IITK at SemEval-2024 Task 1: Contrastive Learning and Autoencoders for Semantic Textual Relatedness in Multilingual Texts

Udvas Basak[*]   **Rajarshi Dutta**[*]   **Shivam Pandey**[*]   **Ashutosh Modi**

Indian Institute of Technology Kanpur (IIT Kanpur)

{udvasb20, rajarshi20, shivamp20}@iitk.ac.in

ashutoshm@cse.iitk.ac.in

Abstract

This paper describes our system developed for the SemEval-2024 Task 1: Semantic Textual Relatedness. The challenge is focused on automatically detecting the degree of relatedness between pairs of sentences for 14 languages including both high and low-resource Asian and African languages. Our team participated in two subtasks consisting of Track A: supervised and Track B: unsupervised. This paper focuses on a BERT-based contrastive learning and similarity metric based approach primarily for the supervised track while exploring autoencoders for the unsupervised track. It also aims on the creation of a bigram relatedness corpus using negative sampling strategy, thereby producing refined word embeddings.

1 Introduction

The semantic relatedness between texts in a language is fundamental to understanding meaning (Halliday and Hasan, 2014). Automatically detecting relatedness plays an essential role in evaluating sentence representations, question answering, and summarization (Abdalla et al., 2023). The fundamental difference between semantic similarity and relatedness is that semantic similarity only considers paraphrase or entailment relationships. In contrast, relatedness accounts for all commonalities between two sentences, e.g., topical, temporal, thematic, contextual, syntactic, etc. (Abdalla et al., 2023). As highlighted in Table 1, Sentences 1 and 2 are semantically similar, but sentences 2 and 3 would have low semantic similarity but high semantic relatedness.

In Track A (Task 1) of the Semantic Textual Relatedness (STR) task (Ousidhoum et al., 2024b), we are expected to calculate the degree of semantic relatedness between pairs of sentences in 14 different languages covering both African and Asian

#	Sentence
1	The mouse was chased by the cat in the yard.
2	The cat chased the mouse around the garden.
3	The dog barked loudly as the mouse scurried away.

Table 1: Difference between Similarity and Relatedness

languages. Each pair of sentences is assigned a relatedness score in the range of 0 and 1. The major challenge lies in the efficient development of a metric to facilitate the calculation of the relatedness score between the sentence pairs and harnessing the structure of multiple languages to create an efficient model (Ousidhoum et al., 2024b). Our system is based on a contrastive learning approach, utilizing a composite lexical similarity-based measure for relatedness score calculation in the supervised task. Additionally, it involved the use of transformer autoencoders for the unsupervised task. We employed Distill-RoBERTa (Sanh et al., 2020) as the model for this purpose. Several other strategies were also tested within this framework, such as employing a Siamese architecture and retraining BERT with vocabulary expansion to incorporate tokens from additional low-resource languages. For the unsupervised task, the base model used to construct the denoising autoencoder was BERT-uncased (Devlin et al., 2019). The major challenge in this task was the devising and implementing data pre-processing schemes for diverse languages and various training methodologies. A number of experiments were conducted to come up with a unified metric for semantic relatedness calculations, which resulted in relatively better performances in various low-resource languages.[1]

2 Background

There have been several attempts to define and distinguish semantic relatedness from semantic simi-

---

[*] Equal Contributions

---

[1]The code can be found at https://github.com/Exploration-Lab/IITK-SemEval-2024-Task-1-Semantic-Relatedness

larity. The basic metric used in these experiments is Spearman Rank Correlation. The correlation coefficient is calculated between the correctly annotated scores for the set of pairs of sentences and the scores returned by the models. Essentially, this removes the absolute values of the scores and focuses on the relative values and, hence, the relative relatedness between the pairs of sentences.

Initial experiments explored frequency measures such as lexical overlap (Shirude et al., 2021), related words, and related subjects and objects, leading to high Spearman correlations of 0.82 and 0.83 for BERTbase (mean) and RoBERTa-base (mean) on the CompLex dataset (Shardlow et al., 2020). Despite marginal improvements over a lexical overlap baseline, unsupervised models offer limited enhancement.

