Contextual Embeddings (ELMo) for Indian Languages

Final Report

Team No. 4 : Thunderbolts

Members:

Gaurang Patil – 2023701002

Ishank Kapania – 2023701012

Anshul Sharma – 2023701011

Problem Statement

One of the limitations of Word2Vec and GloVe embeddings is that they provide a single representation for each word, irrespective of the context in which they appear in. Polysemous words need to be handled to provide better word representation. Contextual embeddings were developed to provide multiple representations for a single word in case the word is polysemous. ElMo embeddings provide contextual representations and are capable of handling polysemy. The project is aimed at development of contextual embeddings for Hindi language and evaluation of the quality of word embeddings obtained by examining the performance of models on downstream NLP tasks (extrinsic evaluation), and to propose a metric for evaluation of contextual word embeddings in a semi-intrinsic setting.

Introduction & Literature Review

Evaluation of word embeddings can be accomplished at 2 levels- intrinsic and extrinsic. [2]

- Intrinsic evaluation:
    - Word similarity [2] : this involves manually creating a dataset comprising of word pairs and their respective similarity as perceived by a human on a rating scale (e.g. 0 to 10, where 10 stands for words that are very similar and 0 for those words that are dissimilar). Dot product between the word embeddings can be computed and its correlation with human assigned scores can be measured to determine quality of word embeddings. Higher correlation with human ratings indicates better word model.
    - Word analogy [2] : Involves finding b' such that a : a' :: b : b' where there exists an analogical relationship between a and a'.

      E.g. king : queen :: prince : princess

      Word embeddings capable of capturing such analogical relationships are considered to be of good quality.

- Extrinsic evaluation:
    - Performance on a downstream NLP task [2] : the performance of a word embedding model on a downstream NLP task like POS tagging, NER or sentiment analysis can be used to determine the quality of the word model. A good word model typically performs better on downstream tasks.
    - Search and retrieval performance [1]: the performance of embeddings on search related tasks and other information retrieval related tasks can depict the quality of embeddings produced. For example, query expansion may require word embeddings to determine additional words that can be included in the query to enhance retrieval performance. A good word model may thus enhance retrieval performance.

Deep Contextualized Word Representations (ElMo) [3]

- Given a sequence of N tokens, (t1, t2, ..., tN), a forward language model computes the probability of the sequence by modeling the probability of token tk given the history (t1, ..., tk−1). At each position k, each LSTM layer outputs a context-dependent representation $\overrightarrow{h}_{k,j}^{L,M}$ where j = 1, . . . , L. The top layer LSTM output, $h_{k,j}^{L,M}$, is used to predict the next token t_{k+1} with a Softmax layer. A

backward LM is similar to a forward LM, except it runs over the sequence in reverse, predicting the previous token given the future context. A biLM combines both a forward and backward LM. For each token t_k, a L-layer biLM computes a

$$\begin{aligned} R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\} \\ &= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\}, \end{aligned}$$

set of 2L + 1 representations.

Where $h_{k,0}^{L,M}$ is token layer and $h_{k,j}^{L,M}$ [$\overrightarrow{h}_{k,j}^{L,M,}$; $\overleftarrow{h}_{k,j}^{L,M}$] for each BiLSTM layer. For a downstream task ElMo collapses all layers into a single vector or more generally a task specific weighting is computed of all layers in a biLM

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}.$$

s^{task} are softmax-normalized weights and $\ddot{y}^{task}$ allows the task model to scale ElMo vector. $\ddot{y}$ acts as some sort of optimizer. [3]

- In a 2 layer LSTM Elmo Model the first LSTM layer often captures the synatactic dependecies and erstwhile the top layer often captures the semantic contextualised representations . [3]
- To add ELMo to the supervised model, we first freeze the weights of the biLM and then concatenate the ELMo vector $ELMo_k^{task}$ with x_k and pass the ELMo enhanced representation [x_k; $ELMo_k^{task}$] into the task biLSTM. [3]

Word Level Adversarial Attacks

- Authors attribute vulnerability to word level adversarial attacks to the fact that similar words get mapped to dissimilar representations [5]

Dataset Details

Corpus for training the language models on next word prediction: IndicNLP corpora

https://indicnlp.ai4bharat.org/pages/indicnlp-corpus/

Total sentences sampled for training LMs: 1,00,000

Datasets for semi-intrinsic evaluation:

We have used a self-created dataset using 100 samples from Product Reviews Dataset from IndicGLUE.

Datasets for extrinsic evaluation:

IndicGlue [4]

Link - https://ai4bharat.iitm.ac.in/indicglue/

The GLUE dataset, which stands for the General Language Understanding Evaluation benchmark, is a collection of diverse natural language understanding tasks. Its purpose is to evaluate and compare the performance of models on a wide range of linguistic tasks. IndicGlue is the natural language dataset adapted for indic languages.

