I.J. Information Engineering and Electronic Business, **2023, 1, 41-50**
Published Online on February 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2023.01.04

Enhanced Word Sense Disambiguation Algorithm for Afaan Oromoo

Abdo Ababor Abafogi
Department of Information Technology, College of Computing and Informatics, Wolkite University, Ethiopia
Email: abdo.ababor@wku.edu.et/abdosu2013@gmail.com

Received: 10 March, 2022; Revised: 18 May, 2022; Accepted: 14 July, 2022; Published: 08 February, 2023

Abstract: In various circumstances, the same word can mean differently based on the usage of the word in a particular sentence. The aim of word sense disambiguation (WSD) is to precisely understand the meaning of a word in particular usage. WSD utilized in several applications of natural language to interpret an ambiguous word contextually. This paper enhances a statistical algorithm proposed by Abdo [36] that performs a task of WSD for Afaan Oromoo (one of under-resourced language spoken in East Africa by nearly 50% of Ethiopians). The paper evaluates appropriate methods that used to increase the performance of disambiguation for the language with and without morphology consideration. The algorithm evaluated by 249 sentences with four evaluation metrics: recall, precision, F1 and accuracy. The evaluation result has achieved state of the art for Afaan Oromoo. Finally, future direction is highlighted for further research of the task on the language.

Index Terms: Afaan Oromoo, Word Sense Disambiguation, Unsupervised Approach, Adopted Algorithm for Sense Clustering

1. Introduction

In human language, regularly a word is utilized in numerous ways. The same word often interpreted differently based on its utilization in a particular context. The context of a word defines lots about its meaning. Thus, in natural language processing (NLP) task, dealing with textual content information, we need a way to interpret a word contextually, the same word with specific meanings. Moreover, words cannot be divided into discrete meanings. Words regularly have similar meanings or plenty of unrelated meanings, which causes a notable challenge in NLP applications.

WSD is a field of NLP that targets at figuring out the right sense of vague word in a specific context [1]. Interpreting a word in a particular context is very critical for NLP due to the word ambiguity, richness of natural languages and so on [2]. The intention of WSD is to exactly recognize the meaning of a word especially utilization for the appropriate labeling of that word.

WSD has numerous applications in various textual content processing and NLP fields. WSD can be used in area of information extraction, text mining, machine translation, and IR and so on. In IR system to execute queries within the query context instead of understand query as a key-word only, and to judge the relevance of documents in contextual meaning. In machine translation to provide contextual meaning we need WSD. Consequently, disambiguation is the most crucial task and also challenging at all levels of the natural languages, in particular for under-resourced languages the challenge is desperate.

Afaan Oromoo is one of the under-resourced languages spoken by the largest ethnic group in Ethiopia, which amounts nearly 50% of the Ethiopian population [3-5]. Afaan Oromoo writing system, is called “Qubee” a Latin-based alphabet adopted and grow to be the legitimate script since 1991 [5-7]. Afaan Oromoo consists 33 basic letters, grouped into vowels, consonants, and paired consonant (i.e. ‘dh’, ‘ny, etc.) letters [8].

The word, in Afaan Oromoo named “jecha” is the smallest unit of the language. Blank space commonly indicate the boundary of a word. Also parenthesis, brackets, and quotes indicate a word boundary [9]. Afaan Oromoo sentence structuring is different from English sentences structure. Afaan Oromoo uses subject-object-verb (SOV) language [8]. Punctuation marks are used within the same manner and for the same purpose as used in English; except an apostrophe. An apostrophe (’) in English indicates possession however in Afaan Oromoo called “hudhaa” that is part of a word that assumes to symbolize missed consonants in a word. For instance “jaha” is written as “ja’a” to mean six. It plays a vital role in Afaan Oromoo writing and reading system [8].

Unavailability of resources like corpus and knowledge-base and the language complexity and morphology richness are the key challenges yet for Afaan Oromoo language processing. In Afaan Oromoo almost all grammatical

information is carried through affixes attached to the stem of words. Specifically, suffixes are the largest morphological features while prefixes and infixes are also in the language. A word behaves differently when joint with a suffix gives a different sense [10].

The research of WSD for Afaan Oromoo language is sort of at the start. In Afaan Oromoo, vague word might occur in numerous morphological forms. Hence, identifying vague words and assigning the correct sense relies on morphology consideration. A morphology consideration plays a vital role in reducing the different forms of vague word into their root forms. In Afaan Oromoo, word's sense rely on the word that preceded by the target word [3, 4]. Ambiguity can happen at numerous levels of Afaan Oromoo language such as syntactic, lexical, pragmatic, and semantic etc. [11]. Furthermore, I confronted a remarkable challenge due to Afaan Oromoo has a useful resource lack. As precise WSD is very critical for NLP application, state of the art for the language needs improvement and more researches on the area. Therefore, to address the challenge, alternate solution suggests that unsupervised sense disambiguation [12].

This paper aims at enhancing an algorithm to outperform the task of WSD for the Afaan Oromoo language. Moreover, a language showing alike patterns with Afaan Oromoo can use the algorithm. Precisely, it will increase the technique of the WSD studies that may be prolonged for semantic textual content similarity.

