Precog Recruitment Task
Programming Report

Amey Choudhary

March 27, 2025

Contents

1 Recruitment Task Overview

For my recruitment task, I chose the "NLP & Responsible AI" domain. I completed all the tasks in the section 'Language Representations', including **Part 1: Dense Representations**, **Part 2: Cross-lingual alignment**, and the **Bonus task on Harmful Associations**.

2 Part 1: Dense Representations

In this part, we explore various methods to generate word embeddings from a text corpus and assess their quality.

2.1 Building a Co-Occurrence Matrix

we used the Wortschatz Leipzig English News Corpus 2024 (300k) as the source text to generate word co-occurrence statistics. The raw corpus was preprocessed through the following steps:

1. Converted all text to lowercase for consistency.
2. Removed punctuation, numerical digits and ordinal expressions (e.g., `1st`, `2nd`, etc.).
3. The corpus was tokenized into individual words.
4. Common stop words are removed to reduce noise.
5. Applied lemmatization to reduce words to their base forms (e.g., *running* $\rightarrow$ *run*).

After preprocessing, we restricted the vocabulary to the top 50,000 most frequent words in the corpus to manage computational complexity. A symmetric co-occurrence matrix was then constructed using a context window size of 6. This window size was chosen based on two considerations. First, according to prior work [2], smaller windows (e.g., < 4) tend to capture syntactic relationships, while larger windows (e.g., > 8) are more suited to capturing semantic similarity. A window size of 6 provides a balance between the two. Second, given that the average sentence length in the corpus is approximately 11 tokens, a window size of 6 ensures that most context words within a sentence are captured effectively.

2.2 Dimensionality Reduction

To obtain dense word embeddings from the high-dimensional co-occurrence matrix, we applied two dimensionality reduction techniques: **Principal Component Analysis (PCA)** and **Random Projection (RP)**. While PCA captures

directions of maximum variance and is commonly used in NLP, Random Projection offers a computationally efficient alternative with faster runtime and lower memory overhead.

The original co-occurrence matrix of size $N \times N$ (with $N = 50{,}000$ vocabulary items) was projected down to a lower-dimensional space of size $N \times d$, where $d < N$. we chose $d = 100$ as the target dimensionality based on empirical findings[2] that show diminishing returns in embedding quality beyond 200 dimensions. Moreover, setting $d = 100$ allows for a fair comparison with pre-trained GloVe embeddings of the same dimensionality.

2.3 Evaluation Methodology

To evaluate the quality of the generated word embeddings, we used two widely accepted semantic similarity benchmarks: **SimLex-999** and **WordSim-353**. Each dataset consists of word pairs annotated with human similarity judgments.

For each word pair (w_1, w_2) in the datasets, we computed the cosine similarity between their corresponding embeddings:

$$\text{cosine}(w_1, w_2) = \frac{\vec{w_1} \cdot \vec{w_2}}{\|\vec{w_1}\| \|\vec{w_2}\|}$$

These model-predicted similarity scores were then compared against the human-annotated scores using Pearson correlation. A higher correlation indicates that the embeddings are better at capturing semantic similarity consistent with human perception.

This evaluation framework allowed for a principled way to compare embedding variants produced under different preprocessing, window size, and dimensionality reduction configurations.

2.4 Hyperparameter Study

In the initial stages of the project, choices such as window size, dimensionality (d), and dimensionality reduction method were guided primarily by prior work in the literature. For example, it is known that smaller context windows tend to emphasize syntactic relationships, while larger windows emphasize semantic similarity. Similarly, embedding dimensions beyond 200 often offer diminishing returns in downstream tasks.

However, these choices were made in the absence of a concrete evaluation criterion. With the integration of SimLex-999 and WordSim-353 as evaluation benchmarks, I was now able to conduct a systematic hyperparameter study.

The hyperparameter study examined the impact of:

- **Context Window Size:** Values such as 2, 4, 6, 8 and 10 were tested.
- **Embedding Dimensionality (d):** Explored 1, 50, 100, 150, 200 and 250 dimensions.

- **Dimensionality Reduction Method:** Compared PCA and Random Projection.

