I.J. Intelligent Systems and Applications, **2021, 6, 40-50**
Published Online December 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2021.06.04

Normalized Statistical Algorithm for Afaan Oromo Word Sense Disambiguation

Abdo Ababor Abafogi
Department of Information Technology, College of Computing and Informatics, Wolkite University, Ethiopia
E-mail: abdo.ababor@wku.edu.et

Received: 27 August 2021; Revised: 24 September 2021; Accepted: 03 October 2021; Published: 08 December 2021

Abstract: Language is the main means of communication used by human. In various situations, the same word can mean differently based on the usage of the word in a particular sentence which is challenging for a computer to understand as level of human. Word Sense Disambiguation (WSD), which aims to identify correct sense of a given ambiguity word, is a long-standing problem in natural language processing (NLP). As the major aim of WSD is to accurately understand the sense of a word in particular context, can be used for the correct labeling of words in natural language applications. In this paper, I propose a normalized statistical algorithm that performs the task of WSD for Afaan Oromo language despite morphological analysis The propose algorithm has the power to discriminate ambiguous word's sense without windows size consideration, without predefined rule and without utilize annotated dataset for training which minimize a challenge of under resource languages. The proposed system tested on 249 sentences with precision, recall, and F-measure. The overall effectiveness of the system is 80.76% in F-measure, which implies that the proposed system is promising on Afaan Oromo that is one of under resource languages spoken in East Africa. The algorithm can be extended for semantic text similarity without modification or with a bit modification. Furthermore, the forwarded direction can improve the performance of the proposed algorithm.

Index Terms: Afaan Oromo, Word Sense Disambiguation, Normalized Statistical Algorithm, Unsupervised Approach, Sense Cluster Algorithm.

1. Introduction

In human language, often a word is used in many ways. In various situations, the same word can mean differently based on the usage of the word in a particular sentence. The usage of words defines a lot about their meaning. But the problem lies that, in NLP, while dealing with text data, we need some way to interpret the same words with different meanings. Furthermore, words cannot be divided into discrete meanings. Words often have related meanings or unrelated multiple meanings and this causes a lot of problems in NLP applications.

Word Sense Disambiguation is a field of NLP that aims at determining the correct sense of an ambiguous word in a particular context [1, 2, 3]. Interpreting a target word in a given situation is very important for NLP because of the word ambiguity and richness of human languages [4]. As the major aim of WSD is to accurately understand a sense of a word in particular usage, it can be used for the correct labeling of words.

WSD acts as an intermediate phase in numerous NLP applications like Information Retrieval (IR), Machine Translation, Speech Processing, Parts-of-Speech tagging, and Hypertext navigation [3]. WSD can be used in Information Extraction and Text Mining tasks. Additionally, WSD is used in machine translation to give contextual meaning source language [5], in IR system it is used to execute queries in the query context, also to determine contextual documents relevancy. Disambiguation is the most challenging task at all levels of a natural languages, especially for under resource languages the challenge is desperate.

Afaan Oromo is one of under resource languages used by the largest ethnic group in Ethiopia, which amounts to 50% of the Ethiopian population [6, 7, 8]. With regards to the Afaan Oromo writing system, Qubee (a Latin-based script) has been adopted and became the official script since 1991 [8, 9, 10], which is about three decades ages. Afaan Oromo consists of 33 basic letters, grouping into three groups named consonants, vowels, and paired consonant letters (such as 'dh', 'ch', etc.) and the letters in each group are 24, 5, and 7 respectively [2].

In Afaan Oromo, like in English languages, all letters are characterized as capital and small. Blank space shows a boundary of a word. Additionally, parenthesis, quotes, and brackets are used to show a word's boundary. Sentence boundary punctuations are a period (.), a question mark (?), or an exclamation mark (!) [11]. Punctuation marks are used in the same way and for the same purpose as used in English languages; except an apostrophe mark ('). An apostrophe in English shows possession but in Afaan Oromo known as "hudhaa" which is part of a word. It plays an important role in the language reading and writing system [2].

In human language a word can be interpreted in more than one way depending on different contexts. However, for a computer it is challenging to recognize as level of human. Many efforts have been made to solve the WSD problem for Afaan Oromo. For under-resource language like Afaan Oromo an unsupervised approach is recommended. However, there is no common agreement on context window sizes that used with unsupervised method. Consequently, it needs additional research aims at possible concern of handling all context modifiers without window sizes limitation. From this viewpoint, the author motivated to minimize the gap of window sizes consideration in unsupervised learning for Afaan Oromo text based applications.

Unavailability of a resources and the complexity of the language also the richness of the morphology are the main challenges for Afaan Oromo language processing [12, 13]. In Afaan Oromo, the sense of words is based on the words preceded by the word (modifies) [6, 7]. The modifiers and contextual information were the basis of the linguistic properties of Afaan Oromo word sense. Ambiguity can occur at several levels of the language such as lexical, semantic, syntactic, and pragmatic, etc. [14].

On the target language an unsupervised approach is obtained encouraging performance and also rule-base with unsupervised approach is outperformed. On the other hand, the author faced a noteworthy challenge because Afaan Oromo has a resource lack. So, to cope with the challenge, an alternate solution propose that relies on unsupervised sense disambiguation.