1. **News Category Classification**
   Predict the genre of a given news article. We use BBC news classification dataset.
2. **Product Review Sentiment classification**
   Predict the sentiment of a given product review.
3. **Movie Review Sentiment Classification**
   Predict the sentiment of a given movie review.
4. **Midas Discourse Classification**
   Predict the type of discourse for a given snippet of text sourced from a story.
5. **Wikipedia Section Title Prediction**
   Predict the correct title for a Wikipedia section from a given list of four candidate titles.
6. **Cloze-style Question Answering**
   Given a text with an entity randomly masked, the task is to predict that masked entity from a list of 4 candidate entities.

7. **Named Entity Recognition**
   Recognize entities and their coarse types in a sequence of words. HiNer dataset is to be used.

8. **Natural Language Inference**
   This task involves determining whether a hypothesis is true (entailment), false (contradiction), or undetermined (neutral) given a premise. IndicXNLI dataset is to be used.

Summary of Implementation

Training LMs, Extrinsic and semi-intrinsic evaluation:

- Training of the forward and backward language models
    - Train on a generic corpus using 1 lakh sentences on next word prediction and previous word prediction.
    - Forward LM is used to provide left-context and backward LM is used to provide right-context.
- Creation of task specific ELMo embeddings for each of the tasks
- Train biLSTM for the downstream tasks
- Evaluate the performance of the model on the downstream tasks
- Development of a new dataset for semi-intrinsic evaluation of word embeddings
- Evaluate the word embeddings using the proposed metric
- Obtain embeddings with frozen lambdas and evaluate their quality using the proposed metric.
- For the MCQ's based dataset we concatenated the Question and the respective Questions to one fixed tensor that was fed to a downstream LSTM as suggested in the IndicGLUE paper. While then the classification was performed on 4 classes meaning 4 options for the MCQs

Proposed metric:

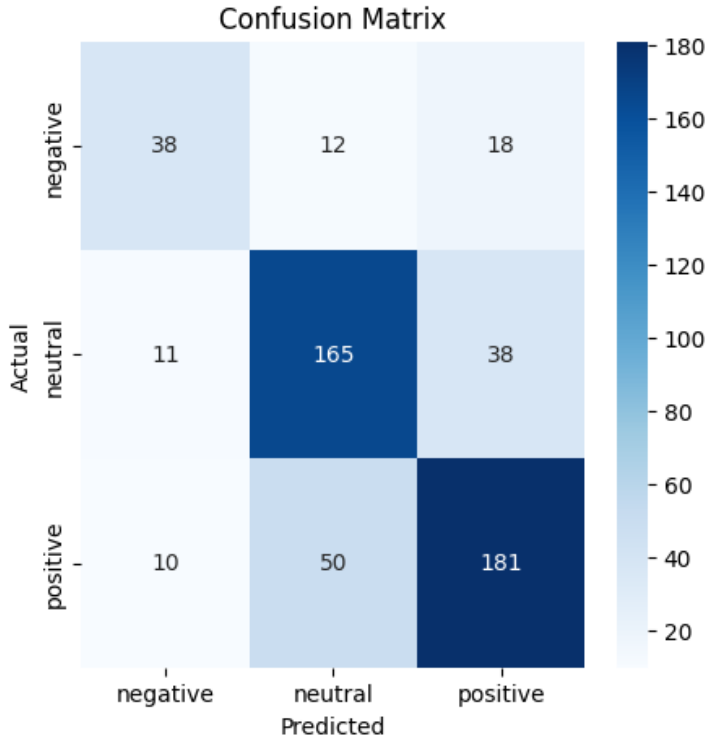
A semi-intrinsic approach to evaluate robustness of word embeddings

1. Create a set of 100 original sentences
2. For each sentence, substitute an "important" word with an appropriate synonym. Make grammatical changes if necessary and obtain the new sentence.
3. Let the classifier obtain y_hat1 and y_hat2 which are the predicted probabilities for the 2 sentences for the target class.
4. Compute | y_hat1 – y_hat2 | for each pair and store it in a list.
5. Median of the list gives the final value of the metric.