Each configuration was evaluated using cosine similarity and Pearson correlation on both benchmark datasets. The results were compiled into a comparative table, allowing for a data-driven selection of the most effective hyperparameter settings. A comprehensive summary of all evaluated configurations across different window sizes, dimensions, and reduction methods is provided in Appendix A in the Appendix.

2.5 Insights from Hyperparameter Study and Final Embedding Selection

From our hyperparameter study, the following observations are derived.

1. **Window Size:** The effect of context window size on embedding quality was found to be minimal. Interestingly, smaller window sizes performed marginally better. This may be due to the nature of the corpus—news articles typically contain concise and information-dense sentences. Furthermore, since stopwords and low-information words were removed during preprocessing, the effective context narrowed, reducing the need for larger windows to capture semantic meaning.

2. **Embedding Dimensionality:** Very low-dimensional embeddings (e.g., $d = 1$ or $d = 50$) significantly degraded correlation with human similarity judgments, indicating a loss of semantic information. Higher-dimensional embeddings (up to $d = 250$) provided consistently better performance, although improvements beyond $d = 200$ showed diminishing returns.

3. **PCA vs. Random Projection:** Random Projection outperformed PCA in nearly all configurations. This can be attributed to its ability to preserve pairwise distances while being less sensitive to the dominant directions of variance. RP avoids overfitting to the high-variance axes, making it better suited for sparse, high-dimensional data like co-occurrence matrices. Additionally, RP is computationally more efficient, allowing for scalable embedding generation.

To select the best-performing embedding, we normalized the SimLex-999 and WordSim-353 correlation scores for each configuration and computed their average. Based on this composite metric, the embedding generated using a window size of **2**, dimensionality $d = 250$, and **Random Projection** emerged as the top-performing configuration. This embedding was subsequently used for further downstream analysis.

2.6 Clustering and Visualization using K-Means and t-SNE

To analyze the semantic structure captured by the embeddings, we performed **K-Means clustering** followed by **t-SNE visualization**. First, the optimal number of clusters K was determined using the elbow method by plotting the within-cluster sum of squared distances (inertia) against increasing values of K. The point at which the curve visibly "bent" suggested a suitable balance between cluster compactness and overfitting, and was selected as the optimal K.

With the chosen value of $K = 20$, we applied K-Means to the 100-dimensional embedding vectors to obtain cluster assignments. To visualize these high-dimensional clusters, we used **t-distributed Stochastic Neighbor Embedding (t-SNE)** to reduce the embeddings to two dimensions. The resulting 2D scatter plot was color-coded by cluster and is shown in Figure 1.

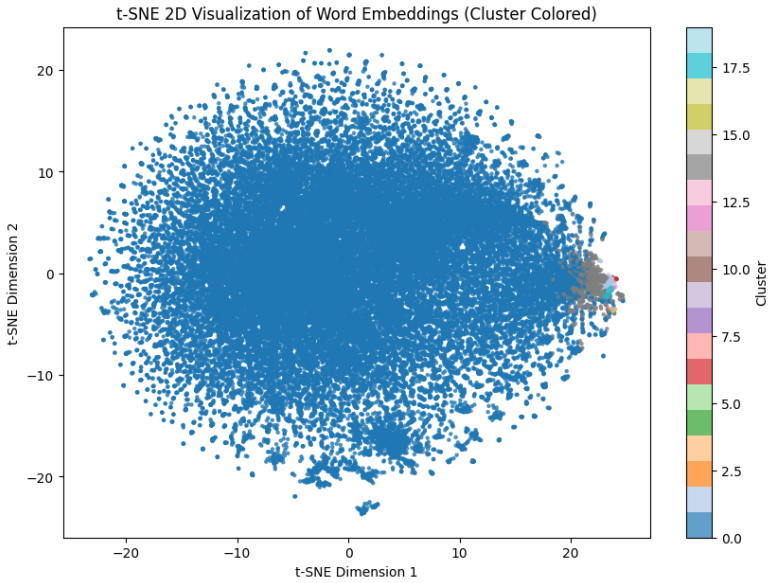

Figure 1: t-SNE 2D visualization of clustered word embeddings

The resulting clusters revealed semantically coherent groupings:

- **Cluster 2**: Included modal and auxiliary verbs such as *always*, *never*, *still*, *continue*, which represent general intent or continuity—common in natural discourse.
- **Cluster 7**: Contained high-frequency cognitive and action verbs like *see*, *think*, *want*, *go*, and *look*, representing mental and physical processes.

