Arguably at SemEval-2023 Task 11: Learning the disagreements using unsupervised behavioral clustering and language models

Guneet Kohli
Thapar University
guneetsk99@gmail.com

Vinayak Tiwari
Netaji Subhas University of Technology
vinayaktiwari2897@gmail.com

Abstract

We describe SemEval-2023 Task 11, on behavioral segregation of annotations to find the similarities and contextual thinking of a group of annotators. We have utilized a behavioral segmentation analysis on the annotators to model them independently and combine the results to yield soft and hard scores. Our team focused on experimenting with hierarchical clustering with various distance metrics for similarity, dissimilarity, and reliability. We modeled the clusters and assigned weightage to find the soft and hard scores. Our team was able to find out hidden behavioral patterns among the judgments of annotators after rigorous experiments. The proposed system is made available [1]

1 Introduction

In the field of natural language processing (NLP), it has been assumed that a given natural language expression (such as a sentence) has a single, clear and unambigous meaning in a particular context. However, this assumption is now being recognized as an idealization or simplification that does not always hold in practice. For example, the sentence "I saw her duck" could mean that someone observed a woman's pet duck, or that someone watched a woman quickly lower her head to avoid hitting it on a low object. In these cases, the interpretation of the sentence depends on the context and the particular meanings of the words used.

The Learning with Disagreement shared task (Leonardellli et al., 2023) aims to provide a testing framework for machine learning models that can learn from disagreements in natural language interpretation. Our approach involves developing a model that can take into account multiple possible interpretations of a single sentence and identify the most likely one based of available context and information.

We have utilized unsupervised learning approach and experimented using various clustering techniques and different types of similarities of annotations and metadata to bring up associations and relationships among various groups of annotators. This could tell us how a specific group of annotators think given a particular scenario.

2 Background

This shared task provides a testing framework for learning from disagreements using datasets containing information about disagreements for interpreting language. The focus of this task is on subjective tasks, where training with aggregated labels is less effective. There are four textual datasets with different characteristics, including social media and conversation genres, English and Arabic languages, and tasks related to misogyny, hate speech, and offensiveness detection. These datasets also use different annotation methodologies, including experts, specific demographic groups, and AMT-crowd, and all provide multiple labels for each instance.

The Le-Wi-Di dataset is a collection of four existing datasets that have been harmonized into a common json format that emphasizes their commonalities. These datasets include the HS-Brexit dataset(Akhtar et al., 2021), which is a new dataset of tweets on abusive language related to Brexit that has been annotated for hate speech, aggressiveness, and offensiveness by six annotators from two distinct groups. The ArMIS dataset(Almanea and Poesio, 2022) is a dataset of Arabic tweets annotated for misogyny and sexism detection by annotators with different demographic characteristics. The ConvAbuse dataset(Curry et al., 2021) is a dataset of English dialogues between users and conversational agents, annotated for abuse by at least three experts in gender studies. The MultiDomain Agreement dataset(Leonardelli et al., 2021) is a dataset of English tweets from three domains, annotated for offensiveness by five annotators via

[1] **https://github.com/guneetsk99/LeWiDi_Arguably**

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 2137–2142

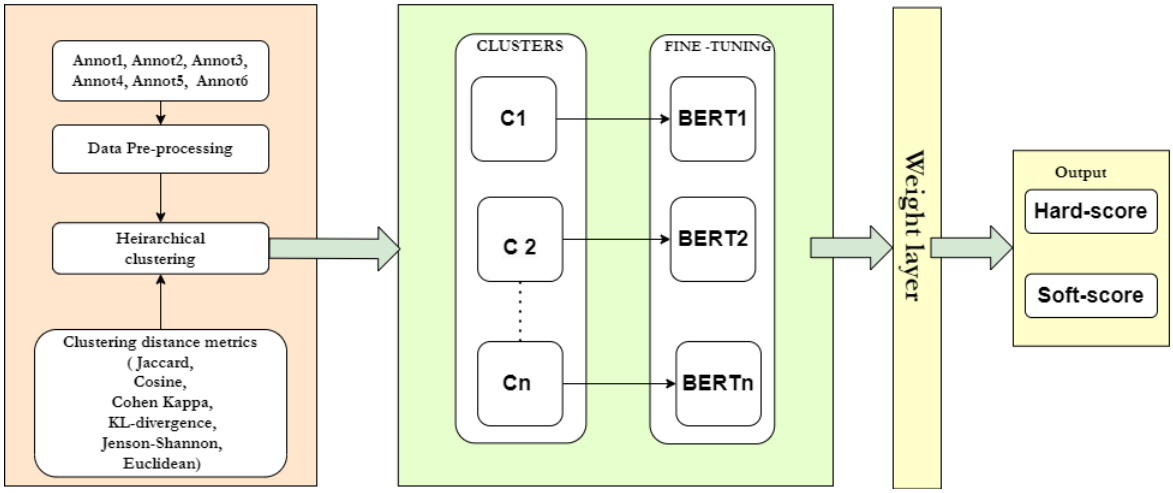

Figure 1: Our proposed system architecture

AMT.