In this research, an algorithm is adopted to get contextual clues of actual vague word sense, from dataset. The algorithm begins through preprocessing (i.e. tokenization, stopwords removal, stemming) and recognizing vague words, analyzing contextual information of word's sense, pairing all words in the sentence with any available vague word and with their normalized proximity degree. The computed proximity degrees with their sense-specific frequencies are grouped into a sense-specific cluster. The cluster provides information about ambiguity and their discrimination degree. This is the ultimate step in a training dataset. Actually, in a testing phase, the same steps are implemented up to pairing a word with their weighted proximity degree. Then, searching pairs from sense cluster, calculates context overlap with all respective sense clusters. Eventually, calculating overlapped pair and the finest cluster scores determines a sense of the target word. Furthermore, the algorithm plays a widespread function in improving the computer's ability to perform WSD with morphology consideration.

2. Related Work

People are pretty in figuring out the precise sense of a word, however for machines it is very hard. The research on automatic WSD was critical problem since its emergence due to its coverage in wide range applications. WSD is a vital task at several levels of various NLP application. As a result, numerous systems proposed and demonstrated on standard data which might be specialized in WSD evaluation [13]. I reviewed numerous pieces of literatures on diverse Afaan Oromoo NLP tasks and concluded that yet a considerable gap exists in the Afaan Oromoo NLP compared to that in English NLP.

In WSD history, various algorithms were proposed, which classified as a knowledge-based (KB), unsupervised, and supervised approach. KB approach includes linguistic information assets like dictionaries, thesauri, Wordnet, and rule bases. A Supervised approach depends upon annotated data where unsupervised strategies use free text to learn from.

In KB methods algorithms like random walk, Lesks, and Walker's do a machine-readable dictionary lookup to perform WSD task. According to Yarowsky [14] word collocations are crucial in determining lexical ambiguity meaning from statistical information perspective. In neural network era dominates many state-of-the-art in various NLP task, a multi-languages KB approach proposed as in [15] that outperformed state-of-the-art of WSD both in the English language and multilingual. To reduce the load of manually-tagging corpora information from lexical-semantic in Wikipedia and BabelNet can be used as in [15].

Supervised methods are available from pure supervised [16, 17] to hybrid supervised and KB approaches [2, 18-23]. Generally, the supervised WSD concerns purely data-driven models [16], supervised models using glosses [2], supervised models using relations in a KB like that of WordNet hypernymy and hyponymy relations [24], and supervised approaches using knowledge like Wikipedia, Web search and so on [23]. In recent years, many supervised systems were proposed to improve WSD tasks. The best supervise approaches is the one relies on neural networks [25].

In WSD process, supervised approaches give better result than unsupervised one [26]. It was the best approach to WSD through all English datasets [24, 27, 28]. However, recently the approaches were surpassed by KB approach [23]. Unavailability of standard corpora and insufficient corpora size affect the approach.

Unsupervised WSD and knowledge-based WSD are evolving as a remarkable choice to overcome annotated corpora limitation of supervised systems and to improve performance [29]. For less-resourced language supervised approach is challenging, because of unavailability of huge tagged corpora. In this case, unsupervised methods come up with the solution which combines the advantages of both supervised and KB approaches. It gathers statistical information from occurrence and frequency of a word from a corpus and without tagging a corpus [26].

Unsupervised techniques pose the big challenge to researchers. These strategies basis on assumption that similar senses or meanings occur in a similar setting that used to identify vague word from the neighboring words. Prepares the

clusters from emerged information of word occurrences of unlabeled dataset to be trained, after training it predict for an vague words [30, 31].

Unsupervised learning identifies styles in a big pattern of information, without the advantage of any sources of manually-tagged or external knowledge. These styles are used to divide the information into clusters, in which all member of the cluster has many shared features than a member of another clusters. The different techniques of this approach are word clustering, context clustering, and k-means clustering where various and massive dataset available [30]. In general two methods of unsupervised approaches are affiliation rules and clustering that are applied for ambiguity.

Specially, the research of WSD for Afaan Oromoo language is sort of at the start. Tesfa carried out a Naïve Baye's theory to disambiguate 5 Afaan Oromoo ambiguity words [19]. The work was trained on 1116 sentences and tested 124 sentences. The author recommended that four both sides (right and left) of the targeted vague word window size is sufficient for disambiguation task in Afaan Oromoo.

Likewise, Shibiru in [32] conducted WSD research using knowledge base on a window size of one to five right side and left of the ambiguous one. He conducted using Wordnet that developed for Afaan Oromoo with morphological and without morphological analyzer. The experimental valuation performed on fifty sentences then he suggested that ±3 (left and right) windows size was ample for Afaan Oromoo WSD with morphology consideration.

On other hand, Yehuwalashet [34] proposed a hybrid (of rule-based and unsupervised ML) approaches to solve the problem of Afaan Oromoo WSD. The vague word is defined via the built vector representations of word co-occurrences and relies on rules to extract the neighbor of the vague word. Hence, she computed the cosine similarity of the context vectors and evaluated on 20 vague words. The evaluation utilized the same window size across all clustering algorithms (K-Means, EM, Complete link, Average Link and Single link). Finally, she decided that window sizes of one and two in both sides achieved ample result via K-means algorithms but in accuracy EM algorithm was outperformed.

Moreover, hybrid approach was utilized in [30] in which an unsupervised ML supported by handcrafted rule to cluster the contexts and to mine the vague word modifiers. The researchers used the same algorithms as of [34] and both researches were suggested that cosine similarity of window size of two left and right of the target word yielded better accuracy, especially on the EM algorithm [4].

