MMLabUIT at CoMeDi Shared Task: Text Embedding Techniques versus Generation-Based NLI for Median Judgment Classification

Le Duc Tai[1,2], Trong-Tai Dam Vu[1,2], Dang Van Thin[1,2],
[1]University of Information Technology-VNUHCM, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
23521374@gm.uit.edu.vn, {taidvt,thindv}@uit.edu.vn

Abstract

This paper presents our approach in the COLING 2025 - CoMeDi task in 7 languages, focusing on sub-task 1: Median Judgment Classification with Ordinal Word-in-Context Judgments (OGWiC). Specifically, we need to determine the meaning relation of one word in two different contexts and classify the input into 4 labels. To address sub-task 1, we implement and investigate various solutions, including (1) Stacking, Averaged Embedding techniques with a multilingual BERT-based model; and (2) utilizing a Natural Language Inference approach instead of a regular classification process. All the experiments were conducted on the P100 GPU from the Kaggle platform. To enhance the context of input, we perform Improve Known Data Rate and Text Expansion in some languages. For model focusing purposes Custom Token was used in the data processing pipeline. Our best official results on the test set are 0.515, 0.518, and 0.524 in terms of Krippendorff's α score on task 1. Our participation system achieved a Top 3 ranking in task 1. Besides the official result, our best approach also achieved 0.596 regarding Krippendorff's α score on Task 1.

1 Introduction

The CoMeDi 2025 shared-task (Schlechtweg et al., 2025) aims to investigate and model disagreements in word sense annotation within context. Specifically, the task focuses on predicting the median annotator judgment for word usage pairs based on an ordinal scale and exploring the linguistic and semantic factors that contribute to annotation disagreement. Two sub-tasks were proposed for participants in this shared task. The first challenge called Median Judgment Classification with Ordinal Word-in-Context Judgments, aims to measure the meaning of a word in two different contexts by classifying them into four ordinal judgments: "homonymy", "polysemy", "context variance", and "identity". While the second task, Mean Disagreement Ranking with Ordinal Word-in-Context Judgments aims to predict the mean of pairwise absolute judgment differences between annotators.

In general, the data annotation process is often hindered by disagreements among annotators and misunderstandings in daily communication. These challenges stem from the inherent ambiguity of language, where a single word can have multiple meanings and word meanings can shift based on context. Such ambiguity can significantly impact communication quality, leading to misinterpretations and reduced clarity. Addressing these issues is essential to improve the accuracy and reliability of both human and automated communication. As a result, in this paper, we present our solutions for Task 1 - Median Judgment Classification with Ordinal Word-in-Context Judgments in the CoMeDi 2025 shared-task (Schlechtweg et al., 2025). Specifically, we employ two different approaches to address this task: (1) stacking and average text embedding methods, and (2) BERT-based and generative-based models with natural language inference, combined with custom tokens.

2 Related Works

In recent years, researchers have made significant advancements in linguistic features such as Named Entity Recognition and part-of-speech tagging. However, there has been limited exploration of utilizing BERT-based models with Natural Language Processing approaches or custom tokens. An early SemEval shared task, Task 3, was introduced by (Armendariz et al., 2020), which had a substantial impact on advancing research in grading word similarity within context. This challenge is closely related to our CoMeDi task (Schlechtweg et al., 2025). A study by Hettiarachchi and Ranasinghe (2020) proposed an innovative method to enhance model performance using Stacked Embeddings. In this approach, different word embeddings are concatenated to create a final vector. By combining embeddings from various learning techniques, this method integrates their distinct characteristics. Additionally, average embeddings, which consider the mean of weights across different layers, are used to merge the information learned at each layer. Cosine similarity is then computed to generate predictions.

The work by Costella Pessutto et al. (2020) introduced a technique called BabelEncoding, which significantly improved word similarity grading in the context of Croatian. BabelEncoding involves three key steps: translation, multi-embedding extraction using BERT and Mono Word Embeddings, and the calculation of weighted averages. Chen et al. (2020) enhanced prediction results by incorporating sentence structure and TF-IDF (term frequency-inverse document frequency) features along with BERT word embeddings. In their approach, TF-IDF features were integrated into a masking layer of the BERT model, rather than just feeding the input text into BERT alone. Meanwhile, Gamallo (2020) proposed an innovative solution for word similarity tasks by combining BERT word embeddings with Dependency-Based Contextualization. This technique improves inference by considering the contextual meaning of a word in a sequence, taking into account the static embeddings of syntactically related words to the target word.

