Journal of
Information Technology & Software Engineering

Tesema, J Inform Tech Softw Eng 2016, 6:5
DOI: 10.4172/2165-7866.1000191

Review Article Open Access

Afan Oromo Sense Clustering in Hierarchical and Partitional Techniques

Workineh Tesema*

Department of Information Science, Jimma University, Jimma, 378, Ethiopia

Abstract

This paper presents the sense clustering of multi-sense words in Afan Oromo. The main idea of this work is to cluster contexts which is providing a useful way to discover semantically related senses. The similar contexts of a given senses of target word are clustered using three hierarchical and two partitional clustering. All contexts of related senses are included in the clustering and thus performed over all the contexts in the corpus. The underlying hypothesis is that clustering captures the reflected unity among the contexts and each cluster reveal possible relationships existing among the contexts. As the experiment shows, from the total five clusters, the EM and K-Means clusters which yield significantly higher accuracy than hierarchical (single clustering, complete clustering and average clustering) result. For Afan Oromo, EM and K-means enhance the accuracy of sense clustering than hierarchical clustering algorithms. Each cluster representing a unique sense. Some words have two senses to the five senses. As the result shows an average accuracy of test set was 85.5% which is encouraging with the unsupervised machine learning work. By using this approach, finding the right number of clusters is equivalent to finding the number of senses. The achieved result was encouraging, despite it is less resource requirement.

Keywords: Hierarchical clustering; Partitional clustering; Ambiguous; Algorithms; Clustering; Machine; K-means; Sense

Introduction

One of the most critical task in natural language processing (NLP) application is semantic. Most of words in natural language have multiple senses that can only be determined by considering the context in which it occur [1]. Given instances of a target word used in a number of different contexts, word sense disambiguation is the process of grouping these instances into clusters that refer to the same sense. Approaches to this problem are often based on the strong contextual hypothesis of [2], which states that two words are semantically related to the extent that their contextual representations are similar. Hence the problem of word sense disambiguation reduces to that of determining which contexts of a given target word are related or similar. Sense Clusters creates clusters made up of the contexts in which a given target word occurs [3]. All the instances in a cluster are contextually similar to each other, making it more likely that the given target word has been used with the same sense in all of those instances. Each instance normally includes two or three sentences, one of which contains the given occurrence of the target word [4]. Sense Clusters [1] was originally intended to discriminate among word senses. However, the methodology of clustering contextually (and hence semantically) similar instances of text can be used in a variety of natural language processing tasks such as synonymy identification, text summarization and document classification. Sense Clusters has also been used for applications such as email sorting and automatic ontology construction [5].

Related Work

The state of the art in sense clustering is insufficient to meet the needs where there is lack of sense definitions like Word Net. Current sense clustering algorithms are generally unsupervised, each relying on a different set of useful features. Hierarchical algorithms produce a nested partitioning of the data elements by merging clusters. Agglomerative algorithms iteratively merge clusters until all-encompassing cluster is formed [6], while divisive algorithms iteratively split clusters until each element belongs to its own cluster. The merge and split decisions are based on the similarity metric. The resulting decomposition (tree of clusters) is called a dendrogram. The different versions of agglomerative clustering differ in how they compute cluster similarity. The most common versions of the agglomerative clustering algorithm are [7]:

Single link clustering

The single link algorithm is a MIN version of the hierarchical agglomerative clustering method which is a bottom-up strategy, compare each point with each point. Each context is placed in a separate cluster, and at each step merge the closest pair of clusters, until certain termination conditions are satisfied. For the single link, the distance of two clusters is defined as the minimum of the distance between any two points in the clusters. In single-link clustering the similarity between two clusters is the similarity between their most similar members for example using the Euclidean distance [8].

Complete link clustering

The complete linkage algorithm is the MAX version of the hierarchical agglomerative clustering method which is a bottom-up strategy: compare each point with each point. Each context is placed in a separate cluster, and at each step merge the farthest pair of clusters, until certain termination conditions are satisfied. In complete-link clustering, the similarity between two clusters is the similarity between their maximum similar members for example using the Euclidean distance [9].

Average link clustering

Average-link clustering produces similar clusters to complete link clustering except that it is less susceptible to outliers [4]. It computes the similarity between two clusters, as the average similarity between all pairs of contexts across clusters (e.g. using the Euclidean distance). Figure 1 shows merging decisions single, complete and average linkage algorithms.