From the discussed related works of Afaan Oromoo, utilizing different datasets, different context window size, and different numbers of vague words have seen as a gap and challenges. According to ref. [33] window sizes one of left and right and also window size two of left and right achieve satisfactory result, particularly size ±1. Similarly, the finding of ref. [4] and [34] shows widow of sizes two of both left and right was suggested for the language. In another research window size three in both sides was recommended in ref. [32]. The last finding is context size 4 of both sides suggested in ref. [35] for Afaan Oromoo. Generally, there is no agreement on window size which is emerged from the standard datasets unavailability, particularly in performance evaluation.

In general, in Afaan Oromoo literature of WSD no agreement made on window sizes. Consequently, Abdo [12] proposed a novel algorithm to solve this problem on small size dataset without text operation like stopword removal, stemming, etc. Indeed, additional experimentation is needed aiming promising concern of reducing window sizes related gap and improving disambiguation effectiveness of Abdo's algorithm.

3. Methodology

This section describes the methods of Afaan Oromoo WSD regarding datasets for the training and testing, text processing, an architecture, targeted word identification, generating co-occurrence and calculating proximity and frequency, sense clustering, and disambiguation algorithm.

There is no annotated corpus publicly available for Afaan Oromoo. However, a few researchers have been prepared small datasets for disambiguation tasks even though it's not available publically. In this, research a dataset prepared in [12] that contains 747 sentences amongst 20 sentences hasn't targeted vague words used in testing. In addition, 1,743 sentences were added that were collected from 3 Afaan Oromoo's news websites (BBC Afaan Oromoo, VOA Afaan Oromoo, and Fana Afaan Oromoo). Unlabeled corpus size of 2490 sentences used for training and testing a target word disambiguation. In the corpus, each sentence is unique and contains at least one vague word except the mentioned 20 sentences. For training 2,241 sentences are used and 249 sentences used for testing. This paper focuses on 30 ambiguous words that frequently occur in Afaan Oromoo texts.

The architecture shown in Fig. 1 is similar to the architecture used in [12] with few modifications such as morphology consideration and dataset in sense disambiguation. Morphology consideration plays a significant role towards efficient NLP for highly inflectional languages like Afaan Oromoo [32]. Preprocessing like tokenization, stop-word removal and stemming performed on Afaan Oromoo texts as on Fig. 1. The next tasks are identifying the respective ambiguous words and clustering a text based on word's sense similarity then generating sense specific word co-occurrence based on word proximity and word frequency. Description of Fig. 1 is presented in the following section.

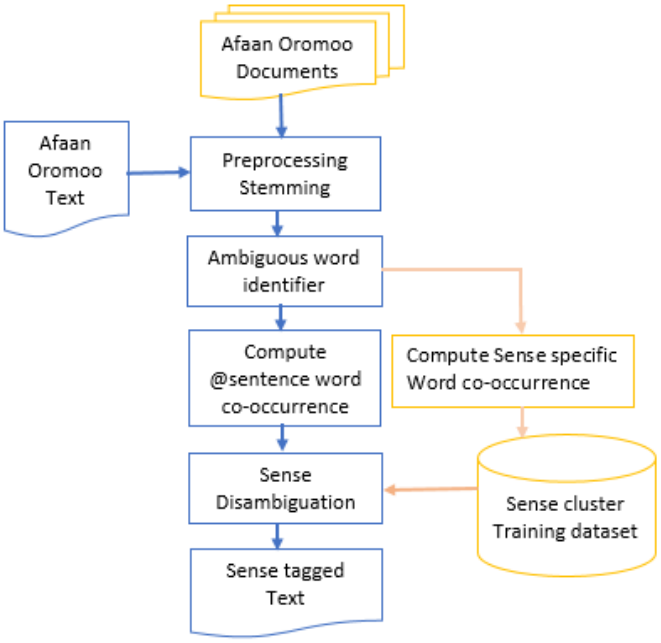

Fig. 1. Afaan Oromoo WSD Architecture

3.1. Preprocessing

In preprocessing step, a text is segmented into sentences based on the punctuation marks such as a period (.), exclamation mark (!) and question mark (?) [36]. Tokenization is a language-dependent and a method of word boundary identification based on a using blank space, parenthesis, brackets, and quotes are show a word boundary. After tokenization any punctuation marks attached at end of a token will removed. An apostrophe (') named "hudhaa" is not removed. As recommended in [12] removing stopword and performing stemming can boost the performance of Afaan Oromoo WSD using this architecture. Based on this idea, 110 manually collected stop words are removed from the datasets.

Once tokenized, and stopword removed, stemming is performed to minimize the variation of a word. The rule-based Afaan Oromoo automatic stemmer (stem++) developed by Daraje [37] has been used in this research. The accuracy of the stem++ is about 95.9% and its implementation code is in python. The stemmer works best except for a few foreseen errors of under-stem. The author modified and implemented (in java code) his rule to handle thus detected errors to the best of my knowledge.

3.2. Ambiguous Word Identification

Once a text tokenized and stemmed, the searching and identifying target word performed on a tokenized target sentence (training/test dataset). If it found in the current sentence, every words in that sentence engaged in generating pair word and engaged in weighting its degree.