3 Task Description

The **CoMeDi (Contextual Meaning Disagreement)** shared-task[1] focuses on exploring and modeling disagreements in annotator judgments regarding word meanings in specific contexts. The primary goal is to understand and predict these disagreements in "Word-in-Context" (WiC) scenarios, where the meaning of a word can change based on its usage. There are two sub-tasks proposed to address as described below.

3.1 Task 1: Median Judgment Classification with Ordinal Word-in-Context Judgments (OGWiC)

In Task 1, the goal is to predict the median of annotator judgments for each word use pair in the WiC data. Each use pair consists of two instances of the same target word in different contexts. Annotators rate the relatedness of these instances on an ordinal scale from 1 to 4. This task can also be framed as a classification problem, where the objective is to categorize the relationship between the two instances into one of four classes: "homonymy", "polysemy", "context variance", and "identity".

3.2 Task 2: Mean Disagreement Ranking with Ordinal Word-in-Context Judgments (DisWiC)

In Task 2, the purpose task is to predict the mean of pairwise judgment differences between annotators for each use pair. This task involves ranking instances based on the level of disagreement observed in annotators' ratings. Unlike Task 1, which focuses on classification, Task 2 explicitly aims to capture and rank instances with higher annotator disagreement, providing insight into areas where word meanings are more subjective or ambiguous.

3.3 Dataset descriptions

The dataset provided by the competition includes seven languages (Chinese, English, German, Norwegian, Russian, Spanish, and Swedish), based on various data sets on semantic change as shown in Table 1. This multilingual scope provides a unique opportunity to explore how annotator disagreement patterns manifest across different linguistic and cultural contexts.

4 Methodology

In this section, we present three approaches for Task 1 in CoMeDi shared tasks in detail.

4.1 Data Processing

Our initial experiments focused on three dataset variations: raw, cleaned, and lemmatized. Specifically, we applied lemmatization and punctuation removal as part of the data cleaning process. However, these pre-processing steps did not lead to improved accuracy. Consequently, we simplified the cleaning process by removing only special characters, hashtags, and URLs.

Given the imbalanced nature of the dataset, we employed the stratified K-fold cross-validation technique (Bates et al., 2023) with $K = 10$ to mitigate the effects of data imbalance on the models. Stratified cross-validation ensures that the class distribution remains consistent across folds, thereby reducing bias in performance estimation caused by unequal class distributions in random splits. This approach enables a more reliable evaluation of model performance across diverse subsets of the data.

[1] https://comedinlp.github.io/

Table 1: Dataset Information for the Median Judgment Task.

Language	Dataset[version]
Chinese	ChiWUG[1.0.0] (Chen et al., 2023)
English	DWUG_EN [3.0.0], DWUG_EN_resampled [1.0.0] (Schlechtweg et al., 2024)
German	DWUG_DE [3.0.0], DWUG_DE_resampled [1.0.0], DiscoWUG [2.0.0], RefWUG [1.1.0] ((Schlechtweg et al., 2024) (Kurtyigit et al., 2021))
	DURel [3.0.0] (Schlechtweg et al., 2018)
	SURel [3.0.0] (Hätty et al., 2019)
Norwegian	NorDiaChange1, NorDiaChange2 (Kutuzov et al., 2022)
Russian	RuSemShift_1, RuSemShift_2 (Rodina and Kutuzov, 2020)
	RuShiftEval1, RuShiftEval2, RuShiftEval3 (Kutuzov and Pivovarova, 2021)
	RuDSI (Aksenova et al., 2022)
Spanish	DWUG_ES [4.0.1] (Schlechtweg et al., 2024)
Swedish	DWUG_SV [3.0.0], DWUG_SV_resampled [1.0.0] (Schlechtweg et al., 2024)

For data augmentation, we employed back-translation, applying it to entire sentences while preserving the target word. However, this method did not yield significant improvements, probably due to contextual alterations introduced during the translation process. Consequently, we opted not to use the back-translation technique to address the imbalance problem.