Results

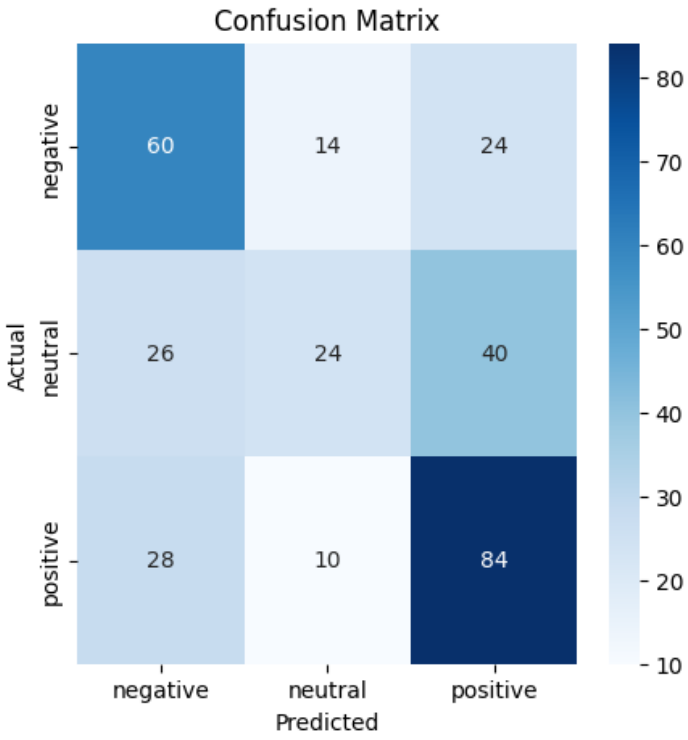
Product Reviews:

	precision	recall	f1-score	support
negative	0.64	0.56	0.60	68
neutral	0.73	0.77	0.75	214
positive	0.76	0.75	0.76	241
accuracy			0.73	523
macro avg	0.71	0.69	0.70	523
weighted avg	0.73	0.73	0.73	523



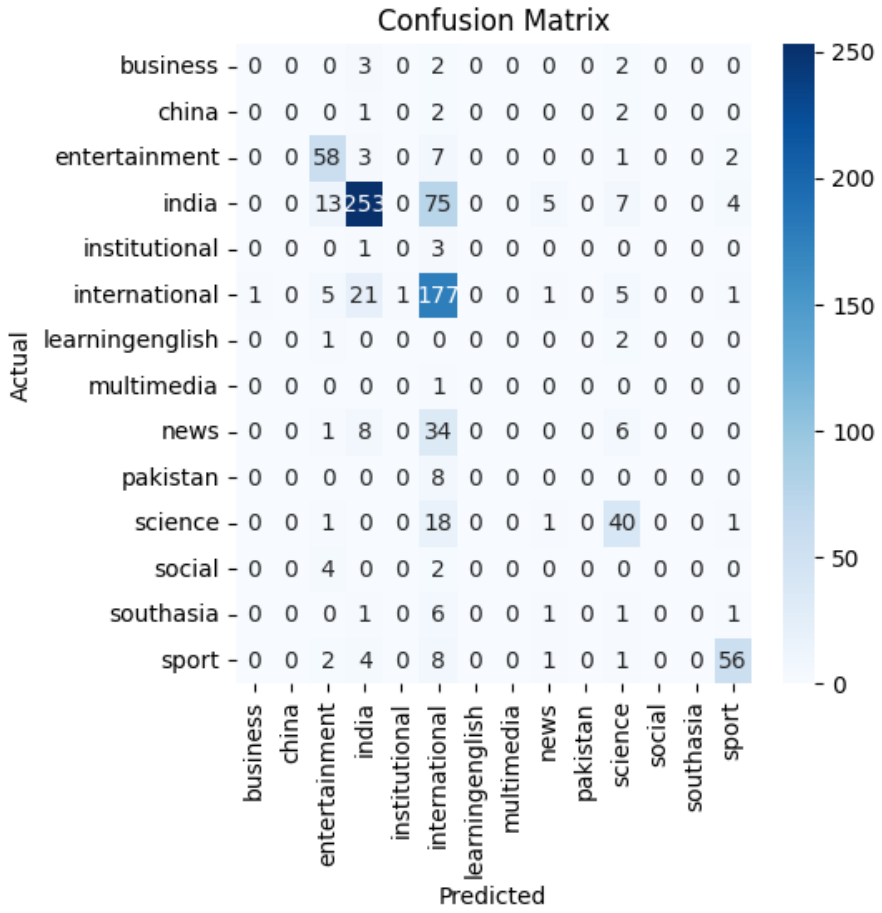
Movie Reviews:

	precision	recall	f1-score	support
negative	0.53	0.61	0.57	98
neutral	0.50	0.27	0.35	90
positive	0.57	0.69	0.62	122
accuracy			0.54	310
macro avg	0.53	0.52	0.51	310
weighted avg	0.53	0.54	0.52	310