***Corresponding author:** Workineh Tesema, Department of Information Science, Jimma University, Jimma, 378, Ethiopia, Tel: +2510471112233; E-mail: workina.info@gmail.com

Received October 17, 2016; **Accepted** November 04, 2016; **Published** November 10, 2016

Citation: Tesema W (2016) Afan Oromo Sense Clustering in Hierarchical and Partitional Techniques. J Inform Tech Softw Eng 6: 191. doi: 10.4172/2165-7866.1000191

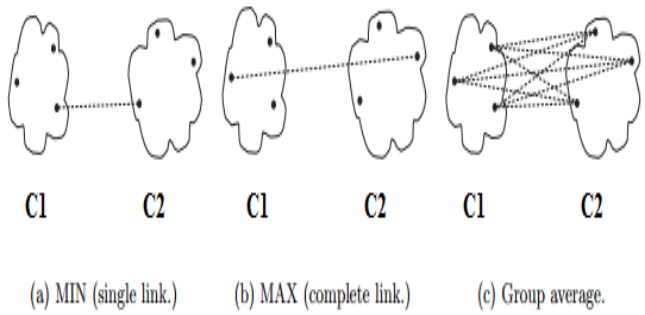

Figure 1: a) Single Link, b) Complete Link and c) Average Link

The other type of clustering used in this work is Partitional clustering. Partitional algorithms do not produce a nested series of partitions. Instead, they generate a single partitioning, often of predefined size *k*, by optimizing some criterion. A combined search of all possible clustering's to find the optimal solution is clearly intractable. The algorithms are then typically run multiple times with different starting points. Partitional algorithms are not as versatile as hierarchical algorithms, but they often offer more efficient running time [4].

K-means: This algorithm has the objective of classifying a set of *n* contexts into *k* clusters, based on the closeness to the cluster centers. The closeness to cluster centers is measured by the use of a Euclidean distance algorithm. K-means is an iterative clustering algorithm in which items are moved among sets of clusters until the desired set is reached. A high degree of similarity among senses in clusters is obtained, while a high degree of dissimilarity among senses in different clusters achieved simultaneously [4]. K-means clustering [10] is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. K-means [11] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume *k* clusters) fixed a priori. The following steps outline the algorithm for generating a set of *k* clusters:

- Randomly select K elements as the initial centroids of the clusters;
- Assign each element to a cluster according to the centroid closest to it;
- Recomputed the centroid of each cluster as the average of the cluster's elements;
- Repeat Steps 2-3 for T iterations or until a criterion converges, where T is a predetermined constant.

Expectation maximization (**EM**): is also an important algorithm of data mining [12]. An Expectation maximization (EM) algorithm is an iterative method for finding maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM [12] iteration alternates between performing an expectation which computes the expectation of the log-likelihood evaluated using the current estimate for the parameters, and maximization which computes parameters maximizing the expected log-likelihood found. These parameter-estimates are then used to determine the distribution of the latent variables [13].

The rest of this paper will proceed as follows. Section 3 will discuss the different aspects of the proposed approach. Section 4 presents result, discussion and performance evaluation of the system. Finally a conclusion is presented in section 5.

Methodology

In our approach, two important features need to be extracted: the first one is determining all possible contexts (the candidate sense words) of the target words and the other one is to group these various contexts (senses) of the word, each group representing a specific sense of the target word. To this end, the developed approach towards the word sense disambiguation is completely machine learning in its nature. Unsupervised machine learning approach extracts the two important features (the various contexts of the target words and their clustering). In this approach feature of Afan Oromo with the semantic feature learned from corpus. Hence we didn't provide explicit sense labels for each group as the machine learning approach is unsupervised. Yet, small list of target words are required to test the algorithm. As already mentioned, the context terms of the target words clustered using their similarity values produced. The clustering algorithms have their own unique nature. The hierarchical clustering begin by assuming that each context of an target word forms its own cluster (and therefore represents a unique sense). Then, it merges the contexts that have the minimum dissimilarity between them (and are therefore most alike). The partitional clustering algorithms started by partitioning into predefined *k* sizes [14]. It found the one which is the nearest to initial centroid. A centroid is usually not an element of the cluster. Rather, it represents the center of all other elements. The minimum specified cutoff which determines the number of clusters is taken. In this case, the minimum specified cutoff of the number of clusters is two hence one target word has at least two senses.

Sense clustering

Our approach for learning how to merge senses relies upon the availability of unlabeled judgments of sense relatedness. Sense Clusters distinguishes among the different contexts in which a target word occurs based on a set of features that are identified from raw corpora. Sense Clusters currently supports the use of N-grams (like unigram, bigram), and co-occurrence features. Unigrams are individual words that occur above a certain frequency cutoff. These can be effective discriminating features if they are shared by a minimum of two contexts, and shared by all contexts. Very common non-content words are excluded by providing a stop-list. Co-occurrences are unordered word pairs that include the target word. In effect co-occurrences localize the scope of the N-gram features by selecting only those words that occur within some number of positions from the target word.