4.2 Stack Embedding

To create a final representation of each word-use pair, we combine BERT-based embeddings from different pre-trained language models, including mBERT$_{\text{large}}$ (Pires et al., 2019) and XLM-RoBERTa$_{\text{large}}$ (Conneau et al., 2019). These models are used to extract the embedding features of BERT words. Stacked embeddings are created by concatenating vectors from multiple embedding models to form a final, richer representation. This approach leverages the complementary characteristics of different embeddings, enabling the models to generalize across domains and adapt more effectively during fine-tuning. Let v_i^{stk} represent the final or stacked word vector corresponding to the word i, and v_{model_i} represent the vector obtained by using the embedding model m. The stacked vector is formed as shown below:

$$v_i^{\text{stk}} = \begin{bmatrix} v_{\text{model}_1,i} \\ v_{\text{model}_2,i} \\ \vdots \\ v_{\text{model}_m,i} \end{bmatrix} \tag{1}$$

After extracting the Stack Embedding features, we calculated Cosine Similarity and followed the baseline approach provided by the organizers. As shown in Table 2, the results on the test set demonstrated the performance of this approach.

Table 2: The result of Stacking Embedding method.

Model	Data	Krippendorff's α
BERT	Raw	0.267
BERT	Clean	**0.312**
XLM-Roberta	Raw	0.217
XLM-Roberta	Clean	0.201

4.3 Averaged Embedding

Instead of stacking the different representations, we also compute the average of the weights across different layers to combine the information learned by each layer. This approach is called an average embedding approach. For word i, the average embedding v_i^{avg} is calculated by considering the last l layers, as shown in Equation 2. The weights in the last layer are represented by the vector v_i^{-1}, and k denotes the number of layers selected for this calculation. The formula of the average embedding technique is presented below.

$$v_i^{\text{avg}} = \frac{v_i^{-l} + \cdots + v_i^{-1}}{k} = \frac{1}{k}\sum_{l=1}^{k} v_i^{(l)} \tag{2}$$

Because each layer returns a distinct embedding and different layers of transformer-based models often capture different types of information, the lower layers tend to capture more syntactic features, such as sentence structure and grammar, while higher layers capture more semantic information, such as word meaning and sentence context. Average Embedding provides a more robust representation of

Table 3: The result of Average Embedding method.

Model	Data	Krippendorff's α
BERT	Raw	0.193
BERT	Clean	**0.341**
XLM-Roberta	Raw	0.229
XLM-Roberta	Clean	0.231

a word by reducing the impact of noisy or outlier activations in individual layers. It also helps reduce the dimensionality of the feature space, creating a more compact representation of the word or sentence. By combining both syntactic and semantic features, Average Embedding can improve the quality of the input embeddings for model fine-tuning. After extracting the Average Embedding features, we computed the Cosine Similarity and followed the baseline approach provided by the organizers. The results of the test set are shown in Table 3.

4.4 Natural Language Inference

Natural Language Inference (NLI) is the task of determining whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) given a "premise". NLI can also be treated as a classification task, but there are some key differences between the two. NLI requires two text inputs, labeled as "hypothesis" and "premise", and the model needs to classify the relationship between them into one of three possible labels. Our team observed that NLI bears a strong resemblance to Task 1: Median Judgment Classification with Ordinal Word-in-Context Judgments, as both tasks involve classifying or grading the relationship between two textual elements. In task 1, the goal is to classify the similarity of two words, which is conceptually similar to determining the relationship between two sentences in NLI. Therefore, conducting experiments in Task 1 using NLI could prove promising.

Our team experimented with two strategies, including:

- **Fine-tuning the original language models on NLI task**: The list of language models includes mBERT [2] (Devlin et al., 2018), XLM-R[3] (Conneau et al., 2019) and XLM-R [4] (Liu et al., 2019).
- **Fine-tuning the language models trained NLI task**: The purpose of this task is to continue fine-tuning the model that is trained on the NLI task for the OGWiC task. We choose the XLM-R-XNLI model[5] as the main language model for this strategy.

Initially, our team conducted experiments on small models due to GPU resource limitations with the aim of testing whether our approach was effective. These initial experiments confirmed that BERT-based models performed better than the stacking and average embedding methods. Subsequently, we analyzed larger BERT-based models, such as *FacebookAI/xlm-roberta-large* and *FacebookAI/roberta-large*.

Even though large BERT-based classification approaches yielded better results than stacking and average embedding methods, as shown in table 4, the results demonstrated that the large BERT-based classification approach achieved Krippendorff's α scores of 0.381 and 0.419, surpassing the best scores of the stacking and average embedding methods, which were 0.312 and 0.341, respectively.