- **Cluster 8**: Grouped financial and business-related terms such as *stake*, *acquire*, *sell*, *purchase*, and *companys*, clearly forming a corporate activity theme.
- **Cluster 14**: Words related to temporal references like *time*, *day*, *one*.
- **Cluster 19**: Included ordinal and time-span references such as *four*, *five*, *later*, *earlier*, and *three*, forming a time-sequence grouping.
- **Other Clusters**: Also exhibited strong semantic groupings—e.g., *buy* (Cluster 3), *share* (Cluster 4), *stock* (Cluster 6), and *company* (Cluster 17).

These findings suggest that the learned embeddings are not only capable of encoding similarity in vector space but also of revealing interpretable semantic groupings when visualized and clustered appropriately. See Appendix B for representative word groupings in each cluster.

2.7 Google Word Analogy Task

To further evaluate the relational structure encoded by the word embeddings, we employed the **Google Word Analogy Task**, a widely-used benchmark consisting of 19,544 analogy questions. These questions are grouped into two broad categories:

- **Semantic Analogies**: These involve real-world relationships such as country–capital (e.g., *Paris* is to *France* as *Berlin* is to *Germany*), currency, and gender (e.g., *man* is to *king* as *woman* is to *queen*).
- **Syntactic Analogies**: These capture grammatical transformations such as verb tense (e.g., *run* is to *running* as *swim* is to *swimming*) and comparative forms (e.g., *fast* is to *faster* as *cold* is to *colder*).

Each analogy is framed as:

$$\text{“}a : b :: c :?\text{”}$$

and solved using vector arithmetic:

$$\vec{d} = \vec{b} - \vec{a} + \vec{c}$$

The model then selects the word from the vocabulary that is closest to $\vec{d}$ in terms of cosine similarity, excluding the original three input words. A `top_k` parameter is used to retrieve the closest k candidates optionally.

Given the limitations introduced by lemmatization and the absence of factual or proper noun retention in our cleaned corpus, we restricted evaluation to a subset of categories that rely less on encyclopedic or syntactic detail. Specifically, we evaluated the following relation types:

- `family`
- `adjective-to-adverb`
- `opposite`

Other categories, such as those involving country–capital pairs or verb inflections, were excluded either due to their reliance on external factual knowledge or because syntactic features were diminished during preprocessing (e.g., lemmatization removed tense distinctions).

Out of 1,615 analogy questions within the selected categories, our best-performing embedding correctly solved 22 (`top_k` = 10), yielding an overall accuracy of:

$$\text{Accuracy} = \frac{22}{1615} \approx 0.0136$$

While the accuracy is low, this is expected given the minimal relational structure in co-occurrence-only embeddings and the aggressive preprocessing. Nevertheless, the Google Analogy Task offers an additional and orthogonal perspective to similarity-based evaluations like SimLex-999 and WordSim-353.

2.8 Evaluation of Pretrained GloVe Embeddings

To establish a strong baseline, we evaluated the widely-used **GloVe** (Global Vectors for Word Representation) embeddings. We used the 100-dimensional pretrained GloVe vectors trained on a large corpus of 6 billion tokens (Wikipedia + Gigaword). Since our vocabulary was derived from a different corpus, we restricted the GloVe embeddings to only those words that overlapped with our vocabulary of 50,000 terms.

Despite this vocabulary filtering, GloVe embeddings demonstrated superior performance across all evaluation metrics compared to the embeddings we trained using co-occurrence statistics.

Similarity Benchmarks: GloVe achieved higher correlation with human similarity judgments:

- **SimLex-999:** 990 valid samples
  - Pearson Correlation: **0.296**
- **WordSim-353:** 324 valid samples
  - Pearson Correlation: **0.556**

Google Analogy Task: GloVe also significantly outperformed our embeddings in analogy completion:

- Total Correct: **249**
- Total Samples: 1615
- Accuracy: **15.42%**

Qualitative Analysis: The GloVe embeddings also produced more coherent and interpretable clusters when subjected to K-Means and t-SNE visualization (see Appendix C). The semantic groupings were clearer, and cluster boundaries were more distinct compared to those formed using our co-occurrence-based embeddings(see Appendix D).