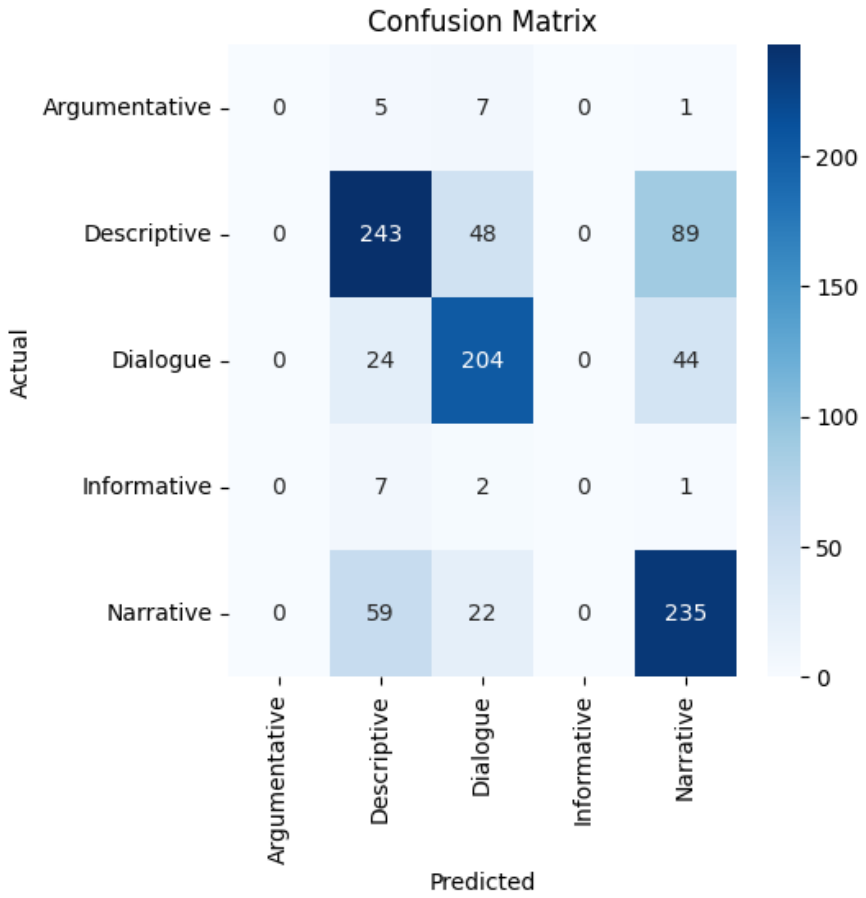
BBC news category classification:

	precision	recall	f1-score	support
business	0.00	0.00	0.00	7
china	0.00	0.00	0.00	5
entertainment	0.68	0.82	0.74	71
india	0.86	0.71	0.78	357
institutional	0.00	0.00	0.00	4
international	0.52	0.83	0.64	212
learningenglish	0.00	0.00	0.00	3
multimedia	0.00	0.00	0.00	1
news	0.00	0.00	0.00	49
pakistan	0.00	0.00	0.00	8
science	0.60	0.66	0.62	61
social	0.00	0.00	0.00	6
southasia	0.00	0.00	0.00	10
sport	0.86	0.78	0.82	72
accuracy			0.67	866
macro avg	0.25	0.27	0.26	866
weighted avg	0.65	0.67	0.65	866



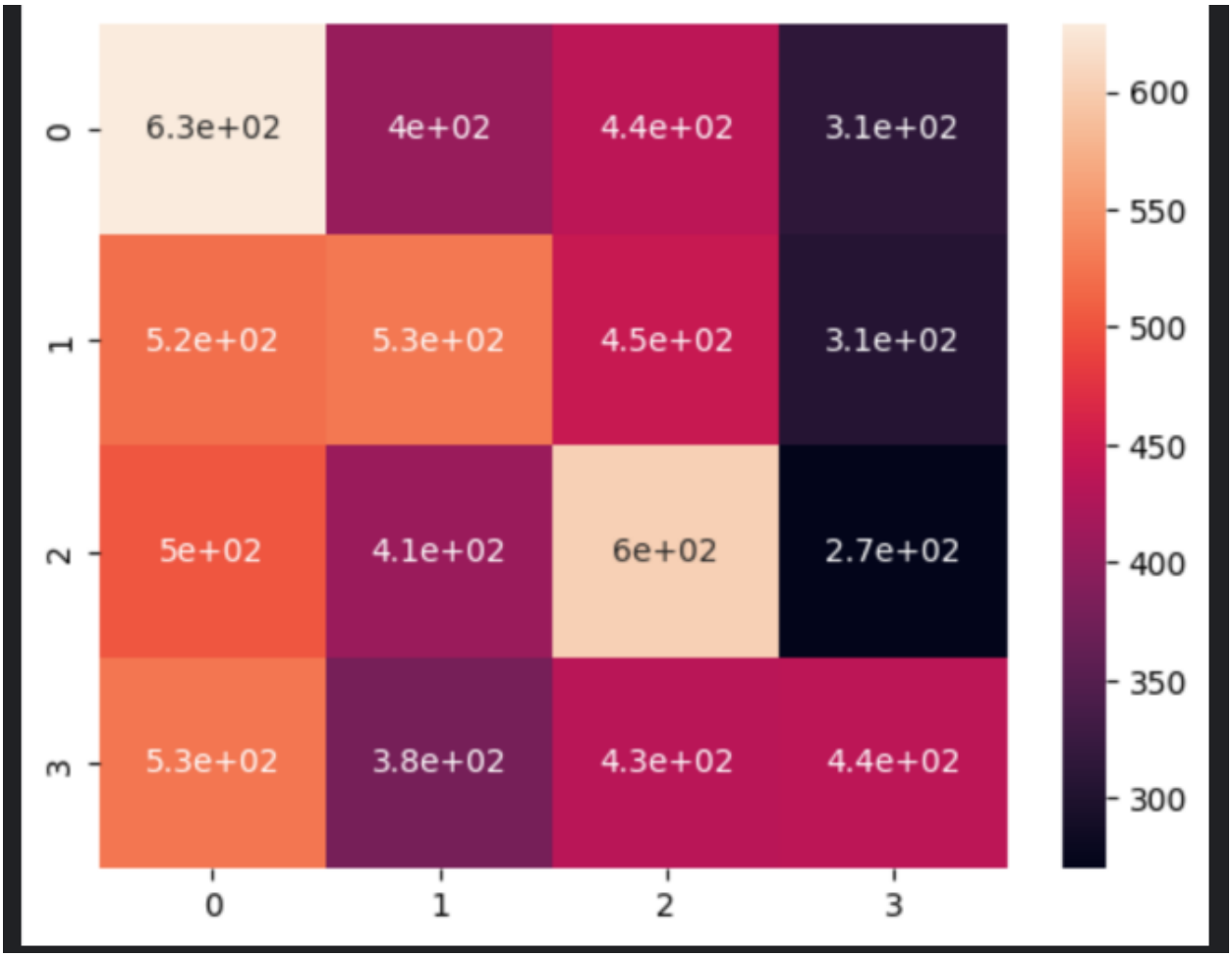
MIDAS discourse classification:

