. Muhie, S.; Ayele, A.; Venkatesh, G.; Gashaw, I.; Biemann, C. Introducing various semantic models for Amharic: Experimentation and evaluation with multiple tasks and datasets. *Future Internet* **2021**, *13*, 275. [CrossRef]
40. Yeshambel, T.; Mothe, J.; Assabie, Y. Amharic *adhoc* information retrieval system based on morphological features. *Appl. Sci.* **2021**, *12*, 1294. [CrossRef]
41. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised cross-lingual representation learning at scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8440–8451.
42. Yeshambel, T.; Mothe, J.; Assabie, Y. Construction of morpheme-based Amharic stopword list for information retrieval system. In Proceedings of the 8th EAI International Conference on Advancements of Science and Technology, Bahir Dar, Ethiopia, 2–4 October 2020; pp. 484–498.
43. Alemayehu, N.; Willett, P. The effectiveness of stemming for information retrieval in Amharic. *Program Electron. Libr. Inf. Syst.* **2003**, *37*, 254–259. [CrossRef]
44. Yimam, B. *Yamarigna Sewasiw (Amharic Grammar)*, 2nd ed.; CASE: Addis Ababa, Ethiopia, 2000.
45. Wolf, L. *Reference Grammar of Amharic*, 1st ed.; Otto Harrassowitz: Wiesbaden, Germany, 1995.
46. Yeshambel, T.; Mothe, J.; Assabie, Y. Amharic document representation for adhoc retrieval. In Proceedings of the 12th International Conference on Knowledge Discovery and Information Retrieval, Online, 2–4 November 2020; pp. 124–134.
47. Arora, P.; Foster, J.; Jones, G. Query expansion for sentence retrieval using pseudo relevance feedback and word embedding. In Proceedings of the CLEF 2017, Dubline, Ireland, 11–14 September 2017; pp. 97–103.

48. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. How to fine-tune BERT for text classification? In Proceedings of the 21st China National Conference on Chinese Computational Linguistics, Nanchang, China, 14–16 October 2020; pp. 194–206.
49. Palotti, J.; Scells, H.; Zuccon, G. TrecTools: An Open-source Python Library for Information Retrieval Practitioners Involved in TREC-like Campaigns. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19), Paris, France, 21–25 July 2019; pp. 1325–1328. [CrossRef]
50. Yeshambel, T.; Mothe, J.; Assabie, Y. 2AIRTC: The Amharic Adhoc information retrieval test collection. In Proceedings of the CLEF 2020, LNCS 12260, Thessaloniki, Greece, 22–25 September 2020; pp. 55–66.
51. Yeshambel, T.; Mothe, J.; Assabie, Y. Morphologically annotated Amharic text corpora. In Proceedings of the 44th ACM SIGIR Conference on Research and Development in Information Retrieval, Online, 11–15 July 2021; pp. 2349–2355.
52. Kingma, D.; Ba, J. Adam: A Method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
53. Lin, J.; Nogueira, R.; Yates, A. Pre-trained transformers for text ranking: BERT and Beyond. In Proceedings of the NAACL-HLT, Mexico City, Mexico, 6–11 June 2021; pp. 1–4.
54. Limsopatham, N. Effectively leveraging BERT for legal document classification. In Proceedings of the Natural Legal Language Processing Workshop 2021, Punta Cana, Dominican Republic, 10 November 2021; pp. 210–216.
55. Chen, X.; Cong, P.; Lv, S. A long-text classification method of Chinese news based on BERT and CNN. *IEEE Access* **2022**, *10*, 34046–34057. [CrossRef]
56. Goyal, T.; Bhadola, S.; Bhatia, K. Automatic query expansion using word embedding based on fuzzy graph connectivity measures. *Int. J. Trend Sci. Res. Dev. (IJTSRD)* **2021**, *5*, 1429–1442.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.