These results reflect the advantage of training on a significantly larger corpus with global co-occurrence statistics and optimized objective functions, making GloVe a strong and effective benchmark for word representation quality.

3 Part 2: Cross-Lingual Alignment

In this section, we describe our approach to achieving cross-lingual alignment between English and Hindi representations. We focus on two complementary strategies: aligning static word embeddings via Procrustes analysis and refining sentence-level representations using a transformer-based model with contrastive loss.

3.1 Alignment Techniques and Evaluation Methods

3.1.1 Procrustes Alignment and Word-Level Translation (Top-k)

We employ 300-dimensional fastText embeddings derived from Common Crawl for both English and Hindi. These static embeddings serve as our initial representations and are used as input for the Procrustes analysis.

To align the embedding spaces of English and Hindi, we collected a set of bilingual word pairs to serve as anchor pairs. These anchor pairs represent known translations and are used to guide the mapping between the two embedding spaces.

We applied both single-step and iterative Procrustes analysis to these anchor pairs. In the Procrustes framework, we find an optimal orthogonal transformation that minimizes the Euclidean distance between the transformed English embeddings and their corresponding Hindi embeddings. Our experiments revealed that while the single-step approach failed to yield any correct translations, an iterative version—where the transformation is refined over a maximum of two rounds—produced promising results.

Evaluation: We evaluate the quality of the alignment using a test set of 200 English words. For each word, we retrieve the top-$k = 5$ closest Hindi vectors using cosine similarity and measure the translation accuracy. The experimental results are summarized below:

- **Single-Step Procrustes Alignment:** 0 correct translations.
- **Iterative Procrustes Alignment (up to 2 rounds):** 26 correct translations.

We also display the closest translated vector and their similarity for an example, '*politician*'.

```
Single-step alignment translations for politician:
[('बदचलन', 0.32026792), ('हूंअक्सर', 0.30049154), ('हूँअक्सर',
0.29704675), ('परअक्सर', 0.29599997), ('वहअक्सर', 0.2951046)]
```

Figure 2: Single Procrustes Alignment for '*politician*'

```
Iterative alignment translations for politician:
[('नेता', 0.34650037), ('छुटभैया', 0.321216), ('छुटभैये', 0.31190
825), ('अनदार', 0.30216), ('राजनेता', 0.3000589)]
```

Figure 3: Iterative Alignment for '*politician*'

These results indicate that iterative refinement significantly improves the alignment quality and facilitates more accurate word-level translation.

3.1.2 Transformer-Based Sentence-Level Translation Alignment

Motivated by recent advances in contextualized representations, we further refine our cross-lingual alignment at the sentence level using a transformer-based multi-lingual Sentence Transformer model. Drawing inspiration from [3], we propose a training strategy that leverages a contrastive loss to better align English and Hindi sentence embeddings.

Approach:

1. **Contrastive Loss:** We define a custom contrastive loss function that encourages the embeddings of paired English and Hindi sentences to be similar, while ensuring that embeddings of non-corresponding sentences in the same batch remain far apart. This is achieved by minimizing the distance between positive pairs and maximizing the distance between negatives, effectively improving the discriminative power of the sentence representations.

2. **Data and Training:** We utilize an English–Hindi parallel corpus from IIT Bombay[7] . Each sentence pair is processed through the model to obtain its sentence embedding. During training, the contrastive loss minimizes the distance between the embeddings of translated sentences, while pushing away the embeddings of unrelated sentences in the same batch.

Evaluation: At the end of training, we compute the cosine similarity between test sentence pairs before and after alignment. Our experiments demonstrate an improvement in the average similarity score from **0.69** (pre-alignment)

to **0.79** (post-alignment), indicating that our approach successfully brings semantically equivalent sentences closer in the embedding space while preserving distinctions between unrelated sentences.

Discussion: The transformer-based approach differs from the static word embedding alignment in several key aspects:

- **Problem Rephrasing:** According to [3], cross-lingual alignment can be understood from two perspectives:

  1. **Word-Level Alignment:** The nearest neighbor of a word vector in the target language should be its translated counterpart in the source language. This is typically achieved via Procrustes analysis, which aligns word embeddings across languages.
  2. **Task-Level Consistency:** When switching embeddings from one language to another for a given task, the task output should remain unchanged.