	precision	recall	f1-score	support
Argumentative	0.00	0.00	0.00	13
Descriptive	0.72	0.64	0.68	380
Dialogue	0.72	0.75	0.74	272
Informative	0.00	0.00	0.00	10
Narrative	0.64	0.74	0.69	316
accuracy			0.69	991
macro avg	0.41	0.43	0.42	991
weighted avg	0.68	0.69	0.68	991


Confusion Matrix
Argumentative
Descriptive
Dialogue
Informative
Narrative
Actual
0 5 7 0 1
0 243 48 0 89
0 24 204 0 44
0 7 2 0 1
0 59 22 0 235
200
150
100
50
0
Argumentative
Descriptive
Dialogue
Informative
Narrative
Predicted

Cloze-style-Multiple-choiceQA:

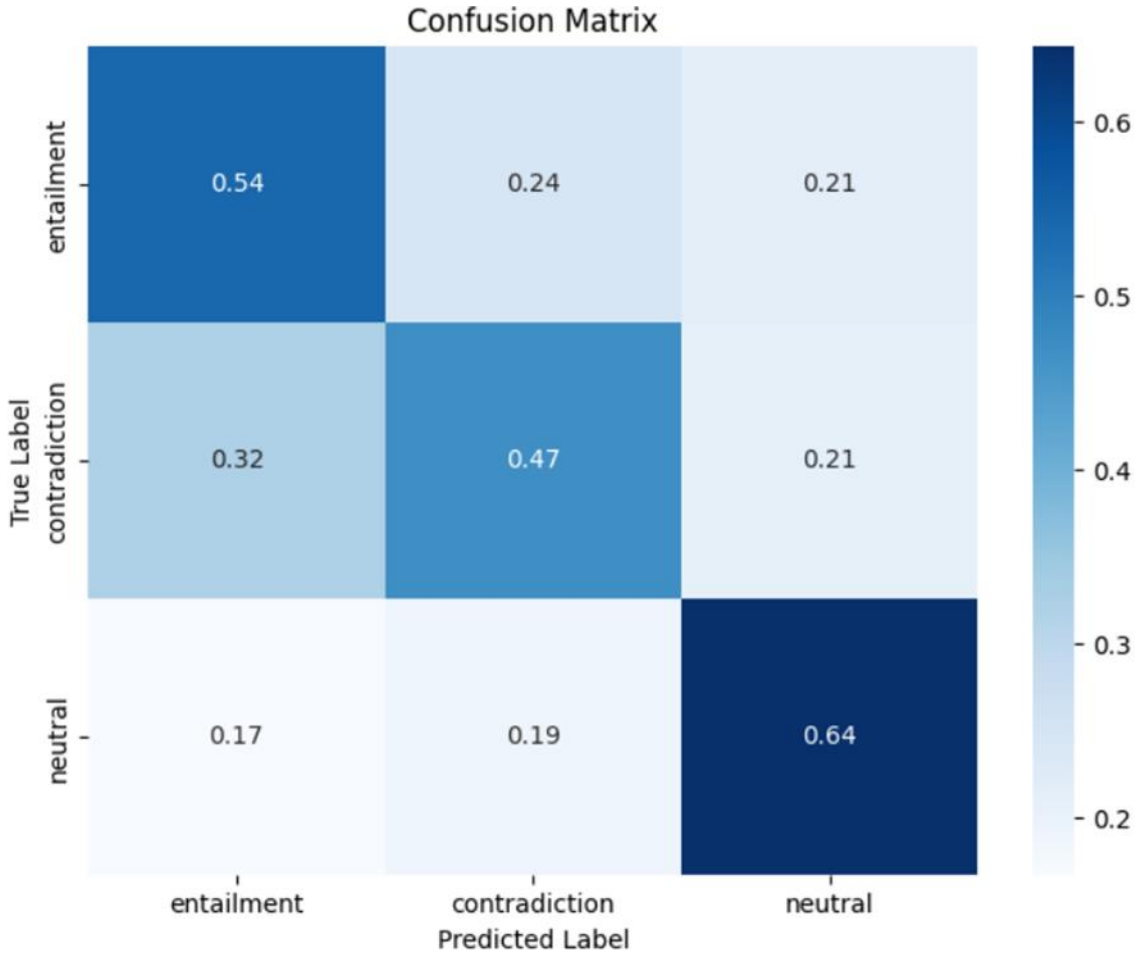
	precision	recall	f1-score	support
0	0.30	0.45	0.36	1761
1	0.38	0.17	0.24	1795
2	0.33	0.37	0.35	1826
3	0.34	0.33	0.33	1758
accuracy			0.33	7140
macro avg	0.34	0.33	0.32	7140
weighted avg	0.34	0.33	0.32	7140