3.3. Generating Word Co-occurrence

In Afaan Oromoo WSD literature many authors recommended different windows size from window size ±1 to ±4 sizes [3, 4, 32-35]. To address window size disagreement gap Abdo [12] was proposed a novel algorithm using unsupervised ML. The equation from 1-4 are taken from ref. [12] and then adopted to enhance effectiveness of disambiguation. Furthermore, size of a dataset and a target word increased, removing stopword and stemming performed to evaluate the algorithm effectiveness with stem and without stem as recommended by the author.

After vague word identified, generating pair word of *wa* (a target ambiguous word) with *wj* (every words in a text as *w1, w2,...wn*) considering their distance. Computing actual distance for pair word depends on their index difference as $wa_{\text{index}} - wi_{\text{index}}$ and normalized by (1). For instant, "*Keeniyaatti balaa* ***lolaatiin*** *lubbuun namoota 200* ***darbe***". There are two vague words, which is shown in bold. Table 1 and Fig. 2 illustrate the computation (1) and (2) just on actual (non-stem) words, but the implementation use stem words.

$$dist = log(0.27/(wa_{\text{index}} - wj_{\text{index}})) \quad (1)$$

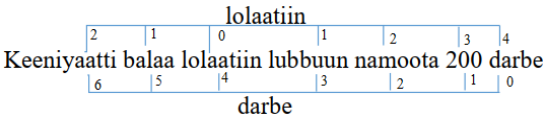

Fig. 2. Sample word co-occurrence generated with their distance

The real distance of every word from the target vague words ("lolaatiin" and "darbe") is shown in Fig. 2. However, from (1) as illustrated in Table 1 it gives negative number. Equation (2) is used to compute weighted pair-wise co-occurrence degree. However, ambiguous word with itself excluded in implementation.

$$Pair(wa, wi) = abs(2.804 * dist * 2.7^{dist}) \quad (2)$$

Where *abs* is an absolute value that removes sign from number. The equation calculates the nearness of words in pairs, at each occurrence in each sentence. Equation (2), computes sentence wise proximity of paired words that define the sense of wa contextually. In (2), the minimum and maximum result is 0 and 1 respectively for all pair.

Table 1. Sample of word co-occurrence and weighting

***wi*→ tokenized sentence**	**wa→Ambiguous words with their weighted co-occurrence degree**			
	lolaatiin		Darbe	
	$dist$	$Pair(wa, wi)$	$dist$	$Pair(wa, wi)$
Keeniyaatti	-2.0025	0.7683	-3.1011	0.3996
Balaa	-1.3093	1.0001	-2.9188	0.4507
Lolaatiin			-2.6956	0.5196
Lubbuun	-1.3093	1.0001	-2.4079	0.6176
Namoota	-2.0025	0.7683	-2.0025	0.7683
200	-2.4079	0.6176	-1.3093	1.0001
Darbe	-2.6956	0.5196		

3.4. Clustering Similar Sense Words

In the training dataset, a sentences having similar sense are grouped together to produce sense-specific word co-occurrence beside their frequency. To disambiguate a word it needs statistical information of words co-occurrence in every sense of vague word computed by (3). It focuses on pairing (clustering) words in the sentence from the sense-specific dataset.

$$SenseCluster_{(wa,wi)} = \frac{\sum abs(2.804 * dist * 2.7^{dist})}{fr(wa,wi)} \quad (3)$$

The distance of *wi* and *wa* are not equally distributed across sentences. $\sum$ to mean summation of pair word (wa and wi) co-occurrence and $fr(wa, wi)$ represent count of pair that occurred in the sense-specific dataset, used to calculate its average. All pair are gathered per their respective sense group to discriminate a word's sense for disambiguation purposes. For all sense of target vague word it trained in similar fashion for sense lookup.

Algorithm for sense clustering requires creating a folder for the training data. Create a file named from the stemmed target word and name of a respective vague word separated by an underscore. Each file contains an average of 29 sentences per sense and each sentence instigated on a new line.

Algorithm for Sense clustering

Open the folder
  Read a file list
    Read a sentences then tokenize it
    Generate co-occurrence via eq.(2)
  Computes via eq.(3)
  Write pair word with value to file and move to next
End of clustering algorithm

3.5. Sense Disambiguation

Afaan Oromoo text is feed to the system. Preprocessing starts by splitting a target text into a list of sentences then after tokenizing it, performs stopwords removal, and stemming successively. Once preprocessing is done, it starts searching for target vague words in the sentence. If target word is unavailable in that sentence, it moves to the subsequent sentence and similarly search until the end of a text. If a sentence has target word, weighted each word with target vague word co-occurrence computed via (3).

Once pair word proximity and frequency statistics is computed via (3), two weightings is taken into account χ for the target sentence and γ for a dataset as in (4). Co-occurrence degree of every pair word of target sentence, is multiplied by χ to reduce the ambiguity noise. Likewise, for every pair word in sense-specific dataset, their occurrence degree is multiplied by γ to reinforce sense dependency that computed as (4).

$$Sense_{wa}\ sent_j = \sum(\chi Sent_{j_{(wa,wi)}} + \gamma Sense_{Cluster_{(wa,wi)}}) \quad (4)$$

Equation (4) disambiguates sense of ambiguous word (wa) in sentence j by adding value of pair word (from the target sentence and training dataset). Pair word represented by wa, wi to mean vague word with all words $(w1, w2, \ldots wn)$ in text respectively. The first task is generating pair word of wa with wi in $Sent_j$ and computing their co-occurrence degree which multiplied by χ as denoted $\chi(Sent_{j_{(wa,wi)}})$ single pair at a time. The second task is $\gamma(Sense_{Cluster_{(wa,wi)}})$ which implies taking a co-occurrence value of target word wa with wi from sense-cluster dataset then normalizing by γ.