Sense Clusters provides support for a number of similarity measures, such as the cosine. A similarity matrix created by determining all pairwise measures of similarity between contexts can be used as an input to Weka tool clustering algorithms or to Sense Clusters own agglomerative and partitional clustering implementation.

Given a set of N items to be clustered and an N x N similarity matrix, the basic process of clustering is this:

- Start by assigning each item to its own cluster, so that if we have N items, we now have N clusters, each containing just one item. Let the similarities between the clusters equal the similarities between the items they contain.
- Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now we have one less cluster.
- Compute similarities between the new cluster and each of the old clusters.
- Repeat steps i and ii until all items are clustered into a set of different clusters of size N.

Results and Discussion

As the conducted experiment showed, each clusters have a context group, where the sense of these context groups are hopefully different. The underlying assumption is that the senses found in similar contexts are similar senses. Then, new occurrences of the context can be classified into the closest induced clusters (senses). All contexts of related senses are included in the clustering and thus performed over all the contexts in the corpus [14]. The underlying hypothesis is that target word contexts clustering (Figure 2) captures the reflected unity among the contexts and each cluster reveal possible relationships existing among these contexts. The test by our method, that deals with clustering of contexts for a given word that express the same sense. The simple K-Means and EM clustering algorithms achieved much accuracy on the task of WSD for selected target word. The partitional clustering which include K-means and EM resulted 71.2% and 74.6% respectively achieved performance in clustering (Table 1).

An important point here is how to decide which constitutes good clustering, since it is commonly acknowledged that there is no absolute best criterion which would be independent of the final aim of the clustering. Consequently, it is the researcher who supply the criterion that best suits their particular needs and the result of the clustering algorithm can be interpreted in different ways. One approach is to group data in an exclusive way, so that if a certain item of data belongs to a definite cluster, then it could not be included in another cluster. Another approach, so-called overlapping clustering, uses unclear sets of cluster data in such a way that each item of data may belong to two or more clusters with different degrees of membership. The Figure 3 Dendrogram shows the more description of results

Initially, we evaluated our WSD method with all the 15 natural words. This lead, to a total of 15 natural words tested in this evaluation, and these target words have two senses to five senses. Six terms have two senses (the terms with two senses are *afaan, boqote, dubbatate, haare, ji'a, lookoo*), and six terms have three senses (the terms with three senses are *diige, tume, handhuura, dhahe, mirga, waraabuu*) and two terms have five senses (the terms with five senses are *bahe, ija*) and the left one has four senses (the term with four senses is *darbe*) out of

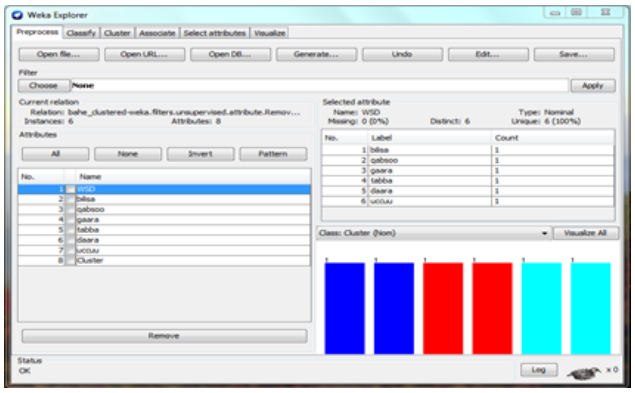

Figure 2: Clustering.

No	Clustering Algorithms	Accuracy (%)
1	Single Link	61%
2	Complete Link	59.70%
3	Average Link	61%
4	K-Means	71.20%
5	EM	74.60%

Table 1: Unsupervised machine learning results.

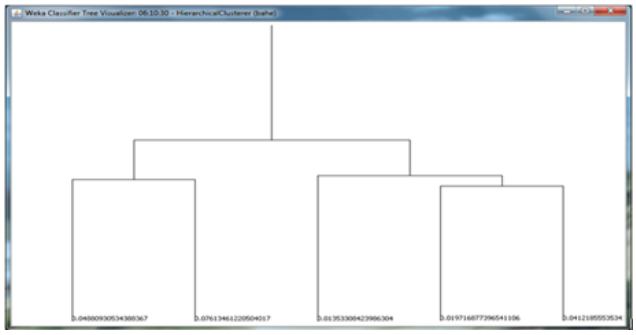

Figure 3: Dendrogram.

15 target terms [15].

As evident from the visualization Figure 4, the output has been classified into 3 correct clusters out of total of 5 clusters using EM and K-means clustering.