We have gone through some of the related works dealing with subjective tasks in NLP domain.A Survey of Natural Language Processing for Social Media" by (Sun et al., 2017). This survey paper provides an overview of NLP techniques for social media analysis, including sentiment analysis, emotion detection, and opinion mining. Another paper by (Zhang et al., 2018) examines the state-of-the-art approaches for detecting abusive language in online user content, including hate speech, cyberbullying, and harassment. "Offensive Language Detection: A Review" by (Razavi et al., 2010). This review paper summarizes the current research on offensive language detection, including different types of offensive language and the challenges associated with detecting them. "A Survey on Sentiment Analysis in Social Media" by (Yue et al., 2019). This survey paper reviews the current research on sentiment analysis in social media, including the challenges of analyzing informal and unstructured text in social media platforms.

3 System Overview

We have focused on behavioral segregation of annotations that had been provided to us in data-set. Using this information we then experimented with various unsupervised clustering techniques and then passed the output to BERT transformer(Devlin et al., 2018) that gave us the predictions for each group. We have observed that assigning weights to each cluster worked well in our case because that indicated the contributing nature of a particular group. Organisers had mentioned that there are two groups of annotators, hence we were curious to know the unity of each group. After doing various experiments we had found that muslim annotators had a less bias so we have kept the cluster with muslim annotators with more weights i.e 2 X (control group). This helped in providing higher soft-score and better hard-score.

3.1 Dataset

For our experiments, we have utilized the HS-Brexit dataset. Which is an entirely new dataset of tweets on Abusive Language on Brexit and annotated for hate speech (HS), aggressiveness and offensiveness by six annotators belonging to two distinct groups: a target group of three Muslim immigrants in the UK, and a control group of three other individuals.

3.2 Heirarchical Clustering

Hierarchical clustering(Cohen-Addad et al., 2019) is a commonly used technique in data analysis for exploring and grouping similar data points based on their pairwise similarity or distance. It can be used for various types of data, including binary data. We have particularly useful for binary data because it allows for the identification of clusters based on similarity in binary patterns. In other words, it can group together binary data points that share a similar pattern of 0s and 1s. In hierarchical clustering, the binary data points are first represented as binary vectors or matrices. Then, a similarity or distance metric is calculated between each pair of vectors, which can be used to build a dendrogram or a tree-like structure of clusters. The similarity

metric used for binary data is often the Jaccard similarity coefficient, which measures the proportion of shared 1s and 0s between two binary vectors.

3.3 Distance metrics

We had kept hierarchical clustering as the key clustering algorithm and experimented with different similarity metrics such as cosine similarity, Jaccard, Kappa and Euclidean distance. There is a specific reason to opt for these distances. We have used the Jaccard coefficient because it is commonly used for binary data and our annotations data is in binary format. It is defined as

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

We had experimented utilised cosine similarity as well which is defined as a measure of similarity between two vectors in a high-dimensional space. It is commonly used for text data, where each document is represented as a vector of word frequencies. The cosine similarity between two vectors is defined as the cosine of the angle between them, and it ranges from -1 to 1, with 1 indicating complete similarity and -1 indicating complete dissimilarity.

The formula for cosine similarity between two vectors x and y is:

$$cosinesimilarity = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2}$$

Cohen kappa coefficient or Kappa similarity is a statistical measure of inter-rater agreement for categorical data which is commonly used in areas of medicine, education and to assess the level of agreement between two or more raters or judges. This made us utilise this coefficient as well. The formula for kappa similarity is:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{3}$$

We have also used Kullback-Leibler (KL) divergence metric, also known as relative entropy, is a measure that tells how different two probability distributions are from each other. It is commonly used in clustering to compare the similarity between two sets of data points or distributions.KL divergence between two discrete probability distributions P and Q

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{4}$$

The Jensen-Shannon (JS) divergence metric is a symmetrical measure of the similarity between two probability distributions. It is commonly used in clustering to compare the similarity between two sets of data points or distributions.