The ultimate aim of this paper is developing a normalized statistical algorithm that relies on an unsupervised approach without any specific windows size to perform the task of WSD for Afaan Oromo language. Additionally, a language exhibiting similar patterns with Afaan Oromo can adapt the algorithm. Specifically, it increases the methodology of the WSD research. The proposed algorithm can be extended for semantic text similarity without modification or with a bit modification. Furthermore, it has been pointed out how NLP plays a significant role in enhancing the computer's capability to process word senses.

In this work, to get the semantic clues of a particular sense of the ambiguous word, the proposed algorithm will be applied upon a dataset described in section dataset development. The algorithm starts by detecting ambiguous words, analyzes contextual information and linguistic properties of word sense. Next, coupling every word with an ambiguous word found in a sentence with their weights (that is proximity degree to ambiguous word), then pair a word with proximity degree grouped into a sense-specific cluster. The coupled words provide information about ambiguous words and their identification degree per sense that is the last step in a training phase.

The first task of the algorithm is text preprocessing (like distinguishing statements boundary, tokenization, and eliminating punctuations). Next, search for an ambiguous word in the given text (sentence), generating and calculating word co-occurrence degree per sentence. Then, recursively search each pair word of the sentence whether available in a sense cluster, if found it calculates weighted context overlap from respective sense cluster. The calculation is continue up to the end of the sentence for the respective ambiguous word and with a concerned cluster. Once context overlap calculation is completed, the summation of all overlapped pair yield is computed and the cluster scores greatest summation result is nominated and used to tag the sense of the ambiguous word. The input to the WSD system is an Afaan Oromo text and the output from the system is a text in which the ambiguous word is tagged with its predicted sense.

2. Related Work

Human beings are pretty apt in determining the perfect sense of the word, but for machines that is a very hard challenge. The research on automatic disambiguation of ambiguous word sense has been a critical concern since its emergence due to its extensive applications. WSD is a necessary intermediate task at many levels to accomplish most NLP tasks. It is essential in many language understanding applications, particularly in human-machine communication. As a result, several systems were proposed and validated on standard datasets that are specialized in WSD evaluation [15-7]. In this section several WSD researches have been reviewed about an approaches, state-of-the-art work from resource rich languages, and all available works of the Afaan Oromo WSD.

In WSD history, a lot of algorithms have been proposed under the category of knowledge-based (KB), supervised, and unsupervised approaches. KB approaches incorporate systems that rely on linguistic information sources like dictionaries, Wordnet, thesauri, and hand-crafted rule bases. Supervised approaches incorporate systems that rely on annotated corpora to learn from. Unsupervised methods learn from unannotated corpora.

KB methods to WSD such as Walker’s algorithm, Lesks algorithm, and random walk algorithm do a machine-readable dictionary lookup. One of the earliest KB algorithms is Lesk algorithm, which computes the word overlap between the target word’s context and possible senses’ definition. Word collocations are important in determining ambiguous meaning. A statistical decision for lexical ambiguity based on decision lists is discussed in [16]. Recently, a multi-languages KB approach was proposed named SENSEMBERT as discussed in [17]. SENSEMBERT proved to be effective both in the English and multilingual WSD tasks. The SENSEMBERT used the lexical-semantic information in BabelNet, and Wikipedia, to overcome the burden of producing manually-tagged corpora. In the neural network era, SENSEMBERT outperformed the state of the art in all the tested languages than most of the supervised [17].

Supervised methods exist ranging from purely supervised [18, 19] to knowledge-based [20], to hybrid supervised

and KB approaches [21, 22, 23, 24, 25, 26]. In general, the supervised WSD approach concerns purely data-driven models [18], supervised models exploiting glosses (human-readable way of clarifying sense distinctions) [21], supervised models exploiting relations in a knowledge graph such as WordNet hyponymy and hypernymy relations [27], and supervised approaches using other sources of knowledge like Wikipedia and Web search [26]. In recent years, more supervised systems are proposed to cope with WSD tasks. "It makes sense" is a system that employs SVM to disambiguate words in context [15]. The best supervises approaches rely on neural networks [28].

Supervised systems have some disadvantages such as scarcity, availability of standard corpora, etc. The supervised WSD approaches have yielded better results as compared to the unsupervised WSD approaches [29]. KB and unsupervised methods are used to improve the system performance [30]. A supervised system was the most successful approach to WSD across all English datasets [31, 32, 27]. In recent years, the approaches surpassed by SENSEMBERT KB approach [26].

Unsupervised WSD and KB WSD are evolving as a great option to resolve the challenges of supervised systems. A supervised approach needs a huge annotated corpora. As a result, for less resource language annotated corpora may not available. In this case, the solution is in the hand of unsupervised methods. It combines the advantages of supervised and KB approaches. As with the supervised approach, it gathers information from the corpus and does not need a tagged corpus [29].

Unsupervised Methods pose the greatest challenge to researchers and NLP professionals. The unsupervised WSD approach does not require annotated corpus. These techniques identify the sense of the ambiguous word from the neighboring words, called context. A key assumption of these models is that similar meanings and senses occur in a similar context. They are not dependent on manual efforts, hence can overcome the knowledge acquisition deadlock. Prepares the clusters of the word occurrences in the input text and then induces senses of a new occurring word into the proper cluster [33]. An unlabeled dataset is required to be trained before applying them to ambiguous words [34].