Much experimentation were conducted and the detailed description will be presented under section discussion. Lastly, the ∑ (summation) computes all pair words of the target sentence with a respective sense cluster. The maximum ∑ yield decides sense of target vague word in sentence j.

Algorithm for Sense Disambiguation

Read a text then splits into a list of sentences
- If contains *wa* tokenize it
  - Take *wa* position
    - Generate a pair word (wa, wi)
      - Take each pair recursively
        - ts←multiply each pair by χ
        - sc←read pair word from a dataset *di* and multiply by γ
        - *sum*=ts+sc
        - *di*+=sum
  - Takes the greatest yield of *di*
  - Decides its sense
- Else go to next sentence

End of the algorithm

4. Experimental Results and Discussion

In this research an experiment conducted on 30 ambiguous words that frequently occur in Afaan Oromoo. In these experiments, the researcher evaluates performance of the adopted algorithm on a dataset size of 2,490 sentences splitting into a training set size of 2,241 sentences and a testing set size of 249 sentences. Finally, the implemented algorithm take input from user then tag a sense as shown on Fig. 3. The system preprocesses the text, then searches vague word in the text, if detected it generates co-occurrence. Next, it calculates via (4) and provides sense tagged output.

The main focus of this work is evaluating the effect of dataset size and testing the effect of stemming of the revealed algorithm in [12]. In this experiment, 2490 sentences will be utilized, and the result will be compared with the result of a small size dataset in [12]. Primarily, the pre-test was performed iteratively and errors encountered in the pre-test were corrected then the experiment trained until the outcome is found satisfactory. In the final experiment, free text (test dataset) feeds into the system then it computes and output a tagged target word with a forecasted sense. Then the manually tagged test dataset of 249 Afaan Oromoo sentences are compared with the forecasted result.

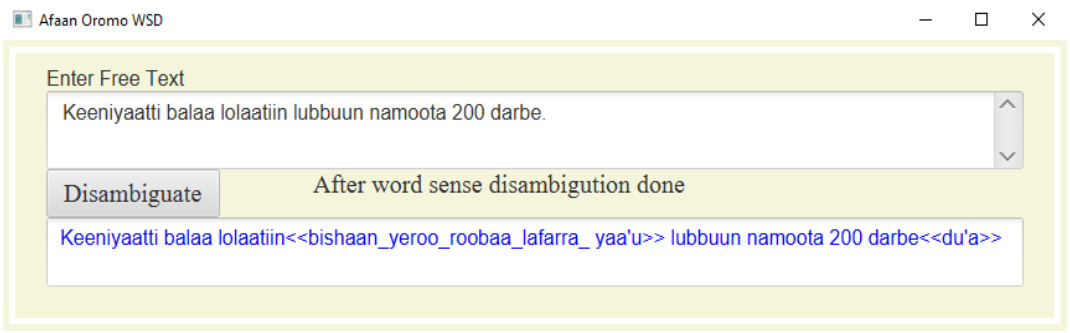

Fig. 3. Screenshot of Afaan Oromoo WSD

There is no common agreement of evaluation metrics in the history of a WSD literature. However, in several literature metrics like precision, recall, F1, and accuracy have been used as performance evaluation metrics [3, 4, 12, 33-35]. This section briefly discusses an evaluation method regarding the adopted algorithm with and without stemming to address window size disagreement and improve WSD effectiveness in Afaan Oromoo. To evaluate the level of system disambiguation the researcher evaluates via various metrics on 249 test sentences, and compares the system result with a human judge. The metrics are precision (P), recall (R), and F1. The evaluation criteria are based on the count of True Positives, False Positives, True Negatives, and False Negatives represented as TP, FP, TN, and FN respectively. Therefore, for all metrics, the respective formula presented as follows.

$$Precision(P) = \frac{\text{TP}}{\text{TP+FP}} \tag{5}$$

$$Recall(R) = \frac{\text{TP}}{\text{TP+FN}} \tag{6}$$

$$F - Measure(F1) = \frac{\text{2PR}}{\text{P+R}} \tag{7}$$

Table 2. Experimental result of Afaan Oromoo WSD

Experiment	χ	γ	**Without morphology consideration**			**With morphology consideration**		
			P	R	F1	P	R	F1
Exp_1	0.1	0.9	0.8775	0.7551	0.8117	**0.8476**	**0.9789**	**0.9085**
Exp_2	0.2	0.8	0.8780	0.7585	0.8139	0.8415	0.9787	0.9049
Exp_3	0.3	0.7	**0.8911**	**0.7710**	**0.8267**	0.8415	0.9787	0.9049
Exp_4	0.4	0.6	0.8819	0.7593	0.8160	0.8415	0.9787	0.9049
Exp_5	0.5	0.5	0.8765	0.7483	0.8073	**0.8476**	**0.9789**	**0.9085**
Exp_6	0.6	0.4	0.8669	0.7363	0.7963	**0.8476**	**0.9789**	**0.9085**
Exp_7	0.7	0.3	0.8618	0.7285	0.7896	0.8232	0.9783	0.8940
Exp_8	0.8	0.2	0.8566	0.7257	0.7857	0.8171	0.9781	0.8904
Exp_9	0.9	0.1	0.8286	0.7173	0.7689	0.8293	0.9784	0.8977
Exp_10	1	1	0.8765	0.7483	0.8073	**0.8476**	**0.9789**	**0.9085**

The weight given to sense cluster and target (test) sentences as depicted in Table 2. The pair word value of a sentence in testing dataset is represented by χ and pair word values in sense cluster (of training dataset) is represented by γ. Both χ and γ are the degrees of vague word that occur with any other word in pair. Also, the proximity of words and frequency of words in pair is under consideration.