Normalized Google Distance(NGD) has been used as a novel metric for measuring semantic relatedness between words or concepts (Lopes and Moura, 2019), utilizing Google search result counts to quantify relatedness. NGD (Cilibrasi and Vitanyi, 2007) normalized counts considering the overall corpus size and the co-occurrence of terms in web pages.

Various approaches have been proposed for the Arabic language (Al Sulaiman et al., 2022) like automatic machine translation to translate English Semantic Textual Similarity (STS) data into Arabic, interleaving English STS data with Arabic BERT models, and employing knowledge distillation-based models to fine-tune them using a translated dataset. Multilingual knowledge distillation (Reimers and Gurevych, 2019) techniques have been proposed where a student model, $\hat{M}$, learns from a teacher model, M, on source language sentences and their translations by minimizing the mean-squared loss function. Focusing on low-resource Indian languages, a range of SBERT models has been introduced for ten popular Indian languages. IndicSBERT (Deode et al., 2023) utilized a two-step training method, fine-tuning models using the NLI dataset followed by Semantic Textual Similarity benchmarking (STSb), resulting in substantial improvements in embedding similarity scores and cross-lingual performance.

3 Dataset Description

The dataset (Ousidhoum et al., 2024a) consists of a total of 14 languages, namely Afrikaans, Algerian Arabic, Amharic, English, Hausa, Indone-sian, Hindi, Kinyarwanda, Marathi, Modern Standard Arabic, Moroccan Arabic, Punjabi, Spanish, and Telugu. Every language consists of pairs of sentences with scores representing the degree of semantic textual relatedness between 0 and 1. The scores have been assigned to sentence pairs through a comparative annotation process (Ousidhoum et al., 2024a).

At the preliminary level, dataset length is the only non-semantic variable in these datasets. To assess semantic relatedness, it is crucial to mitigate these biases. Correlation coefficients between sentence lengths and scores fall in the range $-0.13 < \rho < 0.15$; hence, there is no discernible correlation between sentence lengths and scores, suggesting a well-distributed dataset suitable for training.

4 System Overview

Our baseline system of scoring a pair of sentences uses Jaccard Similarity, a lexical metric that calculates the number of token intersections over total tokens in both sentences. Our approach involves using Contrastive learning for the supervised part and auto-encoders for the unsupervised part. All these methods are discussed below, and our model supervised architecture is shown in Figure 1.

4.1 SimCSE

SimCSE (Gao et al., 2022), or Simple Contrastive Learning, is helpful in supervised and unsupervised learning, particularly in information retrieval, text clustering, and semantic tasks. This approach primarily uses Natural Language Inference (NLI) to create positive and negative sentence samples. It works by inducing slight variation in its representations through dropouts. The following step lies in aligning related sentences close in the embedding space and maximizing distances to unrelated sentences in each batch of data. For a supervised setting, it classifies positive samples as entailment pairs, while negative samples are derived from contradiction pairs. The training proceeds via minimizing the loss function: $-\log\left(\frac{e^{\frac{sim(h,h^+)}{\tau}}}{e^{\frac{sim(h,h^+)}{\tau}}+e^{\frac{sim(h,h^-)}{\tau}}}\right)$, where, h represents the current sentence and h^+ and h^- denotes the positive and negative samples respectively with τ being the temperature hyperparameter which controls the sensitivity and learning dynamics. The τ is mainly adjusted based on validation set perfor-
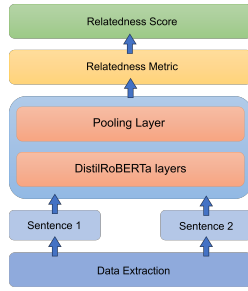
Figure 1: SIMCSE based approach for Track A

mance.

4.2 TSDAE

TSDAE (Wang et al., 2021) or Transformer Denoising through Auto Encoders is an elegant approach aimed at improving the quality of sentence embeddings through self-supervised learning. This method mainly uses sentence embeddings without labels, relying solely on the data structure to initiate learning. The first step lies in corrupting the existing tokens via methods like deleting random tokens, masking tokens, etc., and passing these modified representations through an encoder layer. The encoder layer outputs a dense latent representation, capturing the essence of the data in high-dimensional space. The encoded representations are then passed to a decoder, which attempts to reconstruct the original, uncorrupted sentences from the encoded representations. The decoder is typically another transformer model that has been trained to generate text based on the encoded embeddings. The main objective lies in minimizing the distance between the corrupted and reconstructed sentence representations through cross-entropy loss.