  In our work, we switch to the Task-Level Consistency view, and we use Sentence-BERT, which encodes sentences and is often employed for downstream tasks. Our objective is to ensure that sentence representations remain consistent across languages.

- **Contextuality:** While static embeddings assign a single representation to a word regardless of context, the Sentence Transformer produces context-sensitive embeddings. Therefore, the alignment process must account for sentence-level nuances.

- **Loss Function:** The contrastive loss applied in the transformer model operates over entire sentences, leveraging batch negatives to improve the robustness of the learned representations. This is fundamentally different from cosine-similarity-based evaluation used in static embedding alignment.

Overall, the integration of both static alignment methods and transformer-based fine-tuning offers complementary insights into cross-lingual representation learning. The observed improvements in sentence-level similarity after applying the contrastive loss highlight the effectiveness of our approach in reducing cross-lingual discrepancies.

4 Bonus tasks- Harmful Associations

In this part, we try to find harmful associations and bias in word embeddings, both static and contextual.

```
<Query: Female terms and Male Terms wrt Family and Careers
- Target sets: [['female', 'woman', 'girl', 'sister', 'she', 'her', 'hers', 'daughte
r'], ['male', 'man', 'boy', 'brother', 'he', 'him', 'his', 'son']]
- Attribute sets:[['home', 'parents', 'children', 'family', 'cousins', 'marriage', 'w
edding', 'relatives'], ['executive', 'management', 'professional', 'corporation', 'sa
lary', 'office', 'business', 'career']]>
```

Figure 4: Query generated

4.1 Calculating Harmful Associations using WEAT

We employ the Word Embedding Association Test (WEAT) as introduced in [1] to quantify harmful associations in static embeddings. In our approach, for a given word w, we define its association with two attribute sets A and B (e.g. career and family) as:

$$s(w, A, B) = \frac{1}{|A|} \sum_{a \in A} \cos(w, a) - \frac{1}{|B|} \sum_{b \in B} \cos(w, b).$$

For two target sets X and Y (e.g., male and female terms), the WEAT test statistic is given by:

$$S(X, Y, A, B) = \sum_{x \in X} s(x, A, B) - \sum_{y \in Y} s(y, A, B).$$

The effect size is computed using Cohen's d, which normalizes the difference of means by the pooled standard deviation. A higher absolute effect size indicates a stronger, potentially harmful association. This framework allows us to quantitatively assess bias in static word embeddings.

4.1.1 Using WEFE for WEAT Implementation

We utilized the WEFE toolkit to operationalize the WEAT framework. Specifically, WEFE was used to:

- Load pre-trained static embeddings and define the target sets (e.g., male and female terms) and attribute sets (e.g., career and family).
- Load pre-defined word sets which acted as target sets and attribute sets to make query(See figure 4)
- Use its inbuilt function to run query and obtain the WEAT metric result (See figure 5)

This streamlined approach allowed us to efficiently measure bias in static word embeddings.

Our WEAT score for the query *'Female terms and Male Terms wrt Family and Careers'* is 0.31, denoting that there is a moderate bias in the associations

```
{'query_name': 'Female terms and Male Terms wrt Family and Careers',
 'result': 0.3165842229500413,
 'weat': 0.3165842229500413,
 'effect_size': 0.6779441785725657,
 'p_value': 0.08798240351929613}
```

Figure 5: WEAT Query Result

between gender terms and the attributes of family and careers. It suggests that, on average, one gender's terms (e.g., female) are somewhat more strongly associated with family, while the other (e.g., male) tends to be somewhat more associated with careers.

4.2 Evaluating Harmful Associations in Contextual Embeddings using CrowS-Pairs

To measure bias in contextual models, we use the CrowS-Pairs dataset [4], which contains sentence pairs that differ only in stereotypical content (e.g., one sentence reinforces a racial stereotype while the other does not). The evaluation process is as follows:

1. **Dataset Filtering:** We first filter CrowS-Pairs to retain only sentence pairs related to racial stereotypes, ensuring our focus is specifically on racial bias.