The first method focuses on without morphology consideration experiment. In both experiments, Exp_5 and Exp_10 give the same results. For both training and testing sentence Exp_5 and Exp_10 given equal weight regardless of actual word co-occurrence degree. Exp_1 shows pair word from sense cluster is given highest weight, where pair words of a target/test sentence given the least weight that leads to better disambiguation power. From Exp_1 to Exp_8 weight of a given sentence is increased regularly. In contrast, weight for the cluster is decreased. Consequently, weighting pair words of a test sentence larger than a sense cluster affects disambiguation power as you see on Exp_6, to Exp_9. In contrast, giving more weight to sense cluster than test sentence relatively increased the performance. But, the best performance is 82.67% in F1 that is achieved on Exp_3. Findings of the experiment indicate an optimal pair word weighting 0.3 is given (test) sentence and 0.7 for sense cluster training dataset where the method is without morphology consideration.

Morphology consideration is a very significant step towards efficient NLP for highly inflectional languages like Afaan Oromoo [32]. To ensure that, the last method focuses on performing stopword removal and stemming on the dataset during generating word co-occurrence. Like the first method, sense cluster and test sentence are given different weights except for Exp_5 and Exp_10. Starting from Exp_1 unto Exp_10 weight of the test sentence increases and weight of the cluster decreases. In Exp_7, Exp_8, and Exp_9 weight of the test sentence is exceeding that of the sense cluster, which leads to records least precision. A vague word detection depends on the stem of the target word and normalization is also applied on two target words so, in all experiments recall achieved about 98%. In experiment Exp_1, Exp_5, Exp_6, and Exp_10 achieve the best result in F1 that is 90.85%. The second-best performance is 90.49% in F1 recorded by Exp_2, Exp_3, and Exp_4. Exp_8 achieves the least performance by the difference of 1.81% gap with the best performance that happens when high weight is given to test sentences. Without any weighting mechanism, the system performance ranked the first best. The finding of the experiment indicates that there is no specific optimal weighting when a morphology consideration is applied. On other hand, any weighting values given to a cluster larger than the test sentence achieve the best performance without morphology consideration.

A comparison of the two methods on the 10 conducted experiments is depicted in Table 2. Regarding morphology consideration, the system works its best 82.67% in F1 on Exp_3. Likely, with morphology consideration Exp_1, Exp_5, Exp_6, and Exp_10 achieve the best performance that is 90.85%. Computationally in terms of time, Exp_10 needs no weighting calculation. Accordingly, an algorithm works its best without any effort of weighting optimization. Morphology consideration significantly improves the performance by 8.18% than without the morphology consideration method. The result shows that the adopted algorithm has ability to disambiguate vague words with morphology consideration.

From the experiment, author realized that FN happens when a target vague word is under-stemmed or over-stemmed. Additionally, when dialect variation is happening the stemmer could not handle it. For instance "darbe" and "dabre" are both the same. In addition, the stemmer accuracy is about 96% so, it is not perfect but, normalizing under-stem to correct stem was performed for two target words. The system disambiguation power in precision, recall, and F1 is 89.11%, 77.10%, and 82.67% respectively without a morphological analyzer. Likely, with a morphological analyzer, the achieved precision, recall, and F1 are 84.76%, 97.89%, and 90.85% respectively. Comparatively, both experiments, with a morphological analyzer increased 20.79% of a recall but precision decreased by 4.35%. Generally, in F1 with morphological analyzer outperformed by 8.18%. I can conclude that using stemming is significantly improving the effectiveness of WSD for morphologically rich language similar to Afaan Oromoo.

The performed experiment suggests that the meaning (i.e. semantic meaning) of a word is intently related to the words occur in the same scenario. Particularly, morphology consideration plays a vital to increase recall. The end result acquired through the algorithm was enough to extract semantic information from the specified text. The approach relies upon automatic sense disambiguation, which is more reliable and verifies on its part.

Many efforts were made to address the problem that realized in WSD of Afaan Oromoo [3, 4, 12, 33-35]. They were developed a model and tested on different number of targeted vague words that trained and tested on different corpus.

As revealed in Table 3 except ref. [3] and [12] all are utilized different windows size. Where ref. [3] was rule based ref. [12] was utilized all words in the sentence. Definitely, all mentioned investigators utilized different data, and number of targeted vague words. According to ref. [33] window sizes one of left and right and also window size two of left and right achieve satisfactory result, particularly size ±1. Similarly, the finding of ref. [4] and [34] shows widow of sizes two of both left and right was suggested for the language. In another research window size three in both sides was recommended in ref. [32]. The last finding is context size 4 of both sides suggested in ref. [35] for Afaan Oromoo. Generally, there is no agreement on window size which is emerged from the standard datasets unavailability, especially in performance evaluation.