4.3 Training Scheme

For the supervised track, we used Distil-RoBERTa (Sanh et al., 2020) as the base model, which produced a vector representation for every word in the input sentence, resulting in a matrix of token embeddings. The embeddings were then fed to a pooling layer for the production of sentence-level embeddings, which finally proved to be useful for semantics-relatedness tasks. We used mean pooling because there were no dedicated [CLS] token representations for sequence classification tasks. As for the metrics, our approach was involved in designing a custom relatedness metric by combining standard distance-based metrics like cosine similarity, Mahalanobis distance, Euclidean and Manhattan distances, and lexical overlap-based met-

rics like Jaccard and Dice coefficients. For each pair of sentence embeddings in the dataset, we calculated these metrics. Not only did we calculate these metrics using the original embeddings, but we also calculated them after transforming the embeddings by raising them to higher powers (e.g., squaring them). These calculated metrics were then collected into a dataset, with each column named according to the metric and the power applied to the embeddings. For example, the column "Cosine Distance: 2" depicted the cosine distances between pairs of sentence embeddings after both embeddings in each pair have been squared. The dataset, therefore, finally had rows where each row was a 42-element vector. This vector encompassed the calculated metrics across different powers for the sentence embeddings. These enhanced sentence embeddings, with metrics covering higher orders, were then used to train the RoBERTa model. The goal was to produce scores that indicate how related different sentences are across various languages. The libraries used are in Table 5.

5 Experiments

5.1 Supervised Task

Static Approaches: The baseline models, Jaccard Coefficient, Dice Coefficient, and similar coefficients after removing stopwords were calculated to arrive at reliable baseline metrics to build upon.

Multilingual BERT: Since the best-performing model for English involved BERT, an attempt was made to train multilingual BERT: mBERT (Pires et al., 2019), by extending the vocabulary to allocate the tokens of various low-resource languages like Amharic, Hausa, Algerian Arabic, Afrikaans, Indonesian etc. The approach included generation of the vocabulary of each of the languages from the training data and then calling the pretrained mBERT model and tokenizer. The trained tokenizer was extended to include the new tokens generated from the vocab of the corresponding low resource languages. A trainable feed-forward network was added with the corresponding dropout. The loss metric used was mean squared error loss on both the training validation data and the Spearman rank correlation were calculated at the end of each validation epoch. Finally, finetuning multilingual BERT yielded considerably good results and this avenue was found suitable for exploration, especially for the cross-lingual task.

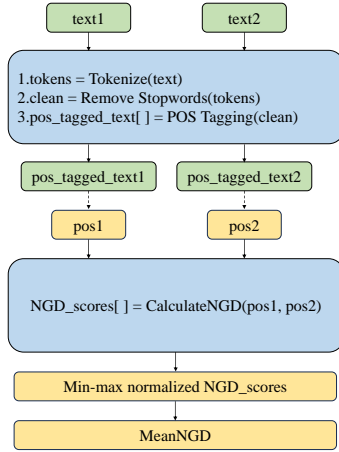The relatedness metric was approximated using a

Figure 2: NGD Calculation flowchart

Figure 3: Covariance Matrix between all 42 metrics

trainable feed forward layer by experimenting with the number of hidden layers and activations. It was observed that having 3 hidden layers resulted in fairly good relatedness scores between pairs of sentences in almost all the languages. It was observed that GeLU performed better than ReLU activation primarily due to its steep curve around 0 which helps to model complex functions better. The combination of learning rate and weight decay also resulted in a stable training curve, avoiding suboptimal loss convergence. Thus these specific hyperparameters were optimal for mBERT retraining in terms of resource constraints and performance. The corresponding hyperparameters of the best performing model is presented in App. Table 4.

Contrastive Learning: The details of the system are described in §4.1. Experiments were run on the number of epochs during training.

