Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2024, Article ID 4429069, 11 pages
https://doi.org/10.1155/2024/4429069

Research Article

Semisupervised Learning-Based Word-Sense Disambiguation Using Word Embedding for Afaan Oromoo Language

Tabor Wegi Geleta [1] **and Jara Muda Haro** [2]

[1]*Department of Information Science, Informatics College, Bule Hora University, Bule Hora, Ethiopia*
[2]*Department of Information Technology, Informatics College, Bule Hora University, Bule Hora, Ethiopia*

Correspondence should be addressed to Jara Muda Haro; jaramudah123@gmail.com

Received 21 March 2022; Revised 19 November 2022; Accepted 1 March 2024; Published 14 March 2024

Academic Editor: Ahmad Al- omari

Natural language is a type of language that human beings use to communicate with each other. However, it is very difficult to communicate with a machine-understandable language. Finding context meaning is challenging the activity of automatically identifying machine translation, indexing engines, and predicting neighbor words in natural language. Many researchers around the world investigated word-sense disambiguation in different languages, including the Afaan Oromo language, to solve this challenge. Nevertheless, the amount of effort for Afaan Oromo is very little in terms of finding context meaning and predicting neighbor words to solve the word ambiguity problem. Since the Afaan Oromo language is one of the languages developed in Ethiopia, it needs the latest technology to enhance communication and overcome ambiguity challenges. So far, this work aims to design and develop a vector space model for the Afaan Oromo language that can provide the application of word-sense disambiguation to increase the performance of information retrieval. In this work, the study has used the Afaan Oromo word embedding method to disambiguate a contextual meaning of words by applying the semisupervised technique. To conduct the study, 456,300 Afaan Oromo words were taken from different sources and preprocessed for experimentation by the Natural Language Toolkit and Anaconda tool. The *K*-means machine learning algorithm was used to cluster similar word vocabulary. Experimental results show that using word embedding for the proposed language's corpus improves the performance of the system by a total accuracy of 98.89% and outperforms the existing similar systems.

1. Introduction

Natural language is a human being language used to communicate with each other. While it seems simple for humans to know intent of a word according to their context, it is tough for the machine to understand human language. Word-sense disambiguation (WSD) is a difficult task for computers and thus requires sophisticated means of machine learning. During natural language processing, many word ambiguities will occur. Ethiopia has more than 85 languages spoken by a native speaker with defined language structure [1]. Among these languages, Afaan Oromo (Oromo language) is one of the major languages spoken by native speakers.

Afaan Oromo is a Cushitic language spoken by about 50 million people in Ethiopia (about 40% of the country's population), Kenya, Somalia, Egypt, and Djibouti, and it is the third largest language in Africa after Arabic and Hausa [2]. Afaan Oromo used a Latin-based alphabet or writing system called "Qubee" which has 26 basic characters and has served as the official language of the Oromia regional state and has served as an academic language for primary schools of the region up to now [3]. This language has a folklore that delivered as fields of study in most Ethiopian universities and other countries [4]. There are a number of ambiguous words in Afaan Oromo language like other African and Ethiopian languages. Hence, it is difficult to understand the meaning of those words in a given context as it is rich with

ambiguity in semantics, which can benefit from WSD research and development.

Many researchers have done word-sense disambiguation in different languages including Afaan Oromo and Amharic from local languages using different methods such as supervised and rule-based approaches. However, there have been few attempts made, as far as the knowledge of researchers is concerned, to develop semisupervised WSD for Afaan Oromo using the word embedding technique.

This study aimed to solve the ambiguity of the Afaan Oromo language using the proposed approach—the problem that a given word can have several related or unrelated meanings depending on a given context.

For example, consider the following two sentences each with a different sense of ambiguous word "Soquu."

(a) Jaarson hiriyaasaa *soquu* magaala deeme.
(b) Caaltuun lafa irraa marga soquu deemte.

The word "soquu" in first sentence can refer to find/looking for/searching/seeking/meeting which is to mean "Jarso went to market to meet with his friend," but the word "Soquu" in the second sentence is clearing or rubbing unwanted things from land, which is to mean "Chaltu went to rub grass from land by hand." Without adding the other words to obtain a clearer definition, the word "soquu" can give a different meaning depending on the context. Therefore, there would be no need to disambiguate the word "soquu," without having the surrounding words as a hint. Given a single word, it is a very difficult to predict the surrounding word in word-sense disambiguation.

As can be seen from the Afaan Oromoo example above, the meaning of the word "soquu" is determined by using the surrounding words in a sense similar to a saying in [5]. To see this, consider instead the phrase "nama soquu" which is "find someone who is humankind" and instead if we see this "lafa soquu" we will get the meaning "clearing the grass from the ground." From this, one can easily understand that context has a big impact for the meaning of a word. They actually define the word token as different meaning.

Enhancing WSD would play a great role in many natural language processing (NLP) problems of machine translation, information retrieval, and information extraction [5–7]. Though many contributions have been made in the area of WSD for different international languages and local languages to solve the issues, it is still an area which requires a great deal of attention, especially in local languages like Afaan Oromo. Most of the Ethiopian languages including Afaan Oromo are underresourced languages. Moreover, the absence of prepared corpus for Afaan Oromo language to disambiguate word sense and unavailability of public resources that can be used for different research works are the major challenges for standardization of Afaan Oromo language. Hence, this study aims to mitigate the stated challenge by developing a model that could generate the order of words that surround a given word and will use automatically annotated data rather than handcrafted resources.