2. **Pseudo-Perplexity Calculation with BERT:** For each sentence in the filtered pairs, we compute a pseudo-perplexity score using BERT. Since BERT is a masked language model, we approximate perplexity by masking each token in turn and computing the cross-entropy loss:

$$\text{Pseudo-Perplexity} = \exp\left(\frac{1}{N}\sum_{i=1}^{N} -\log P(w_i \mid w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_N)\right)$$

   where N is the number of tokens.

3. **Bias Measurement:** We compare the pseudo-perplexity scores of the two sentences in each pair. A lower score indicates that the model finds that sentence more likely. We then compute the proportion of pairs for which the stereotypical sentence is preferred.

4. **Results Interpretation:** Our analysis shows that in approximately **69**% of the racial stereotype pairs, BERT assigns a lower pseudo-perplexity to the stereotypical sentence, indicating a measurable bias in its contextual representations.

4.3 Differences from Static Word Embedding Analysis

- **Representation Type:** Static embeddings yield fixed vector representations for each word, while contextual models produce dynamic, context-sensitive sentence representations.
- **Evaluation Metrics:** For static embeddings, bias is quantified using cosine similarity and effect size via the WEAT framework. In contrast, for contextual models we approximate likelihood using pseudo-perplexity scores.
- **Granularity and Context:** Static analysis is conducted at the word level without regard to context, whereas our contextual evaluation considers full sentences, capturing nuanced effects of surrounding context on bias.

5 Code

For the purpose of this task, we used Kaggle to perform our experiments. The links to the notebooks are:

1. Task 1
2. Task 2
3. Bonus task

Disclaimer: AI tools like ChatGPT were used to provide code assistance as well as refining the content (in terms of grammar, vocabulary, fluency) used in report and presentation.

References

[1] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, April 2017.

[2] Dávid Držík and Jozef Kapusta. Effect of dimension size and window size on word embedding in classification tasks, 06 2024.

[3] Katharina Hämmerl, Jindřich Libovický, and Alexander Fraser. Understanding cross-lingual alignment – a survey, 2024.

[4] Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online, November 2020. Association for Computational Linguistics.

A Full Hyperparameter Evaluation Table

Window	Dim	Method	SimLex Corr	WordSim Corr
2	1	pca	-0.06138	-0.05544
2	1	rp	0.01529	0.13886
2	50	pca	0.00672	0.16688
2	50	rp	0.02767	0.14283
2	100	pca	0.01862	0.20625
2	100	rp	0.05125	0.19356
2	150	pca	0.02155	0.21806
2	150	rp	0.06475	0.17999
2	200	pca	0.02967	0.22611
2	200	rp	**0.06528**	0.20098
2	250	pca	0.03766	**0.23893**
2	250	rp	0.05974	0.21520
4	1	pca	-0.04352	0.02310
4	1	rp	-0.01948	0.05652
4	50	pca	0.01181	0.15775
4	50	rp	0.01596	0.16526
4	100	pca	0.01649	0.17749
4	100	rp	0.03774	0.19657
4	150	pca	0.01941	0.19811
4	150	rp	0.04074	0.18699
4	200	pca	0.02171	0.20087
4	200	rp	0.04403	0.20284
4	250	pca	0.02604	0.20430
4	250	rp	0.04097	0.20476
6	1	pca	-0.01227	0.01009
6	1	rp	-0.03546	0.02431
6	50	pca	0.00854	0.15628
6	50	rp	0.01465	0.14530
6	100	pca	0.01319	0.17453
6	100	rp	0.02945	0.18163
6	150	pca	0.01483	0.18290
6	150	rp	0.03129	0.16616
6	200	pca	0.01831	0.18497
6	200	rp	0.03525	0.18302
6	250	pca	0.02028	0.18680
6	250	rp	0.03160	0.17995
8	1	pca	-0.03356	0.03335
8	1	rp	-0.02856	0.01758
8	50	pca	0.00886	0.14931
8	50	rp	0.01388	0.13090
8	100	pca	0.01213	0.16716
8	100	rp	0.02390	0.17246
8	150	pca	0.01305	0.17327
8	150	rp	0.02719	0.15767
8	200	pca	0.01616	0.17704