Natural language inference:

Classification Report:

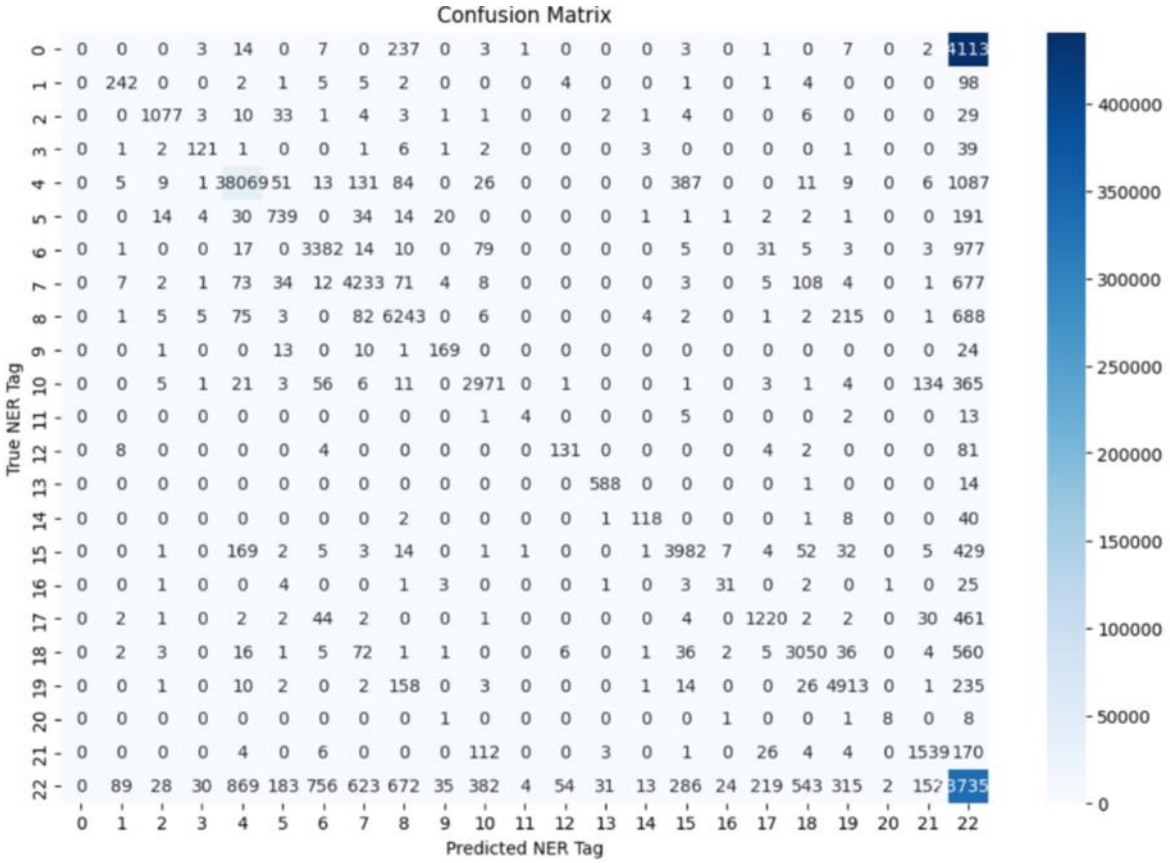
	precision	recall	f1-score	support
entailment	0.53	0.54	0.53	1670
contradiction	0.52	0.47	0.49	1670
neutral	0.60	0.64	0.62	1670
accuracy			0.55	5010
macro avg	0.55	0.55	0.55	5010
weighted avg	0.55	0.55	0.55	5010