Unsupervised learning identifies patterns in a large sample of data, without the benefit of any manually labeled examples or external knowledge sources. These patterns are used to divide the data into clusters, where each member of a cluster has more in common with the other members of its cluster than any other. The different methods of this approach are context clustering, word clustering, and co-occurrence graphs beside k-means clustering in case of different and huge dataset availability [33]. The two well-known unsupervised methodologies are clustering and association rules which are utilized for word ambiguity [35]. The major shortcomings of the unsupervised approach are they do not rely on any shared resources like dictionaries for word senses.

On other hand a few researches made on WSD of Afaan Oromo language. In [36] a supervised approach is applied a Naïve Baye's theory to disambiguate 5 ambiguous words in the Afaan Oromo language. The work has trained and tested on 1116 sentences and 124 sentences respectively. The author concluded that ±4 (right and left side) of either ambiguous word window size is sufficient for sense disambiguation in Afaan Oromo.

In other work, a rule-based research has been conducted to solve the problem of WSD for Afaan Oromo. The work was focused on 15 ambiguous words, due to under the resource of the language [3]. Similarly, Shibiru [37] conducted knowledge base WSD research using a window size of ±1to ±5 to the left and the right side of the ambiguous word. Based on Afaan Oromo Wordnet with morphological analyzer and without morphological analyzer. Besides that, he recommended an optimal windows size for Afaan Oromo WSD. The experimental evaluation performed on 50 sentences shows that a ±3 windows size on either side of the ambiguous word with morphological analysis is ample for Afaan Oromo WSD [37].

In contrast, Yehuwalashet [38] modeled a hybrid approach that relies on a rule-based and an unsupervised machine learn from a corpus to solve the challenge of WSD. The context of the ambiguous word is determined via the constructed vector representations matrix from word co-occurrences and an extracted modifiers of the ambiguous word using rules. As a result, the cosine similarity was computed based on the angle between vectors of the contexts. She evaluated 20 ambiguous words with the same window size, and clustering algorithms such as EM, K-Means, Complete link, Single link, and Average Link. Finally, she concluded that window sizes ±1 and ±2 perform a better result with EM and K-means algorithms. The best accuracy is achieved by EM.

Furthermore, [7] was utilized the hybrid machine learning approach, in which an unsupervised machine learns with the help of a handcrafted rule approach to cluster the contexts and also, to extract the modifiers of the ambiguous word. The words that occur in similar contexts tend to have similar senses. The senses and contexts can be captured in terms of the frequency, co-occurrence neighborhood. The author used the same algorithms and the finding found in both papers show that using the window size of ±2 words on either side of the target word computed using cosine similarity offered better accuracy via the EM algorithm [7].

From the mentioned related works of low-resource languages gap is found as shown on context window size. Indeed, all of the researchers utilized different datasets, different numbers of the ambiguous word. In Afaan Oromo, author [38] concluded that window sizes ±1 and ±2 perform better in general and window size ±1 is in particular. Author [7] recommended window sizes ±2 for Afaan Oromo. The third finding shows that ±3 context window size is enough for WSD in Afaan Oromo. The author of [36], recommend that ±4 window sizes of target word offered better accuracy.

In general, there is no common agreement on context window sizes to determine a sense of a target word.

Consequently, it needs additional experimentation aims possible concern of handling all context modifiers without window sizes limitation. From this viewpoint, I attempt to develop an algorithm that minimizes the gap of window sizes consideration in unsupervised learning.

They are two classifications of WSD tasks: the first one is lexical sample WSD focuses on disambiguating only some particular target words and another one is all-word WSD that conducts disambiguating every word in a document [7]. This paper focuses on lexical sample WSD.

3. Methodology

This section presents the main proposals regarding dataset development, preprocessing, architecture of the system, employment of ambiguous word identification, word co-occurrence generation, sense clustering algorithm, and sense disambiguation algorithm.

3.1. Dataset development

A standard annotated corpus is not available for Afaan Oromo. To prepare a balanced corpus the researcher collected a generic news article from 3 news websites (BBC Afaan Oromo, VOA Afaan Oromo, and Fana Afaan Oromo) regarding a target ambiguous word. After collecting a considerable amount of data, further edited manually by removing those sentences which do not have ambiguous words. Furthermore, there are no repetitive sentences to improve corpus quality. For testing purposes, 20 sentences having no ambiguous words were added to the test dataset. The prepared dataset consists of 727 sentences to evaluate 20 Afaan Oromo ambiguous words. For training 498 sentences are used. The remaining 249 sentences, where 20 of the sentences are having no ambiguous words were used for testing purpose.