To this end, this study favors word embedding used to represent words for text analysis, usually in the form of real-valued vectors that encode the meaning of words in such a way that words closer together in the vector are expected.

The prime objective of this research study is to identify a suitable technique to disambiguate WSD in underresourced Afaan Oromo language and then design a semisupervised learning-based WSD method using the word2vec word embedding technique.

The designed model aimed to provide a solution to alleviate aforementioned problems by predicting the most similar value for specific window size, size of vocabulary, and number of epochs according to vector space dimension; as a result, the outcome of the training of this model gives the nearly related vector value they had in a large document using word embedding.

The current trend in natural language processing work is the use of word embedding for different tasks. The well-known word embedding pretrained models are word2vec, fastText, and Global Vectors (GloVe). Different researchers have been using these pretrained word embedding techniques for different tasks, but they did not clearly tell the performance differences. As indicated in [6, 7], it is difficult to choose among these techniques. Hence, this study employed the word2vec word embedding technique to simplify a user search for similar context word as word embeddings are distributed representations of text in n-dimensional space.

The study evaluates the approach on two standard datasets, using labeled and unlabeled data based on window size, min-count, and vocabulary size parameter settings. The work contributed to automatic generalization of contextual meaning and prediction of neighbor words based on their contextual meanings from large corpora which the previous works failed to address. Thus, the developed model can play paramount role in standardizing the language and for the development of the language under study and hence is the contribution of this the study.

The study uses an unsupervised approach to learn patterns from the corpus in conjunction with human generated rules to cluster similar contexts of ambiguous words and extract contexts, respectively. The key driver for using semisupervised learning is the fundamental issue with corpus-based learning, specifically the sparseness of the training settings. Therefore, the goal of this research is to merge the two machine learning paradigms into a hybrid strategy known as semisupervised. This strategy utilized a combination of semantic strategies, training methods derived from a corpus, and the availability and reliability of linguistic knowledge to facilitate a comprehensive understanding of how words operate within their respective contexts. This paper also uses a form of word-sense disambiguation that combines the benefits of both machine learning techniques, which could lead to improved results. As there is no annotated corpus for Afaan Oromo, the study was limited to using a manually developed corpus (an unsupervised method, which is suitable when there is a scarcity of training data). Even with a limited training dataset, the unsupervised strategy outperformed other approaches. On the other hand, the supervised and unsupervised hybrid methodswere more comfortable than the

unsupervised method since they integrated a set of rules with machine learning. Therefore, the integration of both methods overcame the problems of each other and improved the performance of word disambiguation specifically for underresourced language like Afaan Oromoo language. Therefore, it is innovative to employ word embeddings for semisupervised word-sense disambiguation for a language with limited resources like Afaan Oromoo.

2. Statement of Problem

Nowadays, due to emerging technology, it is difficult to understand the meaning of words in a given context, difficult to predict the feature of word, difficult to understand the Afaan Oromo word meaning (different cultures and morphological structure of Afaan Oromoo language), and difficult to represent a word in low and unique dimensional representation of vector space [6].

The absence of using word embedding in Ethiopian local language as general and Afaan Oromo in specific by itself has a lot of problems due to information explosions to retrieve and extract knowledge.

Searching for relevant documents from an overwhelming number of documents and absorbing a large quantity of relevant information from the abundant documents are other problems stated in [7]. However, some researchers have worked on the vector space model to solve the problem.

In this study, efforts have been made to answer the following technical questions with the consideration of the issues cited in statement of problems:

(i) What is the most suitable technique to disambiguate WSD in Afaan Oromo language?

(ii) How to evaluate the effectiveness of the WSD model in real-life application?

The following are the paper's main contributions:

(i) Identifying a suitable technique to disambiguate WSD for Afaan Oromo language.

(ii) Designing a semisupervised learning-based WSD method using the word2vec word embedding technique.

(iii) Automatically generalizing contextual meaning and predicting the neighbor words based on their contextual meanings from large corpora.

(iv) Increasing the scope of the word-sense disambiguation research by investigating its applicability for Afaan Oromo language.

Therefore, this work aimed to identify a suitable technique to disambiguate WSD in underresourced Afaan Oromo language by semisupervised learning-based WSD using the word2vec word embedding technique.

3. Related Works

In [8], a word-sense disambiguation method for Polish language is discussed. In the work, the investigators compared two approaches, unsupervised and supervised machine learning. A word embedding model word2vec is used in the unsupervised technique, and the best result is 52% precision. For the supervised technique, the system was evaluated at different epochs and an epoch of 200 has shown a better result of 69%.

The second work was done by Nguyen et al. [9] for Japanese language. In addition to the word embedding technique, a concept embedding is also incorporated to increase the performance of the WSD system. They have used three methods of investigation: word embedding alone, word embedding and concept embedding together, and finally concept embedding only. They have confirmed that the use of concept embedding has improved the performance of the system.

Finally, in the work of Bianchi et al. [10], a technique referred to as zero-shot learning is introduced for the English language. This was a recent work which uses sense embedding and graph embedding (convE). The investigators developed the system by making the system to learn from sense data and dictionary data and have achieved state-of-the-art performance, an $F1$ score of 71.8.

Tesema et al. [11] used a rule-based approach for determining the sense of Afaan Oromo word in a given context. In this work, modifiers are used as an indicator of a sense for a given word in a context. The researcher employed coventional approach on limited datasets which makes their work non-the state-of-the-art appraoch.