Additionally, our team examined the performance of a BERT-based model previously trained on the Natural Language Inference (NLI) task. As expected, the *joeddav/xlm-roberta-large-xnli* model significantly outperformed the other two large-sized models.

4.5 Generative-based Model Approach

In this approach, using a generative-based model, our team opted to experiment with the BART model (Lewis et al., 2020) by adapting it for a classification task through fine-tuning. BART functions as a denoising auto-encoder designed for pretraining sequence-to-sequence models. It is trained by intentionally introducing noise into text and then learning to reconstruct the original content.

The model employs a standard Transformer-based neural machine translation framework, which, while straightforward, effectively generalizes over other models such as BERT (with its bidirectional encoder) and GPT (with its left-to-right decoder), along with recent pretraining approaches. For fine-tuning BART for sequence classification tasks, the model processes the input through both the encoder and decoder. The last hidden state of the final token in the decoder is then fed into a new linear classifier for multi-class prediction. This

[2] google-bert/bert-base-multilingual-cased

[3] FacebookAI/xlm-roberta-large

[4] FacebookAI/roberta-large

[5] joeddav/xlm-roberta-large-xnli

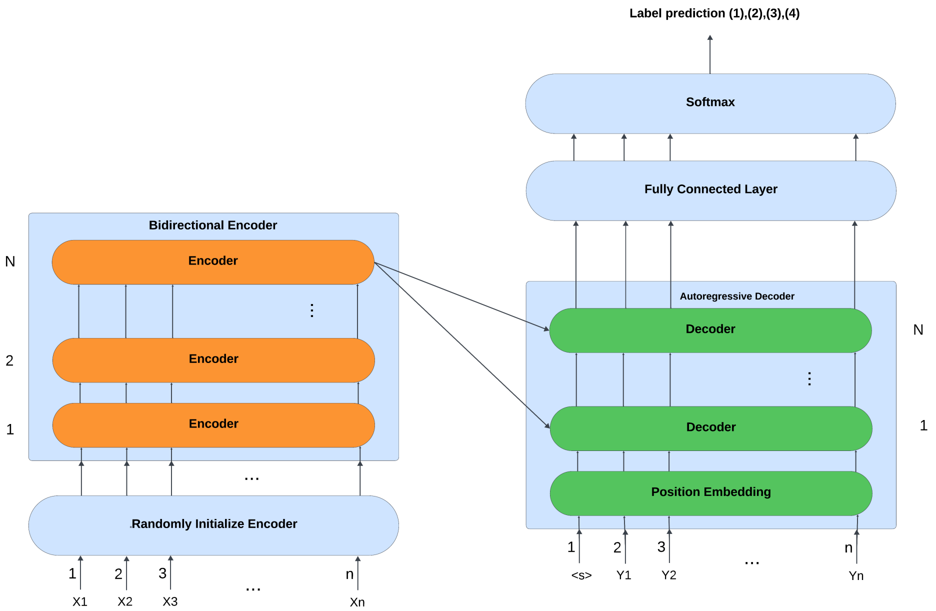

Figure 1: BART Architecture fine-tune for NLI task.

Table 4: The experimental results of BERT-based classification and NLI approach on the test set.

Model	Method	Krippendorff's α
facebook/bart-large-mnli	Natural Language Inference	**0.518**
joeddav/xlm-roberta-large-xnli	Natural Language Inference	0.482
FacebookAI/roberta-large	Classification	0.419
FacebookAI/xlm-roberta-large	Classification	0.381
google-bert/bert-base-multilingual-cased	Classification	0.356

approach resembles the use of the CLS token in BERT; however, an additional token is appended to the input's end, enabling the final token's representation in the decoder to attend to all decoder states generated from the full input sequence.

Similar to the BERT-based approach, we used a tokenizer to tokenize the two inputs, which were then fed into BART. Moreover, we utilized the pre-trained *facebook/bart-large-mnli* (Lewis et al., 2019) model, which was trained on the MNLI (Williams et al., 2018) dataset. The generative-based model achieved remarkable results compared to the BERT-based model, as shown in Table 4.