Ref. [12] was proposed a novel algorithm for unsupervised ML that learns from unlabeled Afaan Oromoo text without stemming to address disagreement on window size. The algorithm performance was ample that is 80.76% of F1. However, the author trained on the small-size dataset for small target words. In Table 3 contains a comparative of all concerned researches with their dataset and performance that mentioned in their paper.

Table 3. Performance comparative

References	Ambiguous Word	Training Dataset (Sentence)	Testing Dataset (Sentence)	P (%)	R (%)	F (%)	Accuracy (%)
[35]	5	1116	124	76	88	81	79
[3]	15	-	-	-	-	-	-
[32]	-	-	50	75.75	69.78	71.60	63.95
[33]	20			89.47	85	-	-
[34]	75	-	-	85	80	-	-
[4]	15	-	-	-	-	-	*65.50*
[12]	20	498	249	88.54	74.23	80.76	-
The adopted method	30	2,241	249	84.76	97.89	90.85	84.50

The main challenge to consider is that the unavailability of public and balanced dataset for WSD makes it difficult to compare performance of researches conducted on the language yet. As depicted in Table 3 different evaluation metrics have been used by different authors. So, it's difficult to compare the performance of all researches that have conducted WSD for Afaan Oromoo. However, to summarize the proposed method is outperformed in both F1 and accuracy.

5. Conclusion

Clustering is one of unsupervised approach used for WSD, focusing on grouping semantically similar information together. This paper contributes a new way to sense disambiguation that depends on computing all pair words of the text with each pair in respective data. It came with effective way to disambiguate vague word without any predefined window size. Each word in a sentence with vague word takes part in disambiguation process. Words in pair proximity degree and pair frequency taken into consideration.

Interestingly, removing stopword and performing stemming play a great role in boosting the result. I can conclude that the adopted algorithm has ability to disambiguate vague words' sense. Besides, the adopted algorithm can detect semantic relation through normalized statistical information based on the nearness of a word in pair and pair word frequency. The achieved best performance in precision, recall, and F1 are 84.76%, 97.89%, and 90.85% respectively with morphology consideration.

As a weakness, the algorithm relies on a dataset grouped per sense, which is unavailable for under-resourced languages. Furthermore, it doesn't disambiguate all vague words not targeted in a corpus.

As a strength, the training dataset was list of sentences grouped per word sense without any sense tag. Then, learning from the data and sense tagging for a target vague word of any text is easy with outperformed yield. Indeed, this is the finest and effortless sense clustering, to solve the issue of windows size variance have seen in Afaan Oromoo word sense disambiguation. Additionally, the algorithm can prolonged for any languages, especially for morphologically rich languages.

6. Future Works

For a full-fledged Afaan Oromoo WSD adding a large dataset and extra vague words along concerned senses are expected. The adopted algorithm can be customized for semantic text similarity, if trained on large corpus.

References

[1] N. Bouhriz, F. Benabbou, and E. Habib, "Word Sense Disambiguation Approach for Arabic Text", *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 4, 2016

[2] M. Bevilacqua, and R. Navigli, "Breaking through the 80% Glass Ceiling Raising the State of the Art in Word Sense Disambiguation by Incorporating Knowledge Graph Information", *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, July 5-10, 2020, pp. 2854–2864

[3] Workineh T., Debela T., and Teferi K., "Designing Rule Based Disambiguator for Afaan Oromo Words", *Am J Compt Sci Inform Technol*, Vol. 5, No. 2, 2017 pp. 1-4

[4] W. Tesema, D. Tesfaye, and T. Kibebew, "Towards the sense disambiguation of Afan Oromoo words using hybrid approach (unsupervised machine learning and rule based)", *Ethiopian Journal of Education and Sciences* Vol. 12, No.1, pp. 61-77, 2016

[5] Beekan E., "Oromoo Language (Afaan Oromoo)",https://scholar.harvard.edu/erena/oromo-language-afaan-oromoo Accessed 20 Sept. 2021.

[6] T. Keneni, "Prospects and Challenges of Afan Oromo: A Commentary", *Theory and Practice in Language Studies*, vol.11, No.6, June 2021, pp.606, Accessed 15 Sept. 2021.

[7] Guya, T. "CaasLuga Afaan Oromoo: Jildii-1", Gumii Qormaata Afaan Oromootiin Komishinii Aadaa fi Turizimii Oromiyaa, Finfinnee, 2003.

[8] W. Tesema, and Duresa T., "Investigating Afan Oromoo Language Structure and Developing Effective File Editing Tool as Plug-in into Ms Word to Support Text Entry and Input Methods", *American Journal of Computer Science and Engineering Survey*, 2021.

[9] Abdo Ababor Abafogi, "Boosting Afaan Oromo Named Entity Recognition with Multiple Methods", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.13, No.5, pp. 50-58, 2021. DOI: 10.5815/ijieeb.2021.05.05.

[10] K. Tune, V. Varma, P. Pingali, "Evaluation of Oromo-English Cross-language Information Retrieval", *ijcai workshop on clia*, hyderabad, india, 2007

[11] B. Sankaran, K. Vijay-Shanker, "Influence of morphology in word sense disambiguation for Tamil", *Anna University and University of Delaware Proc. of International Conference on Natural Language Processing*, 2003.