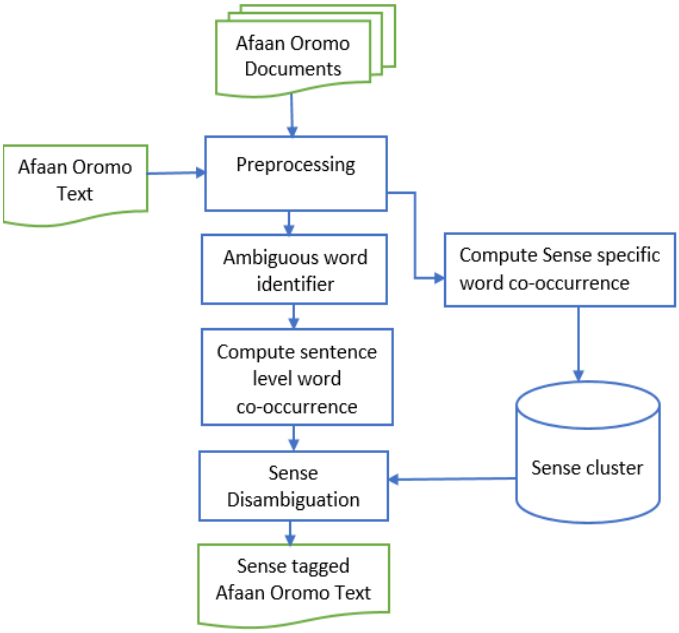

Fig.1. System Architecture

The proposed algorithms for this WSD purpose formulated in equations 1 to 4 are part of the above system architecture. As Fig.1 shows Afaan Oromo Documents represent a sense specific text used for the training purpose. Afaan Oromo text is any given sentences for the sake of testing the system. The next task is preprocessing all text/documents. Then a target words is identified before co-occurrence of target words with any other words in the current sentence generated. If it's on training dataset it stored on sense cluster for later usage, however if it is testing data it utilize a co-occurrence degree and disambiguate according to the algorithm. The detail of the system architecture shown on Fig.1 will be presented in the following sub-sections step by step and in detail

3.2. Preprocessing

For the training and testing dataset, elementary preprocessing like parsing text into sentences and tokenization are performed. Tokenization is a process of word identification based on the boundaries of a token. Tokenization is a language-dependent in Afaan Oromo word demarcation performed splitting text on space. After splitting via blank space, punctuation marks that appended at end of token removed similar to that of English which include the semicolon (;), comma (,), full stop (.), question mark (?) and exclamation mark (!). However, a period for abbreviation and decimal

number as well as hyphen (-), and also apostrophe (') known as "*hudhaa*" which is considered as part of a word not removed if and only if it is written either between alphabet or between number. In Afan Oromo text *hudhaa* is frequently occurred to specify missed consonants in words. If this character is not considered a part of the word, a single word will be split into (three) tokens with no context. In Afaan Oromo, the word o'clock or hour is *sa'aa*.

3.3. Ambiguous Word Identification

In order to perform the word disambiguation task, the first step is detecting the ambiguous word from the given text that is employed by checking each word of the sentence in ambiguous word vocabulary. The vocabulary consists the ambiguous word mentioned in section of dataset development. If ambiguous word is found in the sentence (training and/or test dataset), for the detected ambiguous word, all words in the given sentence partake in the sense disambiguation phase.

3.4. Generating word co-occurrence

In [7] the authors finding shows that smaller window sizes usually lead to better accuracy than bigger window sizes. Thus, from this viewpoint I attempt to develop an algorithm that minimize the gap of window size consideration in unsupervised learning.

Once the ambiguous word is identified, generating word co-occurrence consider the distance of ambiguous word *wa* from any other words *wj* where *wj* become *w1*, *w2*, *w3*,... *wn*. The distance calculates the difference in an index (position) of pair words in the given text in the form of $wa_{\text{index}} - wi_{\text{index}}$. The weighted distance calculated as equation 1. Let, take one Afaan Oromo text "*Gaazexeessichi sodaa nageenyaatiifan biyyaa* ***bahe*** *jedhe"*. In this text *bahe is an ambiguous word. We stand on this position to calculate co-occurrence proximity that simulate as (bahe, Gaazexeessichi)*[4]*, (bahe, sodaa)*[3]*, (bahe, nageenyaatiifan)*[2]*, (bahe, biyyaa)*[1]*, (bahe, jedhe)*[1]. The superscript number is show the actual difference in position of words from ambiguous word wa. The next equation minimize thus gaps.

Fig.2. Word co-occurrence proximity sample

$$dist = \log \frac{0.278}{(wa_{index} - wj_{index})} \tag{1}$$

The result from the above equation (1) for pair words are negative where the actual distance words from ambiguous word *bahe* is 1, 2, 3, and 4 the weighted distance become -1.280, -1.973, -2.379, and -2.667 respectively. In contrast, ambiguous word with itself *(bahe* with *bahe)* is excluded in this equation. All of the results are negative numbers that need changing to positive and weighted co-occurrence of pair word (*wa* and *wi*) is computed by the next equation 2.

$$pair(wa, wi) = abs(2.78 * dist * 2.7^{dist}) \tag{2}$$

Where *abs* is absolute value change any negative number into positive. The equation calculates the closeness of each word in pairs, at every co-occurrence in all sentences. The probability of word co-occurrence is calculated by equation 2, which is used to determine the most frequently and nearly occurred words in a sentence that determine the sense of words according to the context. Even in the same senses, the distance of words *wa* and *wi* are not similarly distributed across the sentences. In equation 2, the maximum and minimum result is 1 and 0 respectively for each pair. Fig.2 shows that the distance between an ambiguous word *bahe* and any other words represented by 1, 2, 3, and 4 normalized into 0.99, 0.77, 0.62, and 0.52 respectively by the equation.

3.5. Clustering similar sense words

In the training corpus, similar sense sentences are gathered together to generate sense-specific word co-occurrence with their pair frequency. To disambiguate word sense it needs word co-occurrence statistics information for each sense of the word that computed according to equation 3. The equation focuses on pair of words in the sentence that makes sense specific.

$$SenseCluster_{(wa,wi)} = \frac{\sum \text{abs}(2.78 * \text{dist} * 2.7^{\text{dist}})}{fr(wa, wi)} \geq 0.278 \tag{3}$$

Where, $\sum$ is the summation of pair word (wa and wi) co-occurrence. $fr(wa, wi)$ is the frequency of pair word occurrence in the sense-specific dataset, used to takes its average. The minimum threshold (0.278) is taken in order to minimize noise that happens in the sense prediction phase. All words greater than the given threshold are grouped per sense to discriminate a word's sense and stored for later sense disambiguation purposes. This is done on a training dataset for each ambiguous word. The cluster is prepared and store for later sense lookup.