Named entity recognition:

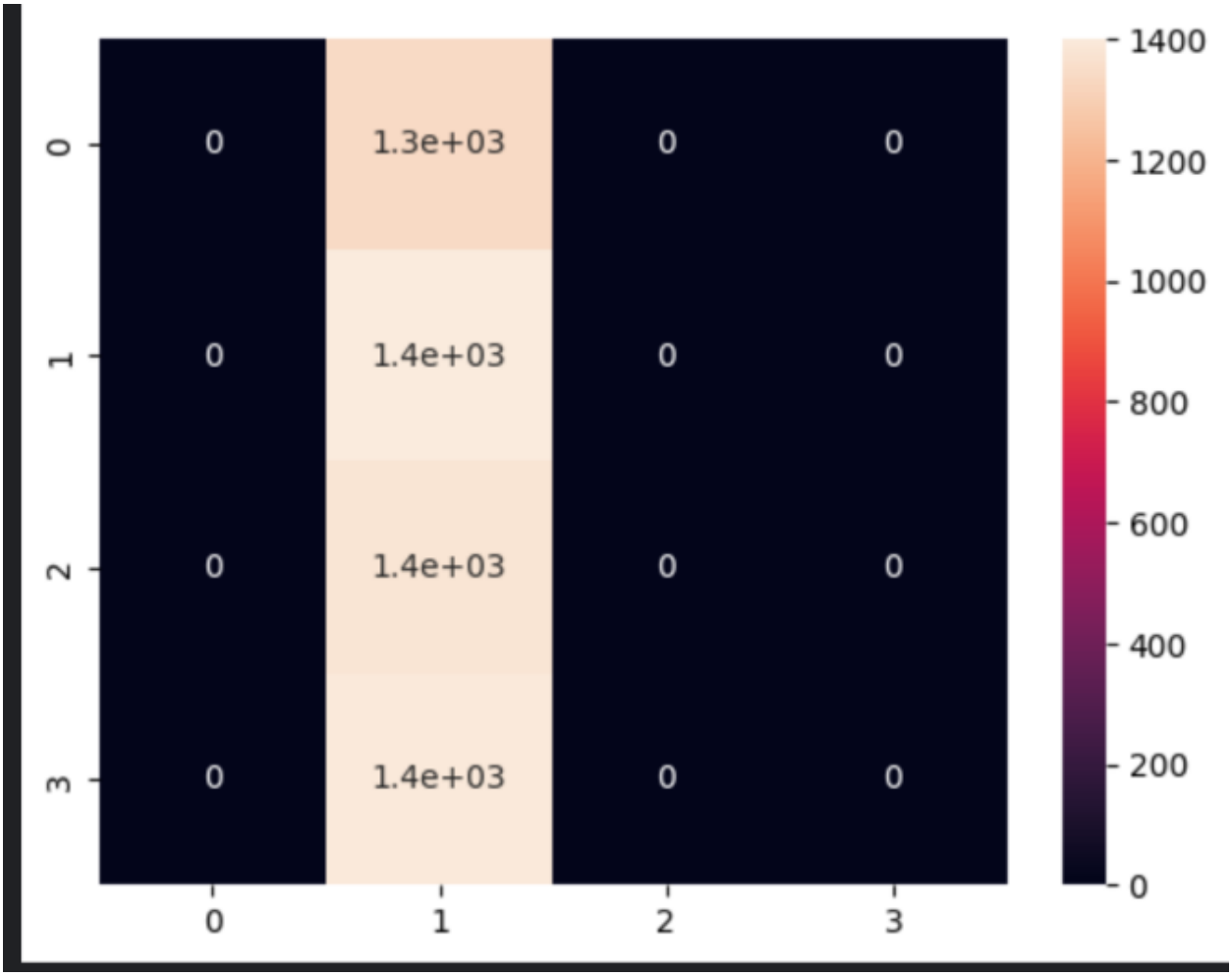
Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	441412
1	0.68	0.66	0.67	365
2	0.94	0.92	0.93	1175
3	0.72	0.68	0.70	178
4	0.97	0.95	0.96	39889
5	0.69	0.70	0.70	1054
6	0.79	0.75	0.77	4527
7	0.81	0.81	0.81	5243
8	0.83	0.85	0.84	7333
9	0.72	0.78	0.75	218
10	0.83	0.83	0.83	3583
11	0.40	0.16	0.23	25
12	0.67	0.57	0.62	230
13	0.94	0.98	0.96	603
14	0.83	0.69	0.75	170
15	0.84	0.85	0.84	4708
16	0.47	0.43	0.45	72
17	0.80	0.69	0.74	1773
18	0.80	0.80	0.80	3801
19	0.88	0.92	0.90	5366
20	0.73	0.42	0.53	19
21	0.82	0.82	0.82	1869
...				
accuracy			0.47	866280
macro avg	0.72	0.71	0.70	866280
weighted avg	0.25	0.47	0.32	866280