Authors in [12], and [13], highlights l. The reason for using the unsupervised approach is to overcome the scarcity of training data. Hence, they used an unsupervised machine learning technique that uses unlabeled data and achieved an accuracy of 81%. They believe that the use of the state-of-the-art technique might improve the performance of Afaan Oromoo word-sense disambiguation.

Another work was done by Mebrahtu Reda [14] for Tigrigna language. In this work, only five (5) ambiguous words were used to test the performance of the word-sense disambiguation system. The investigator has tested his system using five clustering algorithms and achieved a best performance for expectation-maximization algorithm as 67 to 83.3% accuracy. A conventional technique is followed using Weka 3.8.1 tool.

A recent contribution for the Amharic language is the work of Tadesse Birbirso [4], which is based on the WordNet hierarchy. In the work, a knowledge-based approach is preferred to use WordNet as a knowledge base to tackle data scarcity. The investigator used techniques like context glossing and augmented semantic space to disambiguate Amharic language as word level and sentence level. They have achieved an accuracy of 80% and 75% for word level and sentence level, respectively.

Although many works have contributed to solving the problem of word-sense disambiguation at global and local levels, it is still in its infancy for local and underresourced languages like Afaan Oromo. Specially, for the Afaan Oromo language, a limited number of works are available. In the contributions, either less amount of data is used or conventional technique is used. Hence, in the proposed work,

the current state-of-the-art technique known as the word embedding method is used.

4. Research Methodology

This study employed design science research (DSR) to build and evaluate an approach by following design process and to explain how the proposed model has been used as an exceptional job that was the specific method of adapting word embedding and then define the proposed model in which adapted word embeddings are included in a new proposed word disambiguation system. The design science starting from problem identification which is best fit for design science research and used for identifying the core problem and ending with research practical experience by employing quantitative research approach in which quantitative data were taken [15] is cited in Figure 1.

4.1. Data Collection and Analysis. Researchers have collected 456,300 words from different sources (BBC Afaan Oromoo, Bible, legal documents, and previous research studies). Supervised and unsupervised data were employed in this work to do the analysis. Manually annotated supervised data (i.e., small number of labeled data) are taken from the earlier researchers [16], which are proved by language experts and used to compare the earlier works within this study.

The unsupervised data (i.e., unlabeled data) which are generated automatically during train word embedding were employed by the researcher to do the analysis.

4.2. Data Preparation and Data Preprocessing. This study relied on the patterns learned from two corpora: tagged corpora and untagged corpora. In the preparation of the tagged corpus, this study used a manually tagged annotated corpus as an input to train the tagging model from .txt format. And preprocessing components were implemented on the corpus. Those components were sentence splitter, tokenizer, and tag set analyzer. The sentence splitter module splits the document into sentences by using Afaan Oromo's end punctuation marks like ",," "?," "!," and "." The tokenizer module splits the string into words and punctuation marks. Then, they were tagged in the form of "word/tag" at the time of the training phase. The tag set analyzer extracts the tag set from the output of the tokenizer module.

Then, for both corpora, tokenization, stop word removal, and normalization were performed. In the tokenization process, a set of sentences were split into words using white space. After the tokenization process took place, stop words were removed.

As stop words are actually the most common words in any language (articles, prepositions, pronouns, conjunctions, etc.) and do not add much information to the text, removing stop words definitely reduces the size of the dataset and reduces the number of tokens in training, thus reducing training time. These stop words were identified by the help of linguistics experts. Tokenization process and Gensim toolkit of NLP were utilized to remove the stop words for the proposed language.

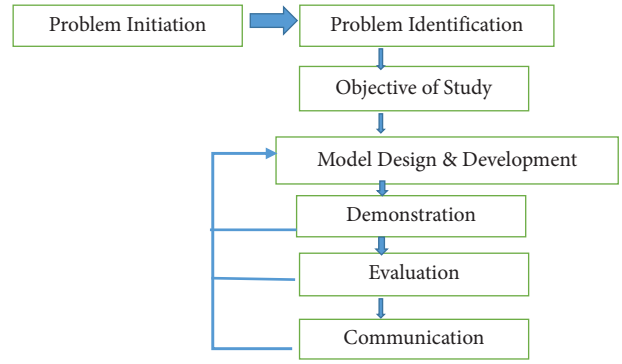

Figure 1: Baseline research design process.

In addition, some characters of the same words might be represented in uppercase or lower case in the corpora as well as in the user input. As a result, we had normalized them into lowercase. The overall data preprocessing techniques employed in this work are explained in Figure 2.

4.3. Training and Testing. As presented in the last section, there is no corpus prepared for underresourced Afaan Oromo language for the disambiguation purpose yet. For this reason, corpus for Afaan Oromo language that contains ambiguous words was prepared manually and labeled by linguistics experts which is taken from earlier researchers [16]. The WSD system was trained and tested with Python programming language on the unannotated and annotated corpus, which constitutes ambiguous words. Researchers selected 20 words with ambiguous meaning which have more than two contextual meanings from identified sources to test the work.

4.4. Implementation Tools. As implementation tool, Anaconda module is used because it is powerful and contains appropriate Python library modules for processing different NLP tasks. From those modules, NLTK was selected since it is an open-source tool like Python modules for linguistic data and documentation for research and development in natural language processing.

4.5. Evaluation. The study evaluates the proposed system with previous work. The evaluations were undertaken on the basis of precision achieved in this work done on 20-word vocabulary.