Algorithm for sense clustering

Starts by reading a folder that contains a subfolder.
- *Read a subfolder that contains the list of files*
  - *Read a file*
    - *Read a list of sentence and tokenize it*
    - *Computes co-occurrence of wa and wi via equation 2*
  - *Calculate via equation 3*
  - *Write to file and move to next sense file*
  - *Move to the next subfolder*

End of sense clustering algorithm

The requirement for sense clustering algorithm is described in this section. Firstly, the researcher creates a folder for training data. Then 20 subfolders were created and named by each ambiguous word in every subfolder, a file has been created for each sense of the by the name of respective ambiguous words. In each file there are many sentences of a single sense, every sentence starts on a new line.

3.6. Sense Disambiguation

The user inputs Afaan Oromo text into the system. Text preprocessing starts by segmenting the given text into a sentence then tokenize it. Starts searching for ambiguous word availability in the sentence from ambiguous word list discussed in section dataset development. If an ambiguous word is not found in the sentence, move to the next sentence and lookup in the same way until the end of the text. If the sentence has an ambiguous word, weighted word co-occurrence with ambiguous word computed for every word in the given sentence according to equation 3.

Unlike, equation 3 the computation is considered co-occurrence frequency per the sentence only. Once pair word co-occurrence degree is computed, two different weightings considered 0.3 for the sentence and 0.7 for a sense cluster dataset. Each pair word occurrence degree in the given sentences is multiplied by 0.3 to reduce the noise of ambiguity. Similarly, for the available pair word per sense specific dataset, their co-occurrence value is multiplied by 0.7 to maximize words sense dependency which is computed as equation 4.

$$Sense_{wa}Sent_j = \sum \left(0.3Sent_{j(wa,wi)} + 0.7SenseCluster_{(wa,wi)}\right) \quad (4)$$

Where $sense_{wa}$ is a sense of ambiguous words (w_a) in the sentence ($Sent_j$) where, wi is a list of words ($w1, w2, w3 \ldots wn$), in the given sentence. The first task is computing co-occurrence of ambiguous word wa with all other words wi in the current sentence $Sent_j$ that multiplied by 0.3 as denoted $0.3Sent_{j_{(wa,wi)}}$ each pair at a time. The next task is $0.7Sense_{Cluster(wa,wi)}$ which is taking the co-occurrence degree of ambiguous word wa with wi from sense clustered dataset and multiplied by 0.7. This weighting was taken after many experimentations conducted and the detailed description present under section result and discussion. Finally, the $\sum$ (summation) computes all pair words in the given sentence with each respective clustered sense. The maximum summation yield determines the sense of an ambiguous word in the given sentence.

Algorithm for Sense Disambiguation

Read a list of text split into a list of sentences.
    If the sentence contains ambiguous word tokenize it
        Take ambiguous word position
            Generate pair word (of the sentence)
                Take each pair one by one
                    f1⇦Computes pair word co-occurrence & multiply by 0.3
                    f2⇦Take pair co-occurrence degree from dataset(*di*) & multiply by 0.7
                    sum=f1+f2
                    di+=*sum*
        Take the greatest summation of *d*i
        Determine the target word sense
    Else move to the next sentence
End of sense identification algorithms

4. Experiment and Discussion

This section presents the result and discussion of the implementation. An experiment has been conducted on 20 ambiguous words listed under dataset development that are to be distinguished. During this experiment, the researcher tries to evaluate the performance of the algorithm. The dataset of 727 Afaan Oromo sentences are divided into a training dataset (498) and a testing dataset (229). In addition, 20 sentences having no ambiguous words are added and a total of 249 sentences are used for the system performance test

Finally, the algorithm is implemented as Fig.2 in the way that free text is given from the user. The system preprocesses a text, then search for an ambiguous word, if available it generates word co-occurrence. Next, the algorithm computes according to equation 4, after that gives sense tagged output.

Fig.2 shows a sample sentence as *Gaazexeessichi sodaa nageenyaatiifan biyyaa bahe jedhe.* After WSD done the final output tagged with a predicted sense.

Gaazexeessichi sodaa nageenyaatiifan biyyaa bahe <<deeme>> jedhe.

For instance term *bahe* is a target ambiguous word and that is contextually to mean left. Generally the sentence translated into the journalist said “I left a country for a safety threat”.

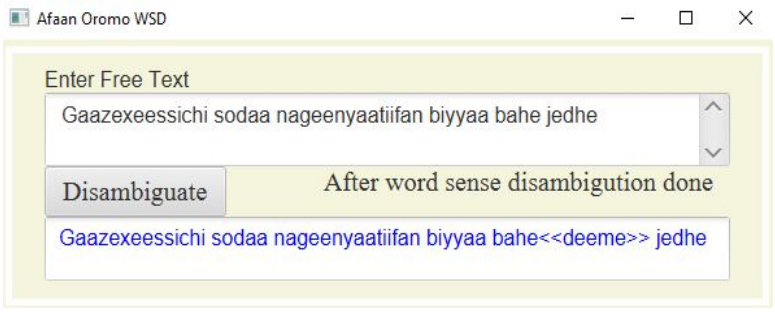

Fig.2. Sample of Afaan Oromo WSD

Initially, the pre-test has been conducted iteratively to increase the performance. The errors encountered during the pre-test have been corrected and the experiment has been done iteratively until the result is found to be satisfactory.