Confusion Matrix
True NER Tag
Predicted NER Tag
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
400000
350000
300000
250000
200000
150000
100000
50000
0

Wikipedia Section Title Prediction:

	precision	recall	f1-score	support
0	0.24	1.00	0.39	1338
1	0.00	0.00	0.00	1402
2	0.00	0.00	0.00	1376
3	0.00	0.00	0.00	1393
accuracy			0.24	5509
macro avg	0.06	0.25	0.10	5509
weighted avg	0.06	0.24	0.09	5509



Setting	Value of the proposed metric (Mean)	Value of the proposed metric (Median)
Trainable Lambdas	0.0820	0.0178
Frozen Lambdas	0.0907	0.0267
Original Embeddings	0.0621	0.0229

Analysis of Results

Comparison with IndicGLUE paper

Task	Our Test Accuracy	Highest Test Accuracy mentioned in IndicGLUE paper	Comments
Product Review Classification	73%	78.97 % (XLM-R)	Considering that we use ELMo embeddings and use a biLSTM for the classification tasks, the models perform slightly worse than XLM-R. We primarily attribute this to the lack of attention mechanism in biLSTM.
Movie Review Classification	54 %	61.61 % (XLM-R)	
BBC news classification	67 %	75.52 % (XLM-R)	
MIDAS discourse mode classification	69 %	79.94 % (XLM-R)	
Cloze-style Multiple-choice QA	33%	41.55 % (IndicBERT base)	The reason for this performance on this dataset is because of no attention mechanism in the downstream task. Also, MCQs focus on fixed keywords while LSTM's might get fixated on memorizing those keywords and thus not learning the concepts
Wikipedia Section Title Prediction	24 %	80.12 % (mBERT)	

Dataset	Task	Our Test metric	Highest Test Metric mentioned in paper	Comments
HiNer (Published in ACL 2022)	**Named entity recognition**	70% (Macro avg f1-score)	86.98 ± 0.22% (Macro avg f1-score XLM-R large)	Considering that we use ELMo embeddings and use a biLSTM for the classification tasks, the models perform slightly worse than XLM-R. We primarily attribute this to the lack of attention mechanism in biLSTM.
IndicXnli (Published in EMNLP 2022)	**Natural language inference**	55.1% (Test accuracy)	78% (Test accuracy XLM-R)	

Analysis of proposed metric:

- Lower values of the metric indicate robust embeddings
- Median is not sensitive to outliers and seems to be appropriate
- Paired with the performance on the 100 samples, we can say that lesser value of metric and good performance on downstream task indicate good embedding quality.
- The proposed metric indicates embeddings obtained through trainable lambdas have high robustness, followed by original embeddings and frozen lambdas.
- A random combination of lambdas results in poor robustness.

Advantages of the proposed metric:

- Can evaluate contextual embeddings, unlike word similarity
- Non-reliance on cosine similarity, unlike word similarity
- Does not unfairly penalize task specific embeddings, unlike word similarity
- Can be useful for evaluating robustness of word embeddings

Disadvantages of the proposed metric:

- There are some corner cases where the metric would fail to detect bad embeddings. E.g. when all word vectors are initialized to the same or very similar values.
    - Solution: Apart from the proposed metric, take into consideration the model's performance on the downstream task using those 100 sentences. A bad word model would perform poorly, even if the proposed metric indicates good embeddings.
- Computational cost is higher
- Requires creation of an additional dataset which needs human resources.
- Dependency on the ability of trained model

References

[1] Hadj Taieb, M.A., Zesch, T. & Ben Aouicha, M. A survey of semantic relatedness evaluation datasets and procedures. *Artif Intell Rev* **53**, 4407–4448 (2020). https://doi.org/10.1007/s10462-019-09796-3

[2] Wang B, Wang A, Chen F, Wang Y, Kuo C-CJ. Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*. 2019;8:e19. doi:10.1017/ATSIP.2019.12

[3] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[4] Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

[5] Yang, Y., Wang, X. & He, K.. (2022). Robust textual embedding against word-level adversarial attacks. Proceedings of the Thirty-Eighth Conference on Uncertainty in

Artificial Intelligence, in Proceedings of Machine Learning Research 180:2214-2224 Available from https://proceedings.mlr.press/v180/yang22c.html.