5. Design and Development of WSD Model

The initial step in the design and development of this model involves preprocessing the text. This includes taking inputs and corpora (both unannotated and annotated), performing tokenization to remove stop words and normalize the text. Next, the model extracts context terms that provide insights into the possible meanings of the ambiguous term. This extraction is done using three parameters: window size, min-count, and vocabulary size. Finally, the model clusters similar context terms together to identify different senses

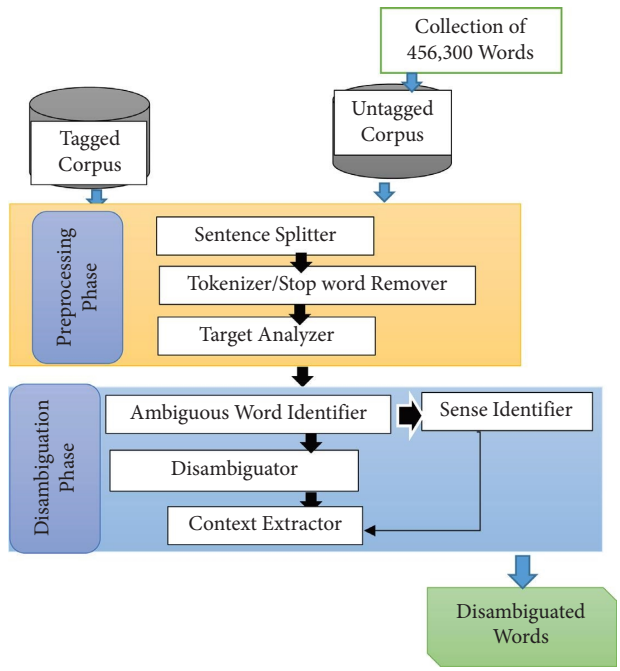

Figure 2: Data preprocessing technique.

```
In [30]: from gensim import models

         tfidf = models.TfidfModel(corpus)
         corpus_tfidf = tfidf[corpus]

In [31]: lsi_model = models.LsiModel(corpus_tfidf, id2word=dictionary, num_topics=2)
         corpus_lsi = lsi_model[corpus_tfidf]

In [32]: lda_model = models.LdaModel(corpus, id2word=dictionary, num_topics=2)
         corpus_lda = lda_model[corpus]

In [33]: for doc, as_text in zip(corpus_lsi, documents):
             print(doc, as_text)
```

```
eraa', 'kennuu', 'irratti', 'hojjechuun', 'hojii', 'xiyyeeffannaa', 'cimaa', 'barbaadu', 'ta', 'mootummaan', 'guddina', 'dina
gdeefi', 'ijaarsa', 'sagantaa', 'sirna', 'aadaa', 'dimookiraasii', 'dagaagsuu', 'keessatti', 'gaheen', 'dubartootaa', 'olaana
a', 'ta', 'uu', 'isaa', 'hubachuusaatiin', 'mirga', 'dubartootaaf', 'xiyyeeffannoo', 'olaanaan', 'akka', 'kennamu', 'kutanno
o', 'qabu', 'mul', 'iseera', 'kana', 'malees', 'mirga', 'hirmaannaafi', 'mirkaneessuun', 'guddina', 'biyyaa', 'galmaan', 'gah
uuf', 'waldaadaaleefi', 'gurmaa', 'anii', 'qaamolee', 'hawaasa', 'adda', 'addaa', 'waliin', 'ta', 'uun', 'ijaarsa', 'saganta
a', 'sirna', 'aadaa', 'diimookiraasii', 'bulchiinsa', 'gaarii', 'nageenyaafi', 'guddina', 'dinagdee', 'mirkaneessuu', 'keessa
tti', 'shoora', 'olaanaa', 'gumaachaa', 'jiru', 'buttaa', 'qaluun', 'aadaa', 'oromoo', 'keessaa', 'isa', 'tokko', 'sagantaa',
'sirna', 'aadaa', 'gadaa', 'waliinis', 'kan', 'geggeeffamaa', 'turedha', 'itoophiyaan', 'bu', 'uura', 'heera', 'mootummaa',
'federaalawaa', 'dimookiraatawaa', 'rippaablika', 'itoophiyaa', 'mfdri', 'bara', 'irratti', 'hundaa', 'uun', 'sagantaa', 'sir
na', 'aadaa', 'bulaa', 'jirti', 'aadaa', 'sirna', 'sagantaa', 'madaallii', 'fi', 'badhaasaa', 'bara', 'marsaa', 'tokkoffaa',
'sadarkaa', 'sadarkaan', 'gaggeeffamuuf', 'haalli', 'qabiinsa', 'ragaa', 'haala', 'gaarii', 'irra', 'jiraachuu', 'isaa', 'aad
aa', 'sirna', 'sagantaa', 'cuufinsaa', 'sanbata', 'darbe', 'tapha', 'waancaa', 'kubbaa', 'miilaa', 'dhiiraa', 'fi', 'gareewwa
n', 'xumuraaf', 'gahan', 'kan', 'taphatan', 'yoo', 'ta', 'tapha', 'waancaa', 'kubbaa', 'miilaa', 'dubartootaatiin', 'godinn
```

Figure 3: Preprocessing model to remove the stop word list.

encoded by the ambiguous term by employing three steps so far. The first step involved cleaning and tokenizing the data. This process typically included converting the text to lowercase, removing non-alphanumeric characters, and eliminating stop words (Figure 3).

The second step focused on generating vector representations of the documents. This entailed mapping the words in the documents to numerical vectors, often accomplished through the use of word embedding techniques.