Next, to evaluate the result of the proposed system, I manually separate the testing set then put it on a hashmap[1] by means of sentences as key and the respective sense as value. During the actual experiment, the system read the test dataset that is untagged free text then compute and give the predicted sense. It automatically compare the result with the manually tagged test set from the hashmap. Finally, the actual test was performed on 249 Afaan Oromo sentences. The result on the test data set was obtained by comparing the result returned by the system with the corresponding test set which was manually tagged.

4.1. Experiment

The experiment focuses on how WSD utilize all words in a sentence to disambiguate a target word. The experiment is conducted to identify the best normalized value for both training and testing dataset concerning the used

[1] one of java collection type

approach. Many experiments have been conducted at different levels on various values of word co-occurrence normalization that tested by incrementing and decrementing 0.1 value consistently on their difference. The normalization was tried by giving various weight values for word co-occurrence in the sense-specific cluster and a given sentence (test data). The summary of 11 experiment results on the test data has shown in Table 1.

As shown in Table 1 there are 2-word co-occurrence values represented by x, and y for sense cluster, and a given text (sentence) of pair word respectively. The weight has been given to both sense cluster and test sentences as shown in Table 1. As well as the last experiment conducted without any weight that gives satisfactory F-measure.

Exp1 shows word co-occurrence from sense cluster is given least weight where co-occurrence in test sentence given is the highest weight which leads to lowest disambiguation power. From Exp1, to Exp8 the weight of the sense cluster was increased almost consistently in contrast, the weight of the test sentence decreased accordingly which leads the F-measure to slightly enhanced from 73.33% (on Exp1) to 80.76% (on Exp8). Exp8 conducted the same as equation 4 and improves 7.43% as a result of applicable weighting used. Exp3 and Exp4 gave 76.19% and 77.14% respectively there is 0.95% enhanced due to test sentence weight decreased.

Table 1. Experimental results of word co-occurrence

	Sense Cluster	**Test sentence**	**F-measure**
Exp1	0.1x	0.9y	73.33%
Exp2	0.2x	0.8y	74.29%
Exp3	0.3x	0.7y	76.19%
Exp4	0.3x	0.6y	77.14%
Exp5	0.4x	0.6y	77.59%
Exp6	0.5x	0.5y	78.09%
Exp7	0.6x	0.4y	79.05%
Exp8	**0.7x**	**0.3y**	**80.76%**
Exp9	0.8x	0.2y	79.05%
Exp10	0.9x	0.1y	79.05%
Exp11	X	y	78.09%

From the result of Exp7, Exp8, Exp9, and Exp10 we recognize that the weight of the sense cluster is greater than that of the given sentence, sense prediction power become high than that of sense cluster weight is less than that of the given sentence. However, in Exp9, and Exp10 when the weight of the sense cluster is greater than 0.7 and the weight of the given sentence less than 0.3 the system decrease performance by 1.71%. When both sense cluster and test sentences are computed without weighting the system performance decreased by 2.62% than Exp8.

Moreover, when the weight of the sense cluster multiplied by 0.7 and the given sentence multiplied by 0.3 the system outperformed as achieved from Exp8. The finding found from these experiments, gives high weight to sense cluster word co-occurrence degree has significantly improved sense disambiguation power. In contrast, giving less weight to test sentences has considerably improved the performance. Generally, the conducted experiments show that the algorithm has the power of discriminating ambiguous word's sense.

4.2. Evaluation Metrics

Many efforts have been made to solve the WSD problem of Ethiopian (low-resource) languages, particularly for Afaan Oromo. Thus, research has been increasing due to it's a wide-range application coverage. As a result, a number of systems [6, 7, 26, 37, 38] were developed and evaluated on different target ambiguous words and on different datasets. According to Tesfa K. [36], supervised approach was applied aiming 5 ambiguous words which that evaluated on 124 sentences. In addition, in [6] the rule-based approach applied aiming on 15 ambiguous words. On other hand, hybrid approach [31] that blend unsupervised machine learning with handcrafted rule proposed targeting 15 ambiguous words the same as with [6]. In both [6] and [7] the size of utilized sentences for evaluation purpose is not mentioned. Similarly, Yehuwalashet [38] conducted a hybrid approach blend an unsupervised machine learning with a handcrafted rule a target on 20 words.

Furthermore, Shibiru [37] conducted a knowledge-based approach relies on Afaan Oromo wordnet that contains 267 synsets from 100 ambiguous words developed by himself. He evaluated the system on 50 sentences.

Indeed, all of the mentioned researchers utilized different datasets, and different numbers of an ambiguous word. Regarding Afaan Oromo, author [38] concluded that window sizes ±1 and ±2 perform better in general and window size ±1 in particular. Window sizes ±2 is recommended for Afaan Oromo with 74.6 % F1-measure according to [7]. The third finding shows that ±3 context window size is enough for Afaan Oromo with 63.95% accuracy [37]. Another, finding shows that ±4 window sizes of offered 81% F1-measure accuracy as [34]. In general, there is no common agreement on context window size to disambiguate a target word, which is emerged from unavailability of standard datasets particularly for performance evaluation. Hence, it needs additional research, aims at possible concern of handling all context modifiers without window size enforcement.