Combined Similarity Metric: Normalized Google Distance (Cilibrasi and Vitanyi, 2007) calculates a relatedness metric for two input sentences. It was proposed as a strong metric for relatedness by Lopes and Moura (2019). It starts by tokenizing and removing stop words from both sentences, followed by part-of-speech tagging. Then, it calculates NGD values for pairs of words with the same part of speech in both sentences. The NGD scores are normalized and averaged to compute the overall NGD score, representing the degree of relatedness between the two sentences. The flowchart for the process is shown in Figure 2. Cosine similarity is the standard metric used to find similarity between two sentence embeddings, which gives a 0.81-0.82 baseline score for this problem.However, similar or better results can be seen when other distance
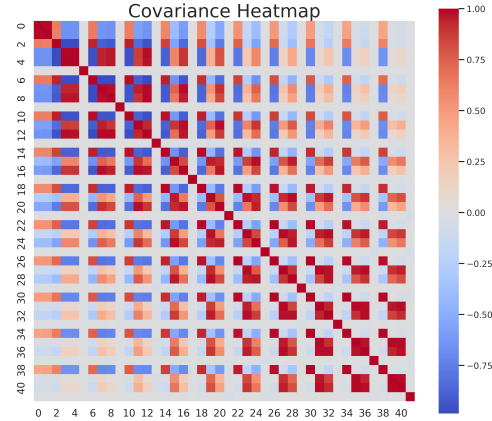
metrics like Mahalanobis Distance(0.82) and Euclidean Distance(0.83) are observed between the embeddings. Further, augmenting this with more direct relatedness metrics like NGD is promising for better results. A simple supervised deterministic regression model can be implemented to combine these metrics. Furthermore, to explore the importance of each of these metrics, a simple covariance matrix (Figure 3) can show how the vector metrics on higher element-wise-powered vectors hold information not caught directly at the lower powers of the vectors.

To implement this supervised regression model, a simple 3-layered feed-forward neural network (with neurons **[25]+[50]+[25]**) is trained with. The layers are chosen to construct a lightweight network. Each data feature **x** was composed as:

$$\mathbf{x_i} = \{S(v_{i,1}, v_{i,2}), S(v_{i,1}^2, v_{i,2}^2), \ldots,$$
$$S(v_{i,1}^{10}, v_{i,2}^{10}), J(v_{i,1}, v_{i,2}), D(v_{i,1}, v_{i,2})\}$$

$$\text{where} \quad v^i = (v_1^i, v_2^i, \ldots v_n^i)$$
$$S(a,b) = \{C(a,b), E(a,b), M_1(a,b), M_2(a,b)\}$$
$$C(a,b) = \text{Cosine Similarity between } a \text{ and } b$$
$$E(a,b) = \text{Euclidean Distance between } a \text{ and } b$$
$$M_1(a,b) = \text{Manhattan Distance between } a \text{ and } b$$
$$M_2(a,b) = \text{Mahalanobis Distance between } a \text{ and } b$$
$$J(a,b) = \text{Jaccard similarity between } a \text{ and } b$$
$$D(a,b) = \text{Dice similarity between } a \text{ and } b$$

Even though this metric did not show promise in English, this was helpful in some low-resource languages, and hence was part of our system design for some languages.

5.2 Unsupervised Task

Bigram Corpus Creation and Training Process: We developed a pipeline to generate a bigram dataset from any language corpus. A three-part tuple was created for every bigram found to note how
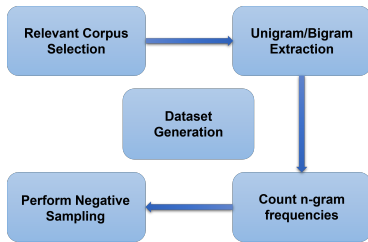
Figure 4: Bigram Corpus Creation Flowchart

often it appeared within the same sentence, paragraph, and entire document. This process aimed to quantify the connections between words by tracking their repeated sentence occurrences. Our main objective was to use these co-occurrence frequencies to produce word embeddings. We planned to enhance our method by applying hierarchical clustering, which helps identify word similarities and relationships. Moreover, we decided to use a 1:1 **negative sampling strategy** to refine the embeddings further. These embeddings were intended for computing relatedness scores, leveraging bigrams derived from sentence pairs and their lexical overlaps. Figure 4 presents a diagram illustrating this process.

TSDAE: A pipeline was developed to implement TSDAE (refer §4.2) on the languages. The number of epochs was changed and experimented on for various languages. Overall, around 20-25 epochs resulted in good results. No weight decay was implemented, and a learning rate of 3e-5 was used.