4.6 Custom Token

Given the promising results achieved by pre-trained BERT-based models on Natural Language Inference tasks, we sought to further explore this approach. While pre-trained Natural Language Inference models offer significant advantages, a key challenge arises in directing the model's focus to specific target words rather than entire sentences. To address this, our team introduced a Custom Token technique designed to enhance the model's attention to target words. Our analysis suggests that by incorporating Custom Tokens around target words, the model can allocate greater attention to these specific words, leading to subtle improvements in prediction accuracy. The following example illustrates the application of Custom Tokens:

Original input:

Context1: *"Esposito has gone for an afternoon walk and fallen asleep, his walking stick in his hand, one knee bent, his head pillowed on a stone."*

Context2: *"Old shopping lists and ticket stubs and wads of listed newsprint come falling around Pafko in the faded afternoon."*

Custom Token

Context1: *"Esposito has gone for an <target> afternoon </target> walk and fallen asleep, his walking stick in his hand, one knee bent, his head pillowed on a stone."*

Context2: *"Old shopping lists and ticket stubs and wads of listed newsprint come falling around Pafko in the faded <target> afternoon </target>."*

Custom tokens help clarify for the model which parts of the input are significant for the task. Thus, with the help of Custom Token, we combined this technique with the Natural Language Inference approach, and our team has recognized a slight improvement in accuracy, which is 0.524 in terms of Krippendorff's α.

4.7 Improve Known Data Rate

In this research, we used pre-trained embedding models, which meant that the dataset included tokens like names of people, organizations, locations, and other entities that weren't part of the model's original vocabulary. To create consistency and make the data more recognizable for the model during embedding generation, we replaced these unfamiliar names of people, organizations, locations, and other entities that weren't part of the model's original vocabulary with more common ones. This transformation was done automatically using the Named Entity Recognition (NER) task, based on the approach described by (Pakhale, 2023). Identified named entities, detected with spaCy (Honnibal and Montani, 2017) tools, were substituted in place of the unknown tokens. The example transformation is shown below:

Original: "**Esposito** *has gone for an afternoon walk and fallen asleep, his walking stick in his hand, one knee bent, his head pillowed on a stone."*

Improve Known Data Rate: "**Person** *has gone for an afternoon walk and fallen asleep, his walking stick in his hand, one knee bent, his head pillowed on a stone."*

As you can see in the example transformation, *"Esposito"* is replaced with *"Person"*. However, due to the limitation of time and resources our team could only perform Improve Known Data Rate transformation in English and Swedish.

5 Experimental Setup

5.1 Data and Evaluation Metrics

We conducted experiments exclusively on the dataset provided by the organizer for training models and testing approaches in this shared task. Table 5 summarizes key information about the training and testing datasets, while Table 6 provides general statistics and the distribution of four classes in the training dataset. By observing the class polarity in Table 6, we note that the ratio between the classes is unbalanced. Specifically, the total samples in classes (1), (2), and (3) are fewer than the total samples in class (4). This imbalance could introduce bias during fine-tuning.

Table 5: The information of the experimental dataset.

Information	Training set	Validation set	Test set
Number of samples	47833	8287	15332
Number of tokens	2990377	436735	985402
The average length	40.41	37.94	40.74
The maximum length	1643	493	605

Imbalanced data was one of the main challenges that competitors needed to address while implementing distinct techniques to achieve optimal results. To handle the imbalance in class labels, our team utilized data augmentation techniques, one of the most effective methods for addressing this issue. Data augmentation helps mitigate bias in performance estimation. Specifically, we applied the back-translation method to classes (1), (2), and (3) to reduce data polarity and make the class distribution less imbalanced.

However, the back-translation method proved suboptimal for addressing the imbalance issue. When translating input while preserving the target word, changes in the sentence's context may negatively impact the prediction of the target word. As shown in Table 5, the number of samples in the training dataset is significantly higher than in the testing dataset, enabling our models to train effectively and generalize well. Additionally, we perform some data cleaning processes before fine-tuning models:

- **Noise Removal:** We observed that there are a lot of noises, such as punctuation and special characters, in the dataset. We found that these noises are not necessary for the sentence-level dataset. Therefore, we remove it from the samples.

- **Text Expansion:** we also perform text expansion in English for example: "*I'll*" into "*I will*" or "*he'd*" into "*he would*". Text expansion was utilized for consistency of data purposes, and this can help the model to generalize better.

Table 6: The statistic of class distribution beyond dataset.