This section, briefly discuss an evaluation made regarding to proposed method that minimize the gap of window

size consideration for WSD in Afaan Oromo. The researcher employs various evaluation metrics on the test data and compares the results of the system with a human judge to know the rate of the system disambiguation on 249 test sentences. The metrics are precision, recall, and F-measure, of the test dataset. Therefore, the evaluation determines words that are correctly disambiguated and words that are wrongly recognized or left unrecognized. The evaluation criteria were based on the number of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). TP and FP are counts the number of words that are recognized correctly and recognized wrongly by the WSD system respectively. TN counts the number of words that are left unrecognized correctly by the algorithm and are not in the test. FN counts the number of words that are left unrecognized wrongly by the WSD system. From the experiment, the researcher realized that FN happens when many ambiguous words occur in the given sentence. Therefore, for each metric, the respective formula is present as the next from equations 5 to 7.

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

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

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

Table 2. The detailed result of test data

	Recognized		Unrecognized
Correct	170		20
Incorrect	59		22

The evaluation result in Table 2 shows that 170 and 20 target words are TP and TN respectively, which is labeled the correct category. Additionally, 59 target words are FP (tagged wrong sense) 22 are FN (unrecognized wrongly). The algorithm fails to detect more than 1 ambiguous word per sentence. Meaning, in 22 sentences the number of ambiguous words or their frequency occurred more than 1 in a sentence.

The system performance is in precision, recall, and F-measure is 88.54%, 74.23%, and 80.76% respectively. The conducted experiment shows that the semantic meaning of words is closely connected to the words which are come in the same situation. As shown, the result obtained by the normalized statistical algorithm was ample as the semantic information extracted from the given text. That the approach relies upon automatically assign sense to words, more reliable and proves to be a most useful sense extraction for word sense disambiguation on its part.

5. Conclusion

Clustering is one of the well-known unsupervised approaches for WSD, which focuses on grouping similar semantic information in a cluster. Besides that, the contribution of this paper is the way of sense disambiguation is novel that depends on calculating pair word co-occurrence degree of all pair words in a text from a user with every pair word in respective datasets. In this way, there is no notion of any predefined context window size utilized. From this viewpoint, all words come with an ambiguous word has involved in sense disambiguation and also, the co-occurrence degree is under consideration (an adjacent one is given the highest value).

Interestingly, clustering words per sense performance is promising based on sense-specific word co-occurrence and despite stop-word removal and stemming the text. To conclude the proposed algorithm has the power to discriminate ambiguous word's sense for a sense disambiguation purpose. Moreover, the algorithm detects semantic word relation through statistical information that is normalized based on word frequency besides, the distance between each word in pair. The performance achieved in precision, recall, and F-measure are 88.54%, 74.23%, and 80.76% respectively.

The weakness of the algorithm is it needs a training dataset that requires documents per sense grouped, which is not available for under resource language. Before implementing a sense cluster algorithm, each sense file contains the list of statements that should be ready for training and each folder contains multiple sense files of a single ambiguous word. It does not work for any ambiguous word that is not mentioned in the corpus.

The strength is, once the training data has been prepared, the execution time of the algorithm as well as tagging sense for user input text, is fast with promised output. Indeed, to the best of my knowledge, this is effortless sense clustering except, dataset development (grouping sentences of similar sense into together). Moreover, no need for sense tagging and no need for further text preprocessing except tokenization and removal of specified punctuation marks. Additionally, a language exhibiting similar patterns with Afaan Oromo can adapt the algorithm. Specifically, it increases the methodology of the WSD research. The proposed algorithm can be extended for semantic text similarity

without modification or with a bit modification. Furthermore, it has been pointed out how NLP plays a significant role in enhancing the computer's capability to process word senses.

6. Future Works

Adding several concerned ambiguous words and their senses are one of the forwarded direction regarding the algorithms. Furthermore, morphological analyzer like performing stop-word removal and stemming or lemmatization then comparing and evaluating the performance is expected to boost the effectiveness of word sense disambiguation. The proposed algorithm can be extended for semantic text similarity without modification or with a bit modification.

References

[1] Nadia Bouhriz, Faouzia Benabbou, and El Habib Ben Lahmar. “Word Sense Disambiguation Approach for Arabic Text” International Journal of Advanced Computer Science and Applications, Vol. 7, No. 4, 2016.

[2] Workineh Tesema and Duresa Tamirat, “Investigating Afan Oromo 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.

[3] Jumi Sarmah, Shikhar Kumar Sarma,"Survey on Word Sense Disambiguation: An Initiative towards an Indo-Aryan Language", International Journal of Engineering and Manufacturing, Vol.6, No.3, pp.37-52, 2016.

[4] Michele Bevilacqua and Roberto Navigli. “Breaking Through the 80% Glass Ceiling Raising the State of the Art in Word Sense Disambiguation by Incorporating Knowledge Graph Information” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2854–2864 July 5 - 10, 2020.

[5] Shweta Vikram, Sanjay K. Dwivedi, "Ambiguity in Question Paper Translation", International Journal of Modern Education and Computer Science, Vol.10, No.1, pp. 13-23, 2018.

[6] Workineh T, Debela T, Teferi K; “Designing a Rule Based Disambiguator for Afan Oromo Words”. Am J Compt Sci Inform Technol, 2017.