Evaluation procedures

On the other hand, the clustering algorithms were evaluated comparing the result produced by the clustering algorithm with the manually grouped similar contexts of the target words in the test set by experts. The evaluation constitutes the following two points:

1. To evaluate how much the produced clusters are comply with the clusters prepared by human experts as a benchmark. In order to achieve this we used the following criteria:

- How many of the clustered contexts are correct, i.e. to evaluate if all the similar contexts of the target words are placed in the same group.

2. Given the number of senses assumed by the target words in the test, judge the system on the basis of the number of senses identified by the system. Similarly, in order to achieve this the following steps performed:

- Start with a small list of target words in the test with known number of senses N.
- Run the algorithm on the test to identify the possible senses based on it's the number of clusters of the context as extracted from the big corpus
- Count the number of clusters
- Compare it against the already prepared sense clusters by experts

Conclusion and Future Work

The overall focus of this research is to investigate Word Sense Disambiguation which addresses the problem of deciding the correct sense. To this end, we relied on clustering technique which is to group related context words. There are several types of clustering algorithms. In this paper we relied on hierarchical and probabilistic algorithms. We did experiments on five different clustering algorithms namely K-Means, EM, single, complete and average link. Based on the result of the experiment out of the five algorithms simple K-Means and EM algorithms are the best of all to identify the sense of target word in a context. We believe that the observed poor performance of hierarchical agglomerative algorithms [16] is because of the errors they make during early agglomeration. This work can be a base for this further research and it can support extended disambiguation covering most of the terms in the Afan Oromo.

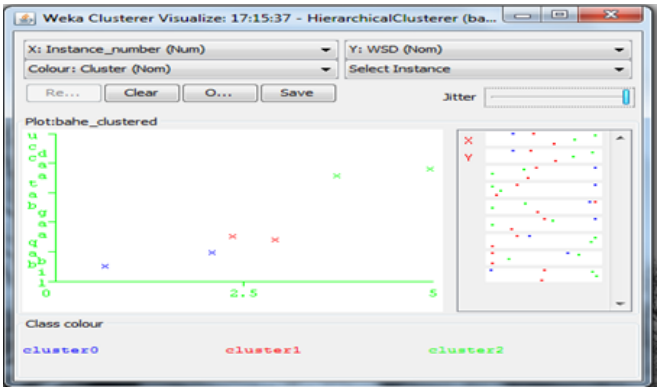

Figure 4: Visualize Clusters of EM and K-Means clustering.

Acknowledgment

I would like to thank my mother Askala Jifar and all my family for their morale and financial support. Secondly, I would like to thank my lovely Sister Miss Sorse Tesema and Worku Jimma (PhD student) and all my colleagues.

References

1. Yarowsky D (2007) Unsupervised word sense disambiguation rivaling supervised methods. Computational Linguistics, Cambridge, M.A 189-196.
2. Miller G and Charles W (2001) Contextual correlates of semantic similarity. Language and Cognitive Processes 6: 1-28.
3. Yarowsky D, Florian R (2002) Evaluating sense disambiguation across diverse parameter spaces. Journal of Natural Language Engineering 8: 293-310.
4. Xie J, Jiang S, Xie W, Gao X (2011) An Efficient Global K-means Clustering Algorithm. Journal of Computers 6: 271-279.
5. Roberto N (2009) Word Sense Disambiguation: A Survey. Journal ACM Computing Surveys 41.
6. Sneath PHA and Sokal RR (2013) Numerical Taxonomy: The Principles and Practice of Numerical Classification. CABI London UK: Freeman 573.
7. Kilgarriff A (1997) I don't believe in word senses. Computers and Humanities 31: 91-113.
8. King B (2001) Step-wise clustering procedures. Journal of the American Statistical Association.
9. Jain AK, Dubes RC (1998) Algorithms for Clustering Data, Prentice-Hall.
10. Murty MN, Jain AK, Flynn PJ (2009) Data clustering: a review. ACM Computing Surveys 31: 264-323.
11. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39: 1-38.
12. Celeux G, Govaert G (2002) A classification EM algorithm for clustering and two stochastic versions. Computational statistics and data analysis 14: 315-332.
13. Han J, Kamber M (2001) Data Mining-Concepts and Techniques. Morgan Kaufmann.
14. Shao F and Yanjiao C (2005) A New Real-time Clustering Algorithm. Linguistic Studies in Honour of Jan Svartvik, London, Longman.
15. Pedersen T, Bruce R (1997) Distinguishing word senses in untagged text. In Empirical Methods in Natural Language Processing, New York Routledge.
16. Xu R, Wunsch D (2005) Survey of Clustering Algorithms. Journal of IEEE Transactions on Neural Networks 16: 645-678.