Class samples	Homonymy	Polysemy	Context variance	Identity
Training set	7099	4 510	5967	30257
Validation set	1055	817	739	5676
Full dataset	8154	5327	6706	35933

5.2 System Settings

We conducted our training process using HuggingFace (Wolf et al., 2020), and all BERT-based models were trained for 10 epochs. The AdamW optimizer was utilized to optimize the models. We selected a learning rate of 5e-5,3e-5 for BERT-based models. The batch sizes were set to 16 and 32, the random seed was set to 221, and the maximum token length was 512.

Due to computational resource limitations, we had to adjust system settings for fine-tuning the BART-MNLI model (Lewis et al., 2019). Specifically, we reduced the batch size to 8 and employed gradient accumulation to effectively train on larger effective batch sizes. This technique allows us to accumulate gradients over multiple smaller batches before updating the optimizer, mitigating memory constraints. Furthermore, we utilized mixed precision training (FP16) and gradient checkpointing to accelerate training and reduce memory usage. Mixed precision training combines 16-bit and 32-bit floating-point operations, enabling efficient training of large-scale models like transformers. Dynamic loss scaling was employed to maintain numerical stability. Given GPU limitations, we trained BART for only 6 epochs and opted for the AdaFactor optimizer, known for its efficiency in training large models, instead of AdamW. All models were evaluated using the metric provided by the task organizers. Our team leveraged a P100 GPU, available for up to 30 free hours per week on Kaggle, for computational resources.

6 Main Result

The official evaluation phase and post-evaluation phase submission results are presented in Table 7. The *facebook/bart-large-mnli* model with NLI, custom token, and average embedding on Chinese achieved the highest Krippendorff's α score of 0.596. In the official evaluation phase, we submitted predictions created with *joeddav/xlm-roberta-large-xnli* with Improve Known Data Rate and Custom Token for NLI, and *facebook/bart-large-mnli* fine-tuned for NLI, which attained a Krippendorff's alpha score of 0.524 and 0.518, respectively. Furthermore, in the last submission we submitted *joeddav/xlm-roberta-large-xnli* combined with Improve Known Data Rate which only achieved 0.515 in Krippendorff's alpha.

Through experimentation, our team observed that all classification or natural language inference approaches performed worse in Chinese compared to the stacking and average embedding methods. As a result, we utilized stacking and average embeddings exclusively for Chinese and found that average embedding outperformed stacking embedding in this context.

By combining different techniques, we leveraged the advantages of each method, leading to better results overall. Additionally, our team's official ranking in the top 3rd position demonstrates promising results in Task 1: Median Judgment Classification with Ordinal Word-in-Context Judgments (OGWiC).

7 Conclusion and Future Work

In this paper, we present our approaches for the shared task CoMeDi 2025 (Schlechtweg et al., 2025), Task 1: Median Judgment Classification with Ordinal Word-in-Context Judgments (OGWiC). Our methods achieved a top 3rd ranking in the official hard-label evaluation of Task 1 shown in Table 8 and achieved the final result by using *joeddav/xlm-roberta-large-xnli* combined with Custom token and Improve Know Data Rate technique which results in 0.524 final scores. Moreover, pretrained BART models on NLI task also achieve 0.518 and *joeddav/xlm-roberta-large-xnli* combined with Improve Know Data Rate only achieve 0.515 in Krippendorff's α.

We introduced various methods and combinations, including stacking, averaged embedding techniques, natural language inference, a generative-based model approach combined with custom tokens, and improved known data rates. Through experimentation and analysis, our approaches yielded promising results for Task 1. Moreover, our approaches can bring novelty in examining how word

Table 7: All evaluation and post-evaluation results.

Model	Method	Score
facebook/bart-large-mnli	NLI + Custom Token + Average Embedding(Chinese)	**0.596**
joeddav/xlm-roberta-large-xnli	NLI + Custom Token + Improve Known Data Rate	**0.524**
facebook/bart-large-mnli	Natural Language Inference	**0.518**
joeddav/xlm-roberta-large-xnli	NLI + Improve Known Data Rate	0.515
joeddav/xlm-roberta-large-xnli	Natural Language Inference	0.482
Baseline		0.123

meaning changes based on different contexts because the former research only uses the text embedding method for this task while our team's main approach is leveraging the power of not only BERT-based models but also generative-based models. We believe these methods apply to real-world tasks due to their low computational cost compared to large language model-based approaches.

Additionally, by analyzing the results, we observed that preprocess stages like data cleaning and data augmentation can improve the clarity and consistency of data representation which can further enhance performance.