[12] Abdo Ababor Abafogi, "Normalized Statistical Algorithm for Afaan Oromo Word Sense Disambiguation", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.13, No.6, pp.40-50, 2021. DOI:10.5815/ijisa.2021.06.04.

[13] Y. Wang, M. Wang, H. Fujitas, "Word Sense Disambiguation: A comprehensive knowledge exploitation framework", *Knowledge-Based Systems*, Vol. 190, 29 February 2020.

[14] D. Yarowsky, "Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and French", *In Proc. of the 32nd annual meeting on Association for Computational Linguistics*, 1994, pp. 88–95.

[15] B. Scarlini, T. Pasini, and R. Navig, "SENSEMBERT:Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation", *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20) Association for the Advancement of Artificial Intelligence*, 2020.

[16] C. Hadiwinoto, H. T. Ng, and W. C. Gan, "Improved Word Sense Disambiguation using pre-trained contextualized word representations", *In Proc. Of EMNLP*, 2019, pp. 5297–5306

[17] M. Bevilacqua, and R. Navigli. "Quasi bidirectional encoder representations from Transformers for Word Sense Disambiguation", *In Proc. of RANLP*, 2019, pp. 122–131.

[18] F. Scozzafava, M. Maru, F. Brignone, G. Torrisi, and R. Navigli,"Personalized PageRank with syntagmatic information for multilingual Word Sense Disambiguation", *In Proc. of ACL* (demos), 2020.

[19] K. Sawan, J. Sharmistha, S. Karan, and T. Partha, "Zero-shot Word Sense Disambiguation using sense definition embeddings", *In Proc. of ACL*, 2019.

[20] T. Blevins, and L. Zettlemoyer, "Moving down the long tail of Word Sense Disambiguation with gloss informed bi-encoders", 2020.

[21] E. Barba, T. Pasini, and R. Navigli, "ESC: Redesigning WSD with extractive sense comprehension", *In Proc. of NAACL*, 2021

[22] S. Conia, and R. Navigli, "Framing Word Sense Disambiguation as a multi-label problem for model-agnostic knowledge integration", *In Proc. of EACL*, 2021.
[23] M. Bevilacqua, T. Pasini, A. Raganato, and R. Navigli, "Recent Trends in Word Sense Disambiguation:A Survey", *Proc. of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21).*
[24] Vial L, Lecouteux B, and Schwab D, "Sense Vocabulary Compression through the Semantic Knowledge of WordNet for Neural Word Sense Disambiguation", *In Proc. of Global Wordnet Conference*, 2019.
[25] A. Raganato, C. Bovi, and R. Navigli, "Neural sequence learning models for word sense disambiguation", *In Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics, 2017, pp. 1156–1167
[26] B. H. Manjunatha Kumar, "Survey on Word Sense Disambiguatio", Sri Siddhartha Institute of Technology, 2018. *Conference on Emerging Trends in Engineering, Science and Technology*, 2015.
[27] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Embeddings for word sense disambiguation:An evaluation study", *In Proc. Of ACL*, Vol. 1, 2016, pp. 897–907
[28] O. Melamud, J. Goldberger, and I. Dagan, "context2vec:Learning generic context embedding with bidirectional LSTM", *In Proc. of CoNLL*, 2016, pp. 51–61
[29] S. Sankar, Reghu Raj, V. U. Jayan, "Unsupervised Approach to Word Sense Disambiguation in Malayalam*", International* .
[30] Krishnanjan et al., "Survey and Gap Analysis of Word Sense Disambiguation Approaches on Unstructured Texts", *Proc. of the International Conference on Electronics and Sustainable Communication Systems* 2020.
[31] M. Gunavathi, and S. Rajini, "The Various Approaches for Word Sense Disambiguation: A Survey", Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, India, *IJIRT*, Vol. 3, No. 10, March 2017.
[32] S. Olika, "word sense disambiguation for Afaan Oromoo using knowledge base", St. University College, 2018.
[33] Yehuwalashet B., "Hybrid Word Sense Disambiguation Approach for Afan Oromoo Words", Thesis, Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia, 2016.
[34] W. Tesema, and D. Tesfaye, "Word Sense Disambiguation and Semantics for Afan Oromoo Words using Vector Space Model", *International Journal of Research Studies in Science, Engineering and Technology, Vol. 4, No. 6, 2017, pp. 10-15.*
[35] K. Tesfa, "Word sense disambiguation for Afaan Oromoo Language",Thesis, Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia, 2013
[36] Getachew T., "Seerluga Afaan Oromoo", Finfinnee Oromiyaa press, 2014.
[37] Daraje K, "Afaan Oromoo Automatic word stemmer", Wollega University, 2018.

Author's Profile

Abdo Ababor Abafogi received his BSc and MSc in Information Technology, from Jimma University, Ethiopia in 2012 and 2017 respectively. His research interest areas are Artificial Intelligence, Data Science, Deep Learning, Machine Learning, Natural Language Processing, and SEO.

How to cite this paper: Abdo Ababor Abafogi, "Enhanced Word Sense Disambiguation Algorithm for Afaan Oromoo", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.15, No.1, pp. 41-50, 2023. DOI:10.5815/ijieeb.2023.01.04