Lastly, a clustering algorithm was applied to the document vectors in the third step. The goal was to identify optimal groups by employing a clustering method similar to the K-means algorithm (Figure 4). The number of clusters represents the number of senses. To accomplish this clustering, the study calculates the degree of vector similarity by using vectors constructed from co-occurrence information.

6. Experiments and Result

6.1. Development of Training Model. In this section, we follow the steps that presented under methodology section. Several comprehensive experiments (brief demonstration of both tagged and untagged datasets, pre-processing phases, Disambiguating Phases and finally, Word sense disambiguation) are conducted to implements and demonstrates the effectiveness of the proposed model for Afaan Oromoo Language.

```
In [40]: clustering, cluster_labels = mbkmeans_clusters(
             X=vectorized_docs,
             k=20,
             mb=500,
             print_silhouette_values=True,
         )
         df_clusters = pd.DataFrame({
             "text": docs,
             "tokens": [" ".join(text) for text in tokenized_docs],
             "cluster": cluster_labels
         })

         For n_clusters = 20
         Silhouette coefficient: 0.79
         Inertia:2312.7339008984422
         Silhouette values:
              Cluster 7: Size:719 | Avg:1.00 | Min:0.56 | Max: 1.00
              Cluster 16: Size:204 | Avg:0.96 | Min:0.06 | Max: 0.97
              Cluster 17: Size:57 | Avg:0.94 | Min:0.12 | Max: 0.96
              Cluster 8: Size:50 | Avg:0.75 | Min:0.24 | Max: 0.85
              Cluster 4: Size:39 | Avg:0.61 | Min:0.11 | Max: 0.75
              Cluster 9: Size:20 | Avg:0.59 | Min:0.25 | Max: 0.70
              Cluster 10: Size:4 | Avg:0.53 | Min:0.19 | Max: 0.64
              Cluster 15: Size:11 | Avg:0.41 | Min:0.10 | Max: 0.57
              Cluster 6: Size:12 | Avg:0.38 | Min:0.09 | Max: 0.55
              Cluster 14: Size:29 | Avg:0.37 | Min:0.05 | Max: 0.55
              Cluster 3: Size:88 | Avg:0.27 | Min:-0.06 | Max: 0.45
              Cluster 5: Size:21 | Avg:0.23 | Min:0.08 | Max: 0.39
              Cluster 19: Size:24 | Avg:0.21 | Min:0.05 | Max: 0.39
              Cluster 13: Size:25 | Avg:0.16 | Min:-0.04 | Max: 0.37
              Cluster 18: Size:8 | Avg:0.08 | Min:-0.15 | Max: 0.26
```

Figure 4: Clustering similar words by K-means algorithm.

```
In [78]: model = gensim.models.Word2Vec (documents, size=100, window=10, min_count=2, workers=10)
         model.train(documents,total_examples=len(documents),epochs=200)

         2020-01-02 20:52:47,212 : INFO : worker thread finished; awaiting finish of 0 more threads
         2020-01-02 20:52:47,212 : INFO : EPOCH - 199 : training on 458718 raw words (14897 effective words)
         ve words/s
         2020-01-02 20:52:47,231 : INFO : worker thread finished; awaiting finish of 9 more threads

         2020-01-02 20:52:47,238 : INFO : worker thread finished; awaiting finish of 8 more threads
         2020-01-02 20:52:47,241 : INFO : worker thread finished; awaiting finish of 7 more threads
         2020-01-02 20:52:47,250 : INFO : worker thread finished; awaiting finish of 6 more threads
         2020-01-02 20:52:47,253 : INFO : worker thread finished; awaiting finish of 5 more threads
         2020-01-02 20:52:47,255 : INFO : worker thread finished; awaiting finish of 4 more threads
         2020-01-02 20:52:47,256 : INFO : worker thread finished; awaiting finish of 3 more threads
         2020-01-02 20:52:47,258 : INFO : worker thread finished; awaiting finish of 2 more threads
         2020-01-02 20:52:47,259 : INFO : worker thread finished; awaiting finish of 1 more threads
         2020-01-02 20:52:47,278 : INFO : worker thread finished; awaiting finish of 0 more threads
         2020-01-02 20:52:47,281 : INFO : EPOCH - 200 : training on 458718 raw words (14901 effective words)
         ve words/s
         2020-01-02 20:52:47,283 : INFO : training on a 91743600 raw words (2980007 effective words) took 13.
         s/s

Out[78]: (2980007, 91743600)
```