6 Results

The results for the supervised and unsupervised are mentioned in Table 2 and Table 3. The leaderboard highlights that the chosen contrastive learning approach did not perform well, especially for some languages where it fell significantly below the baseline scores provided by their system, such as Hausa, Moroccan Arabic, Telugu, etc. Some possible shortcomings of this approach might be that the negative samples were not distinguishable enough from the positive samples, which might also be attributed to the poor performance of the transformer models on languages, especially with complex lexical structures. The other issue might be the traditional loss function, which might not be good enough to capture the degree of semantic relationships between sentences. For the unsupervised track, our approach is performing reasonably well for most languages, which is indicated by the correlation score being more significant than the

Language	Rank	Score	Baseline Score
Amharic	17	0.55	0.85
Hausa	21	0.22	0.69
Kinyarwanda	21	0.14	0.72
Moroccan Arabic	22	0.36	0.77
Spanish	23	0.59	0.7
Algerian Arabic	23	0.34	0.6
Marathi	24	0.67	0.88
Telugu	25	0.28	0.82
English	31	0.81	0.83

Table 2: Evaluation Phase Results in Codalab Leaderboard for Track A

Language	Rank	Score	Baseline Score
Algerian Arabic	2	0.49	0.43
English	4	0.81	0.68
Amharic	6	0.07	0.72
Hausa	6	0.38	0.16
Moroccan Arabic	6	0.36	0.27
Spanish	9	0.59	0.69

Table 3: Evaluation Phase Results in Codalab Leaderboard for Track B

baseline provided.

7 Error Analysis

After the evaluation phase, we were provided with the labels for the evaluation data. On experimenting with the semantic relatedness scores for some languages, mainly **Hausa** and **Kinyarwanda**, we found out that our system was not performing well enough on these languages even after subsequent training and hyperparameter optimizations. The issue would be primarily attributed to the complex lexical variations and grammar rules of these languages. As for the unsupervised track, generating a bigram corpus for the case of **Amharic** seemed difficult due to its language structure.

8 Conclusion

By utilizing various approaches like contrastive learning, autoencoders, a custom relatedness metric incorporating all of the available lexical similarity metrics, we have developed a system capable of evaluating the degree of semantic relatedness between pairs of sentences in diverse high and low resource languages. In future, we will study the properties of each low resource language to find out where the models are performing poorly than relying too much on pre-trained models. This would give much clearer insights into semantics of each language thus improving the overall efficiency and performance of our system.

1447

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study.

Mansour Al Sulaiman, Abdullah M Moussa, Sherif Abdou, Hebah Elgibreen, Mohammed Faisal, and Mohsen Rashwan. 2022. Semantic textual similarity for modern standard and dialectal arabic using transfer learning. *Plos one*, 17(8):e0272991.

Rudi L. Cilibrasi and Paul M.B. Vitanyi. 2007. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383.

Samruddhi Deode, Janhavi Gadre, Aditi Kajale, Ananya Joshi, and Raviraj Joshi. 2023. L3cube-indicsbert: A simple approach for learning cross-lingual sentence representations using multilingual bert. *arXiv preprint arXiv:2304.11434*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. 2014. *Cohesion in english*. 9. Routledge.

Carla Teixeira Lopes and Diogo Moura. 2019. Normalized google distance in the identification and characterization of health queries.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert?

Nils Reimers and Iryna Gurevych. 2019. Making monolingual sentence embeddings multilingual using knowledge distillation. *CoRR*, abs/1908.10084.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex — a new corpus for lexical complexity prediction from Likert Scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.

Neil Shirude, Sagnik Mukherjee, Tushar Shandhilya, Ananta Mukherjee, and Ashutosh Modi. 2021. IITK@LCP at SemEval-2021 task 1: Classification for lexical complexity regression task. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 541–547, Online. Association for Computational Linguistics.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning.

A   Appendix

Hyperparameters	Values
Learning Rate	2e-5
Dropout	0.1
Weight Decay	0.01
Number of Linear Layers	3
Activation	GELU
Max Length	512

Table 4: Hyperparameters for mBERT retraining

Libraries	Version
numpy	1.25.2
PyTorch	2.0.1+cu117
transformers	4.36.2
sentence_transformers	2.2.2
scikit-learn	1.3.2
pandas	2.1.4

Table 5: Libraries used in our system