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M) \tag{5}$$

where D_{KL} is the Kullback-Leibler divergence metric and M is the average of P and Q, defined as:

$$M = \frac{1}{2}(P + Q) \tag{6}$$

3.4 Fine tuned transformer

The preprocessed data undergoes feature extraction, followed by sentence tokenization and mapping of tokens to their respective word IDs. Each sentence is then prepended with a [CLS] token, appended with a [SEP] token, and padded or truncated based on the maximum sequence length determined by the average text length in the dataset. The attention mask is also mapped for each sentence. The resulting sequence and attention mask are encoded to generate contextually rich embeddings passed through BERT Transformers. [2]

3.5 Cluster Distribution Analysis

In this section, we report the clusters generated using different distance metrics and discuss distinctive behavioral patterns that the clusters exhibit, such as Reliability [Cohen Kappa], Similarity [Jaccard, Cosine], and Dissimilarity [KL-Divergence and Jenson Shannon].

3.5.1 Euclidean Cluster

The use of euclidean distance for clustering produced 3 different clusters. Cluster 1 [Ann1, Ann2, Ann3] combined the non-Muslim annotators and divided the Muslim annotators into 2 groups.

3.5.2 Similarity metric Cluster

Both Jaccard and Cosine distances were used to cluster the annotated data. The resulting clusters showed similar behavior: it grouped the Muslim annotators into one cluster while dividing the non-Muslim annotators into two separate clusters. This division of non-Muslim annotators into two clusters suggests the presence of a particular cluster that

[2] https://huggingface.co/docs/transformers/model_doc/bert

Clustering Distance Metric	Cluster 1	Cluster 2	Cluster 3	Weights of clusters
Euclidean	Annotator 1, Annotator 2, Annotator 3	Annotator 4, Annotator 5	Annotator 6	[3,4,2]
Jaccard	Annotator 1	Annotator 2, Annotator 3	Annotator 4, Annotator 5, Annotator 6	[1,2,6]
Cosine	Annotator 1, Annotator 2	Annotator 3	Annotator 4, Annotator 5, Annotator 6	[2,1,6]
Cohen Kappa	Annotator 1	Annotator 2	Annotator3, Annotator 4, Annotator 5, Annotator 6	[1,1,7]
KL- Divergence	Annotator 1, Annotator 2, Annotator 3	Annotator 4, Annotator 5, Annotator 6	None	[3,6]
Jenson-Shannon	Annotator 1, Annotator 2, Annotator 3	Annotator 4, Annotator 5, Annotator 6	None	[3,6]

Table 1: The following table highlights the Annotator's segregations observed after applying the clustering mechanism. The weights refer to the weightage given to each clusters output when calculating the Soft and Hard Score

is unbiased and aligns itself with both the Muslim and non-Muslim contexts of hate speech.

3.5.3 Dissimilarity metric Cluster

Jenson-Sannon and KL-Divergence are dissimilar distance metrics that help segregate the annotators completely. Both the metrics produce 2 clusters, one of Muslims and one of the non-Muslims. The absolute segregation tends to overlook the overlapping ideology we observe in the Similarity metrics case and leads to a decrease in performance, as discussed later in the results section.

3.5.4 Reliability metric Cluster

The utilization of the Cohen Kappa metric allows for a more nuanced understanding of the underlying behavior patterns among annotators. Specifically, it clusters one non-Muslim annotator with a set of Muslim annotators, indicating a high level of agreement among these four individuals. Meanwhile, the remaining two non-Muslim annotators are kept separate, highlighting the high variability in the data that would have been overlooked if only hard scores were considered. Thus, the use of Cohen Kappa enables a more sophisticated behavioral segregation of the annotators.

3.5.5 Cluster Weight Allocation

The weight allocation procedure for the clusters followed a standardized approach, whereby twice the weight was assigned to Muslim annotators, while non-Muslim annotators were allocated unit weight. To illustrate the allocation of weights, we refer to Table 1, where we consider the case of the Cohen Kappa Clustering Distance Metric. In this instance, cluster 1 was assigned to Annotator 1, who is a non-Muslim, cluster 2 was assigned to Annotator 2, also a non-Muslim, and cluster 3 comprised Annotators 3, 4, 5, and 6, where only one annotator was non-Muslim, and the remaining three were Muslim. The corresponding weights assigned to each cluster were [1,1,7]. Specifically, the weight of one was assigned to cluster 1, as the corresponding annotator was a non-Muslim. Similarly, the weight of one was assigned to cluster 2, as the corresponding annotator was also a non-Muslim. In contrast, the weight of 7 [1+2+2+2] was assigned to cluster 3, where one annotator was non-Muslim, and the remaining three were Muslim. This weight allocation scheme ensures a fair representation of the contributions of Muslim and non-Muslim annotators in the clustering process.