[7] Workineh Tesema, Debela Tesfaye and Teferi Kibebew, “Towards the sense disambiguation of Afan Oromo words using hybrid approach (unsupervised machine learning and rule based).” Ethiopian Journal of Education and Sciences 12 (2016): pp. 61-77.

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

[9] Tamene Keneni Walga “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.

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

[11] Getachew Rabirra Furtuu, “Seerluga Afaan Oromoo”, Finfinnee Oromiyaa press, 2014.

[12] Kula Kekeba Tune, Vasudeva Varma, Prasad Pingali, Evaluation of Oromo- English Crosslanguage Information Retrieval, ijcai 2007 workshop on clia, hyderabad, india, 2007.

[13] Abdo Ababor Abafogi, "Boosting Afaan Oromo Named Entity Recognition with Multiple Methods", International Journal of Information Engineering and Electronic Business, Vol.13, No.5, pp. 51-59, 2021.

[14] Baskaran Sankaran, k. Vijay-Shanker, “Influence of morphology in word sense disambiguation for Tamil”, Anna University and University of Delaware Proceedings of International Conference on Natural Language Processing, 2003.

[15] Yinglin Wang, Ming Wang, Hamido Fujitas, Word Sense Disambiguation: A comprehensive knowledge exploitation framework Knowledge-Based Systems vol 190, 29 February 2020.

[16] David Yarowsky. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pages 88–95. Association for Computational Linguistics, 1994.

[17] Bianca Scarlini, Tommaso Pasini, and Roberto Navig, “SENSEMBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation,” The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20) 2020, Association for the Advancement of Artificial Intelligence.

[18] Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. Improved Word Sense Disambiguation using pre-trained contextualized word representations. In Proc. Of EMNLP, pages 5297–5306, 2019.

[19] Michele Bevilacqua and RobertoNavigli. Quasi bidirectional encoder representations from Transformers for Word Sense Disambiguation. In Proc. of RANLP, pages 122–131, 2019.

[20] Federico Scozzafava, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and Roberto Navigli. Personalized PageRank with syntagmatic information for multilingual Word Sense Disambiguation. In Proc. of ACL (demos), 2020.

[21] Michele Bevilacqua and Roberto Navigli. Breaking through the 80% glass ceiling: Raising the state of the art in Word Sense Disambiguation by incorporating knowledge graph information. In Proc. of ACL, pages 2854–2864, 2020.

[22] Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. Zero-shot Word Sense Disambiguation using sense definition embeddings. In Proc. of ACL, 2019.

[23] Terra Blevins and Luke Zettlemoyer. Moving down the long tail of Word Sense Disambiguation with gloss informed bi-encoders. In Proc. of ACL, 2020.

[24] Edoardo Barba, Tommaso Pasini, and Roberto Navigli. ESC: Redesigning WSD with extractive sense comprehension. In Proc. of NAACL, 2021.

[25] Simone Conia and Roberto Navigli. Framing Word Sense Disambiguation as a multi-label problem for model-agnostic knowledge integration. In Proc. of EACL, 2021.
[26] Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato and Roberto Navigli "Recent Trends in Word Sense Disambiguation: A Survey" Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)
[27] Vial, L.; Lecouteux, B.; and Schwab, D. 2019. Sense Vocabulary Compression through the Semantic Knowledge of WordNet for Neural Word Sense Disambiguation. In Proc. of Global Wordnet Conference.
[28] Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017. Neural sequence learning models for word sense disambiguation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1156–1167, Copenhagen, Denmark. Association for Computational Linguistics.
[29] B. H. Manjunatha Kumar, B.E., M.Tech.A Survey on Word Sense Disambiguation Sri Siddhartha Institute of Technology, 2018.
[30] Sruthi Sankar K P, P C Reghu Raj, Jayan V U, nsupervised Approach to Word Sense Disambiguation in Malayalam," International Conference on Emerging Trends in Engineering, Science and Technology, 2015.
[31] Iacobacci, I.; Pilehvar, M. T.; and Navigli, R. 2016. Embeddings for word sense disambiguation: An evaluation study. In Proc. Of ACL, volume 1, 897–907.
[32] Melamud, O.; Goldberger, J.; and Dagan, I. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In Proc. of CoNLL, 51–61.
[33] Krishnanjan et al. "Survey and Gap Analysis of Word Sense Disambiguation Approaches on Unstructured Texts", Proceedings of the International Conference on Electronics and Sustainable Communication Systems 2020.
[34] 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, Volume 3, Issue 10, ISSN: 2349-6002, March 2017.
[35] Huang Heyan, Yang Zhizhuo, and Jian Ping, "Unsupervised Word Sense Disambiguation Using Neighbourhood Knowledge," Beijing Engineering Applications Research Center of High Volume Language Information Processing and Cloud Computing, Beijing Institute of Technology and Department of Computer Science, Beijing Institute of Technology, China.
[36] Tesfa Kebede. Word sense disambiguation for Afaan Oromo Language: published master's Thesis, Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia, 2013.
[37] Shibiru Olika, "word sense disambiguation for afaan oromo using knowledge base" St. University College, 2018.
[38] Yehuwalashet Bekele. Hybrid Word Sense Disambiguation Approach for Afaan Oromo Words: published master's Thesis, Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia, 2016.

Authors' Profiles

Abdo Ababor Abafogi: Received his BSc degree and MSc degree 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 and Natural Language Processing, SEO.

How to cite this paper: 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