Ranking	Team	score
Top 1	Deep-Change	0.656
Top 2	GRASP	0.583
Top 4	JuniperLiu	0.271
Baseline	-	0.123
Ours (Top 3)	**MMLabUIT**	**0.524**

Table 8: Official Results for Task 1: Median Judgment Classification with Ordinal Word-in-Context Judgments

Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

Anna Aksenova, Ekaterina Gavrishina, Elisei Rykov, and Andrey Kutuzov. 2022. RuDSI: Graph-based word sense induction dataset for Russian. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 77–88, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Carlos Santos Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Ivan Vulić, and Mohammad Taher Pilehvar. 2020. SemEval-2020 task 3: Graded word similarity in context. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 36–49, Barcelona (online). International Committee for Computational Linguistics.

Stephen Bates, Trevor Hastie, and Robert Tibshirani. 2023. Cross-validation: what does it estimate and how well does it do it? *Journal of the American Statistical Association*, pages 1–12.

Jing Chen, Emmanuele Chersoni, Dominik Schlechtweg, Jelena Prokic, and Chu-Ren Huang. 2023. ChiWUG: A graph-based evaluation dataset for Chinese lexical semantic change detection. In *Proceedings of the 4th International Workshop on Computational Approaches to Historical Language Change*, Singapore. Association for Computational Linguistics.

Weilong Chen, Xin Yuan, Sai Zhang, Jiehui Wu, Yanru Zhang, and Yan Wang. 2020. Ferryman at SemEval-2020 task 3: Bert with TFIDF-weighting for predicting the effect of context in word similarity. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 281–285, Barcelona (online). International Committee for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Lucas Rafael Costella Pessutto, Tiago de Melo, Viviane P. Moreira, and Altigran da Silva. 2020. BabelEnconding at SemEval-2020 task 3: Contextual similarity as a combination of multilingualism and language models. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 59–66, Barcelona (online). International Committee for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Pablo Gamallo. 2020. CitiusNLP at SemEval-2020 task 3: Comparing two approaches for word vector contextualization. In *Proceedings of the Four-*

teenth Workshop on Semantic Evaluation, pages 275–280, Barcelona (online). International Committee for Computational Linguistics.

Anna Hätty, Dominik Schlechtweg, and Sabine Schulte im Walde. 2019. SURel: A gold standard for incorporating meaning shifts into term extraction. In *Proceedings of the 8th Joint Conference on Lexical and Computational Semantics*, pages 1–8, Minneapolis, MN, USA.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2020. BRUMS at SemEval-2020 task 3: Contextualised embeddings for predicting the (graded) effect of context in word similarity. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 142–149, Barcelona (online). International Committee for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spacy: Industrial-strength natural language processing in python. Explosion AI. Available at https://spacy.io.

Sinan Kurtyigit, Maike Park, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Lexical Semantic Change Discovery. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online. Association for Computational Linguistics.

Andrey Kutuzov and Lidia Pivovarova. 2021. Rushifteval: a shared task on semantic shift detection for russian. *Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Dialog conference*.

Andrey Kutuzov, Samia Touileb, Petter Mæhlum, Tita Enstad, and Alexandra Wittemann. 2022. NorDiaChange: Diachronic semantic change dataset for Norwegian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2563–2572, Marseille, France. European Language Resources Association.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Kalyani Pakhale. 2023. Comprehensive overview of named entity recognition: Models, domain-specific applications and challenges. *arXiv preprint arXiv:2309.14084*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Julia Rodina and Andrey Kutuzov. 2020. RuSemShift: a dataset of historical lexical semantic change in Russian. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1037–1047, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dominik Schlechtweg, Pierluigi Cassotti, Bill Noble, David Alfter, Sabine Schulte im Walde, and Nina Tahmasebi. 2024. More DWUGs: Extending and evaluating word usage graph datasets in multiple languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida. Association for Computational Linguistics.

Dominik Schlechtweg, Tejaswi Choppa, Wei Zhao, and Michael Roth. 2025. The CoMeDi shared task: Median judgment classification & mean disagreement ranking with ordinal word-in-context judgments. In *Proceedings of the 1st Workshop on Context and Meaning–Navigating Disagreements in NLP Annotations*, Abu Dhabi, UAE.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic Usage Relatedness (DURel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, New Orleans, Louisiana.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.