4 Results and Analysis

4.1 Experimental Setup

The training process involved setting the batch size to four and configuring the AdamW optimizer with a learning rate of 1e-05. The language models were fine-tuned with a token length of 150, and the training data was processed for a total of two epochs.

4.2 Performance comparison and Analysis

The effectiveness of a distance metric is a critical aspect of clustering tasks. Our results demonstrate that using an appropriate distance metric is crucial for better clustering results. In Table 2, we evaluate the performance of the systems submitted for hierarchical clustering using different distance metrics. Our best system was generated using the Cohen Kappa distance metric, which yielded a Hard Score of 0.89 and a soft score of 1.11. We observed that Cohen's kappa is an efficient metric to evaluate the behavior of the annotators as it provides a standardized measure of inter-annotator agreement that

Clustering Distance Metric	Soft Score Eval	Hard Score Eval F1-Score
Euclidean	2.69	0.79
Jaccard	1.48	0.88
Cosine	1.48	0.87
Cohen Kappa	1.11	0.89
KL- Divergence	1.71	0.85
Jenson-Shannon	2.12	0.83

Table 2: The table discusses the various distance metrics used to perform hierarchical clustering of the Annotator's group for their behavioral segregations

accounts for both the degree of understanding and the degree of chance agreement. This helps to generate a better soft score and corresponding hard score.

We also evaluate the effectiveness of the Euclidean metric for our binary labeling data. The Euclidean metric, which is more efficient for continuous data, failed drastically and reported a very high soft score of 2.69 and the lowest hard score of 0.79.

In addition, we experiment with similarity metrics such as Jaccard and Cosine to bring the similar behaving annotators as close as possible. We observed an improvement in results compared to the Euclidean metric, with a Soft Score of 1.48 and a hard score of 0.88 [Jaccard] and 0.87 [Cosine].

On the other hand we also report our evaluation on the dissimilarity metrics like Jenson-Shannon and Kullback-Leibler divergence metrics which focus on diverging the dissimilar annotators as far as possible. The following leads to the generation of only 2 clusters as it avoided the presence of overlapping patterns. Jenson-Shannon produced a soft score of 2.12 and hard score of 0.83, whereas KL-Divergence produce a soft score of 1.71 and a Hard score of 0.85

4.3 Conclusion

This paper proposes a hierarchical clustering method to segregate the annotators into different groups and model them using BERT. We experiment with various distance metrics and try to identify the changes in the clusters corresponding to the metrics' properties. We use similarity, dissimilarity, and reliability metrics and report their results. The Kappa Cohen based clustering method helps us in yielding the best hard score among all our experiments both in terms of the hard score [0.89] and soft score [1.11].

In the future, we aim to experiment further with Large Language Models and use sophisticated clustering techniques to map the intrinsic annotator's behaviors and yield high-performance boots.

References

Sohail Akhtar, Valerio Basile, and Viviana Patti. 2021. Whose opinions matter? perspective-aware models to identify opinions of hate speech victims in abusive language detection. *arXiv preprint arXiv:2106.15896*.

Dina Almanea and Massimo Poesio. 2022. Armis-the arabic misogyny and sexism corpus with annotator subjective disagreements. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2282–2291.

Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. 2019. Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, 66(4):1–42.

Amanda Cercas Curry, Gavin Abercrombie, and Verena Rieser. 2021. Convabuse: Data, analysis, and benchmarks for nuanced abuse detection in conversational ai. *arXiv preprint arXiv:2109.09483*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Elisa Leonardelli, Stefano Menini, Alessio Palmero Aprosio, Marco Guerini, and Sara Tonelli. 2021. Agreeing to disagree: Annotating offensive language datasets with annotators' disagreement. *arXiv preprint arXiv:2109.13563*.

Elisa Leonardellli, Gavin Abercrombie, Dina Almanea, Valerio Basile, Tommaso Fornaciari, Barbara Plank, Massimo Poesio, Verena Rieser, and Alexandra Uma. 2023. SemEval-2023 Task 11: Learning With Disagreements (LeWiDi). In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.

Amir H Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive language detection using multi-level classification. In *Advances in Artificial Intelligence: 23rd Canadian Conference on Artificial Intelligence, Canadian AI 2010, Ottawa, Canada, May 31–June 2, 2010. Proceedings 23*, pages 16–27. Springer.

Shiliang Sun, Chen Luo, and Junyu Chen. 2017. A review of natural language processing techniques for opinion mining systems. *Information fusion*, 36:10–25.

Lin Yue, Weitong Chen, Xue Li, Wanli Zuo, and Minghao Yin. 2019. A survey of sentiment analysis in social media. *Knowledge and Information Systems*, 60:617–663.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 745–760. Springer.