Continued on next page

Table 1 – continued from previous page

Window	Dim	Method	SimLex Corr	WordSim Corr
8	200	rp	0.02818	0.16897
8	250	pca	0.01635	0.17838
8	250	rp	0.02538	0.16328
10	1	pca	-0.06148	0.04791
10	1	rp	-0.03439	-0.00202
10	50	pca	0.00729	0.15340
10	50	rp	0.01066	0.13929
10	100	pca	0.01140	0.16754
10	100	rp	0.02157	0.16733
10	150	pca	0.01213	0.17246
10	150	rp	0.02213	0.15574
10	200	pca	0.01469	0.17552
10	200	rp	0.02252	0.16784
10	250	pca	0.01453	0.17804
10	250	rp	0.02224	0.16158

B Cluster-wise Word Groupings from K-Means

The following table presents representative words from each of the 20 clusters discovered using K-Means clustering on the best word embeddings. These groupings reflect semantically coherent categories identified in the embedding space.

- **Cluster 1**: *sultana*, *caliendo*, *rosendahl*, *strapon*, *elokobi*, *taglines*, *chinau*, *maniyar*, *bradda*, *barletta*
- **Cluster 2**: *always*, *great*, *still*, *add*, *never*, *continue*, *try*, *plan*, *happen*, *didnt*
- **Cluster 3**: *buy*
- **Cluster 4**: *share*
- **Cluster 5**: *say*
- **Cluster 6**: *stock*
- **Cluster 7**: *see*, *think*, *want*, *come*, *know*, *work*, *would*, *go*, *also*, *look*
- **Cluster 8**: *stake*, *acquire*, *position*, *period*, *inc*, *sell*, *companys*, *purchase*, *llc*, *per*
- **Cluster 9**: *last*
- **Cluster 10**: *new*

- **Cluster 11**: *average*, *move*, *target*
- **Cluster 12**: *value*, *worth*
- **Cluster 13**: *year*
- **Cluster 14**: *time*, *one*, *day*
- **Cluster 15**: *fact*, *longer*, *conversation*, *agree*, *prepare*, *wouldnt*, *instead*, *although*, *fail*, *similar*
- **Cluster 16**: *rating*
- **Cluster 17**: *company*
- **Cluster 18**: *additional*, *quarter*
- **Cluster 19**: *four*, *five*, *later*, *past*, *six*, *earlier*, *since*, *old*, *end*, *three*
- **Cluster 20**: *own*

C Cluster-wise Word Groupings from GloVe Embeddings

Below are the representative word groupings discovered via K-Means clustering over GloVe embeddings, visualized using t-SNE. Each cluster highlights semantically or stylistically related words captured by the pretrained GloVe space.

- **Cluster 1**: *interestingly*, *vacuous*, *curiously*, *verbose*, *ponderous*, *ironically*, *predictably*, *delightfully*, *comically*, *undeniably*
- **Cluster 2**: *concoction*, *confection*, *preferably*, *yoghurt*, *chilis*, *sawdust*, *scoop*, *beeswax*, *croissant*, *curd*
- **Cluster 3**: *alienware*, *cofounder*, *acquires*, *vf*, *behemoth*, *trane*, *partnered*, *paragon*, *homeside*, *zenith*
- **Cluster 4**: *aforementioned*, *wherein*, *coincidentally*, *conjunction*, *describes*, *referencing*, *incorporating*, *inspiration*, *dubbed*, *explains*
- **Cluster 5**: *moreover*, *furthermore*, *likewise*, *consequently*, *meantime*, *importantly*, *additionally*, *whereby*, *hence*, *whereas*
- **Cluster 6**: *bodacious*, *bauble*, *minion*, *newbie*, *doppelganger*, *incidentally*, *boogeyman*, *schtick*, *monstrosity*, *imposter*
- **Cluster 7**: *dullness*, *nastiness*, *villainy*, *passivity*, *lastly*, *idiocy*, *clumsiness*, *conversely*, *weirdness*, *preoccupation*
- **Cluster 8**: *vicinity*, *underneath*, *riverbank*, *sprung*, *gigantic*, *beside*, *halfway*, *riverbed*, *stuck*, *stretched*