Figure 5: Pretraining all Afaan Oromo corpora to develop model.

```
In [13]: # look up top 6 words similar to 'federaalawaa'
         w1 = ["jaalala"]
         model.wv.most_similar (positive=w1,topn=20)

Out[13]: [('haasawaa', 0.9588596224784851),
          ('waldanda', 0.9583259658813477),
          ('tokkummaa', 0.8893300294876099),
          ('istaadiyoomii', 0.8773956298828125),
          ('fayyummaa', 0.8683531284332275),
          ('dorgomichaa', 0.8108373284339905),
          ('ispoortiin', 0.8063225746154785),
          ('kdj', 0.7997111678123474),
          ('raawwatee', 0.7943683862686157),
          ('kdr', 0.7535940408706665),
          ('ytw', 0.752511739730835),
          ('kd', 0.7408142685890198),
          ('qabsiisuutiin', 0.7305011749267578),
          ('schemarefs', 0.7201000452041626),
          ('gurgurtaa', 0.7194925546646118),
          ('mareen', 0.7182063460350037),
          ('ummatichaa', 0.7181246280670166),
          ('dungoo', 0.716765284538269),
          ('if', 0.7123827934265137),
          ('asallaa', 0.7111687660217285)]
```

```
In [14]: # look up top 6 words similar to 'oromiyaa'
         w1 = ["oromiyaa"]
         model.wv.most_similar (positive=w1,topn=20)

Out[14]: [('naannoo', 0.8775851130485535),
          ('dhimmootni', 0.7922940850257874),
          ('wayitaawoon', 0.7919580936431885),
          ('imaammataafi', 0.7682477831840515),
          ('balballoomsuu', 0.7552952766418457),
          ('biiroo', 0.7528466582298279),
          ('mohaammad', 0.7522608637809753),
          ('guutuu', 0.7093364596366882),
          ('taphni', 0.7073886394500732),
          ('pirezidaantiin', 0.7042554020881653),
          ('erratti', 0.7015513181686401),
          ('lamaaf', 0.698821485042572),
          ('zalaalem', 0.6925062537193298),
          ('ob', 0.6904651522636414),
          ('chaartarii', 0.690310001373291),
          ('biiroleen', 0.6896829605102539),
          ('iftoomina', 0.6888936161994934),
          ('lammiileesaanii', 0.6887475252151489),
          ('torban', 0.6867213845252991),
          ('obbo', 0.6849545240402222)]
```

Figure 6: Sample model for most similar 20 words with vocabulary.

```
In [5]: model = gensim.models.Word2Vec (documents, vector_size=100, window=10, min_count=2, workers=10)
        model.train(documents,total_examples=len(documents),epochs=200)

Out[5]: (2979991, 91743600)

In [6]: model.wv.save_word2vec_format('tabii.model', binary=False)

In [7]: import matplotlib.pyplot as plt
```

Figure 7: Training model to evaluate the vectorized tokens.

```
In [14]: # look up top 6 words similar to 'oromiyaa'
         w1 = ["oromiyaa"]
         model.wv.most_similar (positive=w1,topn=20)

Out[14]: [('naannoo', 0.8775851130485535),
          ('dhimmootni', 0.7922940850257874),
          ('wayitaawoon', 0.7919580936431885),
          ('imaammataafi', 0.7682477831840515),
          ('balballoomsuu', 0.7552952766418457),
          ('biiroo', 0.7528466582298279),
          ('mohaammad', 0.7522608637809753),
          ('guutuu', 0.7093364596366882),
          ('taphni', 0.7073886394500732),
          ('pirezidaantiin', 0.7042554020881653),
          ('erratti', 0.7015513181686401),
          ('lamaaf', 0.698821485042572),
          ('zalaalem', 0.6925062537193298),
          ('ob', 0.6904651522636414),
          ('chaartarii', 0.690310001373291),
          ('biiroleen', 0.6896829605102539),
          ('iftoomina', 0.6888936161994934),
          ('lammiileesaanii', 0.6887475252151489),
          ('torban', 0.6867213845252991),
          ('obbo', 0.6849545240402222)]
```

Figure 8: Vocabulary word of Oromiyaa.

```
training_loss = model_with_loss.get_latest_training_loss()
print(training_loss)

966111.0625

In [47]: from sklearn import metrics
         from sklearn.cluster import KMeans


         labels = kmeans.labels_
         metrics.silhouette_score(X,labels)

Out[47]: 0.8070709071684881

In [48]: metrics.calinski_harabasz_score(X, labels)

Out[48]: 4695.1203824291815

In [49]: from sklearn.metrics import davies_bouldin_score
         davies_bouldin_score(X, labels)

Out[49]: 0.9770058844293829

In [50]: print(kmeans)

         KMeans(n_clusters=20, random_state=37)
```

Figure 9: *K*-means algorithm evaluations.

```
In [47]: from sklearn.metrics import confusion_matrix, classification_report
Y_pred_bi = model.predict_classes(X_test,batch_size = batch_size)
df_test = pd.DataFrame({'true': y_test.tolist(), 'pred':Y_pred_bi})
df_test['true'] = df_test['true'].apply(lambda x: np.argmax(x))
print("confusion matrix",confusion_matrix(df_test.true, df_test.pred))
print(classification_report(df_test.true, df_test.pred))
scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

```
confusion matrix [[1170   41    9]
 [   6 4851   10]
 [   5   18 1890]]
              precision    recall  f1-score   support

           0       0.99      0.96      0.97      1220
           1       0.99      1.00      0.99      4867
           2       0.99      0.99      0.99      1913

    accuracy                           0.99      8000
   macro avg       0.99      0.98      0.99      8000
weighted avg       0.99      0.99      0.99      8000

Accuracy: 98.89%
```

Figure 10: Performance evaluation by using confusion matrix.

```
plt.legend(['train_accuracy','validation_accuracy'], loc = 'lower right')
plt.show()

plt.plot(history6.history['loss'])
plt.plot(history6.history['val_loss'])

plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train_loss','validation_loss'], loc = 'upper right')
plt.show()
```

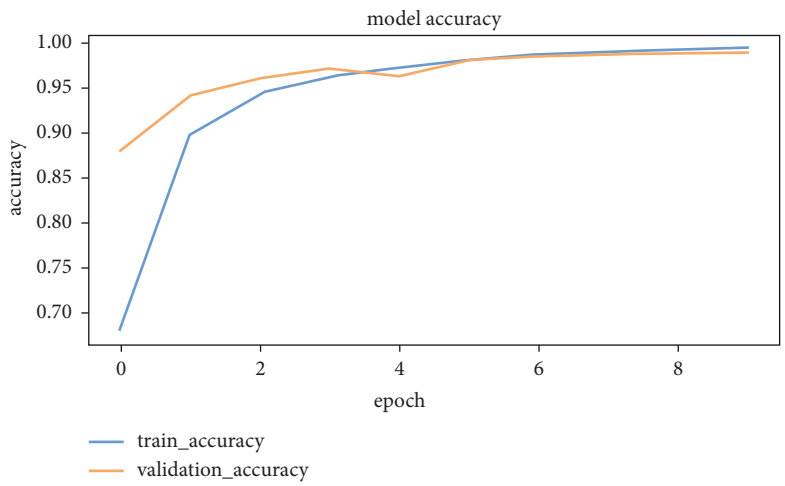

Figure 11: Train accuracy plotting to validate the testing data.

To accomplish this context, the study focused on utilizing the Genism implementation of Word2Vec and effectively putting it into action to enhance performance. This was accomplished through a combination of two factors: (1) the input data and (2) specific parameter values. The input data consisted of documents, with a total size of 91,743,600. The parameter values used were window = 10, min_count = 2, and workers = 10. (Figure 5).

Table 1: Summary of comparison with similar systems.

No	Title	Used algorithm and techniques	Reported results	References
1	Use of part of speech tagging for Afaan Oromo word sense modeling	EM and *K*-means and one to three context window size	ML approach with EM algorithm achieved 74.85% for annotated corpus and 70.35% for unannotated one. Hybrid approach with *K*-means algorithm scored 79.1% for annotated corpus and 74.85% for unannotated corpus.	[16]
2	Towards the sense disambiguation of Afan Oromo words using hybrid Approach (unsupervised machine learning and rule based)	Unsupervised machine learning Approach with *K*-means and EM clustering algorithms	The result argued that WSD yields an accuracy of 56.2% in unsupervised machine learning and 65.5% in hybrid approach	[6]
3	Amharic word sense disambiguation using wordnet	WordNet dictionary to make disambiguation solved contextual words only with small data. By applying Lesk algorithm	For Amharic WordNet with morphological analyzer and Amharic WordNet without morphological analyzer, they achieved an accuracy of 57.5% and 80%, respectively	[17]
4	Word sense disambiguation for Afaan Oromo language	Supervised machine learning techniques are applied to a corpus of Afaan Oromo language, to acquire disambiguation information automatically. It also applied Naïve Bayes theorem to find the prior probability and likelihood ratio of the sense in the given context.	For annotated corpus and 79.5% for unannotated 74.67% accuracy respectively	[5]
5	Word sense disambiguation using hybrid swarm intelligence approach	Partitional clustering (EM and *K*-means) algorithm was employed	The achieved result was encouraging; despite it is less resource requirement. The system yielded an accuracy of 76.05% for the unsupervised approach and 89.47% for the hybrid approach, respectively.	[18]

The screenshot showcases a collection of 20 words that share similar meanings and context with the terms "Jaalala" (Love) and "Oromiyaa" (Region) vocabulary by ignoring binary weight is presented in Figure 6.

During training the model, the study used the following parameters:

(i) Sentences: expect a list of lists with the tokenized documents.

(ii) Vector size: Defines the size of the word vectors. In this case, it used 100.

(iii) Workers: Define how many cores you use for training. We set it to 1 to make sure the code is deterministically reproducible (Figure 7).

To validate the trained model, this study used the most common vocabulary which has been mentioned earlier and clustered. When it defines the Vocabulary Oromiyaa, it is one part of Region in Ethiopia and also the meaning of Naannoo is a region. Therefore, it has a meaning of Naannoo Oromiyyaa [Oromia Region] as explained in Figure 8, and finally, the study evaluates the word2vec (embedding) with the *K*-means algorithm (Figure 9).

7. Results and Discussion

The study used word embedding with semi-supervised technique for Afaan Oromo that was to investigate the consequence on the Word Sense Modeling. The words have efficient and effective rather than using an annotated with supervised technique.

To assess the effectiveness of the study, the researchers employed K-fold cross-validation with 10 folds and applied the k-means algorithm. Evaluation of the research was conducted based on the significance of context, resulting in the highest F1-measure, recall, and precision scores, all reaching 99%. The overall performance of the model was analyzed using a confusion matrix, revealing an accuracy rate of 98.89% for the proposed model (Figure 10).

To evaluate the training model, the study has used a confusion matrix that can be used to evaluate the performance of a classification model. At this point, it is used to evaluate data based on their number of target classes. As we have observed, the value of matrix compares the actual target value with the value predicted by the model. The confusion matrix image results summarize the training data and testing data and validate them. The number of correct and incorrect predictions is summarized by count value and grouped by class. This is the key to the confusion matrix (Figure 11).

The researchers investigated the performance of these models using *F*-measure measurements. It measures the meaning of a selected vocabulary from a sentence or a large corpus. It is the most powerful method to predict the neighbor words from the given meaning based on the given vocabulary. Furthermore, the researchers conducted a comparison with similar systems to determine whether the newly proposed system surpassed existing methods as discussed somewhere in Table 1. The comparison of similar systems was done, and the proposed system outperformed the existing methods with promising result.

8. Conclusions