- **Cluster 9**: *whatsapp*, *selles*, *umaine*, *livs*, *sibiya*, *buchinger*, *lainey*, *bennu*, *hubspot*, *degrowth*
- **Cluster 10**: *alternatively*, *breakage*, *symptomatic*, *localized*, *fixation*, *snakebite*, *microorganism*, *precursor*, *sufferer*, *overload*
- **Cluster 11**: *chaturvedi*, *dahan*, *chaudhary*, *agarwal*, *ridwan*, *attah*, *sahu*, *mallam*, *shein*, *didi*
- **Cluster 12**: *ibarra*, *pacheco*, *villanueva*, *mendez*, *herrera*, *acosta*, *otero*, *vidal*, *tavares*, *rivas*
- **Cluster 13**: *ags*, *bmf*, *icb*, *iccs*, *nalc*, *mtas*, *fhc*, *wca*, *stc*, *tmo*
- **Cluster 14**: *bromyard*, *muswellbrook*, *abergele*, *coleshill*, *holmfirth*, *dapto*, *llanidloes*, *turriff*, *yarm*, *tenby*
- **Cluster 15**: *contextualize*, *subsume*, *emote*, *actualize*, *deconstruct*, *sidestep*, *generalise*, *enliven*, *clobber*, *jettison*
- **Cluster 16**: *arsenault*, *purdy*, *blevins*, *cronin*, *haggerty*, *finlayson*, *brophy*, *theis*, *hallock*, *mcdougall*
- **Cluster 17**: *gardner*, *bennett*, *hayes*, *nicholson*, *baker*, *freeman*, *payne*, *barker*, *patterson*, *bates*
- **Cluster 18**: *youngster*, *skipped*, *standout*, *midway*, *newcomer*, *contention*, *pairing*, *ensured*, *knock*, *duel*
- **Cluster 19**: *behest*, *demanded*, *asserted*, *insistence*, *insist*, *considers*, *refusing*, *handing*, *declaring*, *enraged*
- **Cluster 20**: *though*, *indeed*, *instead*, *instance*, *fact*, *although*, *yet*, *noting*, *turned*, *nevertheless*

D t-SNE 2D Visualization of GloVe Word Embeddings

E Data Samples from IIT Bombay English Hindi parallel corpus

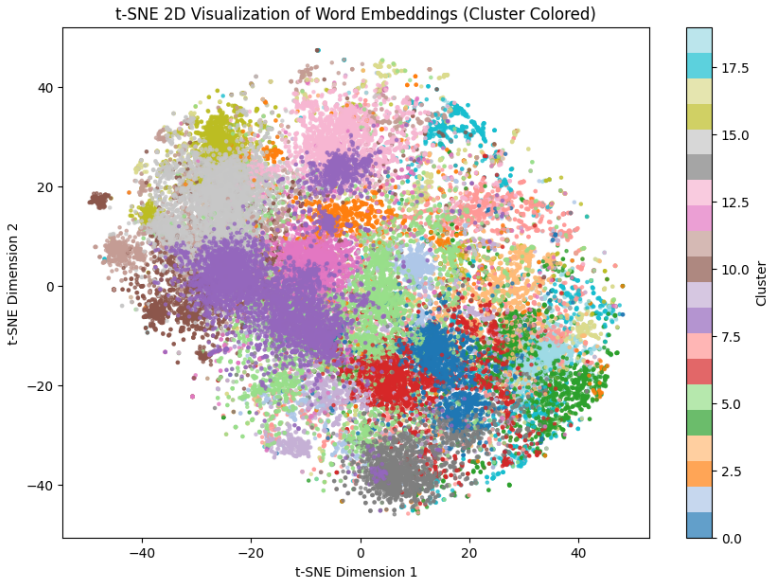

Figure 6: t-SNE 2D Visualization of GloVe Word Embeddings with K-Means Cluster Coloring

	hindi	english
0	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें	Give your application an accessibility workout
1	एक्सेर्साइसर पहुंचनीयता अन्वेषक	Accerciser Accessibility Explorer
2	निचले पटल के लिए डिफोल्ट प्लग-इन खाका	The default plugin layout for the bottom panel
3	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका	The default plugin layout for the top panel
4	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...	A list of plugins that are disabled by default

Figure 7: Sample Hindi-English pairs demonstrating various interface strings.