The overall focus of this study is to identify a suitable technique to disambiguate WSD in underresourced Afaan Oromo language and then design a semisupervised learning-based WSD method using the word2vec word embedding technique that addresses the problem of deciding the correct sense of an ambiguous word based on its context and word prediction.

To conduct the study, 456,300 words of Afaan Oromo words were taken from different sources and preprocessed for experiment by Natural Language Toolkit and Anaconda tool. The study has used 20 ambiguous words which have 3-4 senses to test the model. NLTK and Python programming language were used to train the corpus.

This study explores various methods of incorporating semantic knowledge from word embedding into the model for word sense disambiguation. It conducts a thorough analysis of different parameters and strategies across multiple WSD tasks, leading to three key findings.First, word embedding can be employed as novel features to enhance the performance of state-of-the-art word disambiguation methods. Second, integrating embedding within the framework of WSD proves to be more reliable and yields higher performance compared to other approaches, such as vector modeling (using cosine similarity) and dimensional analysis. Lastly, utilizing word embedding in conjunction with the K-means algorithm to cluster word vocabulary, leveraging insights from Semi supervised datasets, can surpass the performance of state-of-the-art supervised/unsupervised models that rely on traditional WSD features.Overall, this study demonstrates the efficacy of integrating semantic knowledge from word embedding in WSD models, highlighting its potential to improve accuracy and outperform existing approaches.

Experimental results show that using word embedding for both labeled corpus and unlabeled corpus improved the performance of the system by total accuracy of 98.89%. Although the best performance is obtained when standard WSD features are augmented with the additional knowledge from word2vec vectors on the basis of a WSD, the study hopes that this work will serve as the first step for further studies on redesigning standard WSD features for Afaan Oromo.

9. Future Works

As a future work, in order to advance the existing model, the researchers recommend to work on the advancement of semisupervised approach and other state-of-the-art systems like sense and concept embedding.

Data Availability

The datasets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] D. Edward Crummey, "Britannica," 2022, https://www.britannica.com/place/Ethiopia/Soils#ref37685.

[2] B. Erana, *Oromo Language (Afaan Oromoo)*, Harvard University, Cambridge, MA, USA, 2022.

[3] T. Degeneh Bijiga, *The Development of Oromo Writing System*, University of Kent, Canterbury, UK, 2015.

[4] D. Tadesse Birbirso, "The power of afaan Oromo as a device for explaining africa's prehistory: an africology perspective," *East African Journal of Social Sciences and Humanities*, vol. 4, no. 1, pp. 73–90, 2019.

[5] T. Kebede Hundesa, *Word Sense Disambigation for Afaan Oromo Language*, Addis Abab University, Addis Ababa, Ethiopia, 2015.

[6] W. Tesema, D. Tesfaye, and T. Kibebew, "Towards the sense disambiguation of afan Oromo words using hybrid approach(unsupervised machine learning and rule based)," *Ethiopian Journal of Education and Science*, vol. 12, no. 1, pp. 61–77, 2016.

[7] D. Oele and G. Van Noord, "Simple embedding-based word sense disambiguation," in *Proceedings of the 9th Global Wordnet Conference*, Singapore, January, 2018.

[8] P. Kędzia, M. Piasecki, and M. Orlińska, "Word sense disambiguation based on large scale Polish CLARIN heterogeneous lexical resources," *Cognitive Studies | Études cognitives*, vol. 29, no. 15, pp. 269–292, 2015.

[9] Q.-P. Nguyen, A.-D. Vo, J.-C. Shin, and C.-Y. Ock, "Effect of word sense disambiguation on neural machine translation: a case study in Korean," *IEEE Access*, vol. 6, pp. 38512–38523, 2018.

[10] F. Bianchi, S. Terragni, D. Hovy, D. Nozza, and E. Fersini, "Cross-lingual contextualized topic models with zero-shot learning," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, Europea, April, 2021.

[11] W. Tesema, D. Tesfaye, and T. Kibebew, "Designing a rule based disambiguator for afan Oromo words," *American Journal of Computer Science and Information Technology*, vol. 05, no. 02, pp. 147–156, 2017.

[12] P. Ahmad, P. Ali, S. Matwin, and M. Sokolova, *One Single Deep Bidirectional LSTM Network for Word Sense Disambiguation of Text Data*, Cornell University, Ithaca, NY, USA, 2018.

[13] A. Pai, *An Essential Guide to Pretrained Word Embeddings for NLP Practitioners*, Analytics Vidhya, Haryana, India, 2020.

[14] M. Mebrahtu Reda, "Unsupervised machine learning approach for Tigrigna word sense disambiguation," *Computer Engineering and Intelligent Systems*, vol. 9, no. 6, pp. 10–16, 2018.

[15] K. Taghipour and H. Tou Ng, "Semi-supervised word sense disambiguation using word embeddings in general and specific domains," in *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, Denver, Colorado, June, 2015.

[16] L. Daniel, *Use of Part of Speech Tagging for Afaan Oromo Word Sense Modeling*, Addis Ababa University, Addis Ababa, Ethiopia, 2019.

[17] S. M. Dereje, T. Y. Tesfa, and W. T. Yitbarek, "Sentence level Amharic word sense disambiguation," *American Journal of Education and Technology*, vol. 1, no. 2, pp. 83–87, 2022.

[18] W. Al-Saiagh, S. Tiun, A. Al-Saffar, S. Awang, and A. S. Alkhaleefa, "Word sense disambiguation using hybrid swarm intelligence approach," *Plose One*, vol. 13, no. 12, 2018.