Assignment 3: Concepts
ExpandNet Implementation for English-to-Urdu Sense Projection

Chirooth Girigowda
girigowd@ualberta.ca
University of Alberta

Sahir Momin
smomin1@ualberta.ca
University of Alberta

1 Introduction

We implement an English-to-Urdu sense projection pipeline using **ExpandNet**. Specifically, we translate English sentences into Urdu using the **facebook/nllb-200-distilled-600M** model. For alignment, we employ **DBAlign** with a bilingual English-Urdu dictionary, which combines lexical resources from **Kaikki.org** and the **GoldenDict** project. Alignments between English and Urdu tokens are obtained, and senses from English tokens are projected onto corresponding Urdu lemmas based on these alignments. This workflow enables ExtendNet to function for the English-Urdu pair, showcasing the adaptability of the sense projection approach to low-resource languages. Further dataset details are provided in the Appendix (Data Description).

2 LLM Baseline

Our baseline investigates whether a large language model (LLM) can accurately determine the primary sense of an English gloss and generate an appropriate Urdu translation. The baseline uses **google/gemma-3-4b-it**[1], prompting the model with an English BabelNet synset gloss (definition) and tasking it to produce a single Urdu word that best matches the definition.

The **google/gemma-3-4b-it** model is accessed via the Hugging Face `transformers` library, supporting efficient, on-device text generation. All synsets are read from `se_gloss.tsv` (BabelNet ID and gloss). For each gloss, a prompt requests the best single Urdu word translation. Details of the prompt and rationale are included in the Appendix (LLM Baseline Prompt Design).

Manual inspection revealed four major failure modes in the LLM outputs:

1. The model often defaulted to generic verbs (e.g., "کرنا"(to do)), rather than specific content words.

2. Output contamination by English formatting (e.g., "you:coerce").

3. Failure to adhere to lemma constraints, producing multi-word phrases or agentive forms (e.g., "جدوجہدکرنےوالا"; 'struggler') rather than root lemmas.

4. Morphological fragmentation, most common with outputs like "دری" (dari), a non-lexical suffix or sub-word unit, resulting in invalid Urdu lemmas.

3 Method

We follow ExpandNet's official guidelines[2] to project senses from English to Urdu. The released code supports only BabelNet IDs but was adapted to process WordNet IDs, for which we confirmed results were identical. The final filtering step of ExpandNet's code checks if an English token and its Urdu lemma are paired in the dictionary before projecting senses. However, due to the limited coverage of our Urdu dictionary, some source tokens were missing. To mitigate this, we relax this constraint by the following assumption: if an English word is absent from the dictionary, we assume a correct translation and project its sense to the corresponding Urdu lemma.

3.1 Translation

We use sentence translations from Assignment 2. We also experimented with Urdu

[1] https://huggingface.co/google/gemma-3-4b-it

[2] https://github.com/UAlberta-NLP/ExpandNet

LLMs, including **large-traversaal/Alif-1.0-8B-Instruct** (Shafique et al., 2025) and **google/gemma-3-4b-it**, but these models produced inferior results compared to **facebook/nllb-200-distilled-600M**[3] (Koishekenov et al., 2023). Typical errors in LLM outputs included incorrect Urdu usage and shifts in sentence meaning, as confirmed by our native speaker(L1) and co-author.

3.2 Alignment

To improve English-Urdu alignment, we constructed a comprehensive bilingual dictionary by merging two primary resources: (1) open-source dictionary files from GoldenDict[4]; (2) a machine-readable Urdu dictionary from Kaikki.org[5], produced by Wiktextract (Ylonen, 2022). These were concatenated, cleaned (removing filler words), and used with **DBAlign**[6] for alignment.

3.3 Filtering

The same merged dictionary is used to confirm valid English-to-Urdu token pairs before projecting senses. If an English token is not found in the dictionary, we assume a valid projection (as above).

4 Analysis

Evaluation showed that **98** senses annotated by ExpandNet were correct out of **715** evaluated. While many incorrect senses may be plausible in alternate contexts, they most often reflected contextual ambiguity rather than translation defects.

Manual analysis of a 10-sentence sample found: 39 projected senses, of which 27 were correct and 12 incorrect. Most errors were due to misalignment; e.g., English nouns (including entities) aligned with Urdu filler words, such as "report" aligned to "ایک" ('one'), or "American" mapped to "میں" ('in'). Multi-word concepts also presented problems ("death penalty" sometimes reduced to "موت" ('death')).

Alignment errors were the main source: over-general senses, inconsistent mapping, and filler words in dictionary entries led to projection failures, especially for low-resource languages like Urdu.

Our error analysis indicates noise in the merged dictionary: a high frequency of function words and stopwords resulted in alignments between English content words and Urdu fillers, producing many spurious projections.

5 Results

Quantitatively, ExpandNet outperforms the LLM baseline: ExpandNet achieved a sense-level F1 of 9.9, compared to 3.1 for the LLM baseline. The full table of results appears in the Appendix (Table 1).

6 Conclusion and Discussion

We implemented and analyzed two approaches for projecting English semantic senses onto Urdu: ExpandNet and a generative LLM baseline. While both face challenges typical of low-resource languages, **ExpandNet produced consistently better results** (F1 of 10.0 versus 3.1 for the LLM).

The core limitation for ExpandNet proved to be **alignment quality**: reliance on incomplete dictionaries means contentful English words may be mapped to Urdu fillers, limiting sense projection accuracy. The LLM, despite producing some fluent translations, failed to meet strict requirements for sense disambiguation, by returning single-word lemmas out of given context.

Although both systems performed modestly, ExpandNet achieved higher precision and recall, especially at synset level. Addressing these core limitations will require improved alignment methods, more comprehensive dictionaries, and higher-quality parallel data which is a common need across low-resource languages. Without better resources, the effectiveness of sense projection approaches will remain constrained.

[3] https://huggingface.co/facebook/nllb-200-distilled-600M

[4] http://goldendict.org/dictionaries.php

[5] https://kaikki.org/dictionary/Urdu/words/index.html

[6] https://github.com/UAlberta-NLP/ExpandNet/blob/main/align_utils.py

References

Yeskendir Koishekenov, Alexandre Berard, and Vassilina Nikoulina. 2023. Memory-efficient

NLLB-200: Language-specific expert pruning of a massively multilingual machine translation model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3567–3585, Toronto, Canada. Association for Computational Linguistics.

Muhammad Ali Shafique, Kanwal Mehreen, Muhammad Arham, Maaz Amjad, Sabur Butt, and Hamza Farooq. 2025. Alif: Advancing urdu large language models via multilingual synthetic data distillation. *arXiv preprint arXiv:2510.09051*. Accepted to the EMNLP 2025 Workshop on Multilingual Representation Learning (MRL).

Tatu Ylonen. 2022. Wiktextract: Wiktionary as machine-readable structured data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1317–1325, Marseille, France. European Language Resources Association.

7 Appendix

7.1 Data Description

The dataset provided includes the following files:

- **se_gold_ur.tsv**: Urdu lemmas mapped to BabelNet synset IDs; empty rows indicate missing projections (gold standard).
- **corebnout.txt**: Filtered BabelNet IDs for restricted evaluation and sense selection.
- **se_gloss.tsv**: BabelNet synset IDs with English glosses (definitions); drives LLM prompting and evaluation.
- **se13_sentences.tsv**: Original English sentences, one row per sentence (with ID).
- **se13_tokens.tsv**: Per-token information (sentence ID, token, lemma, POS, etc.); required for alignment and projections.
- **se13.key.tsv**: Gold standard English sense annotations for every instance token.
- **xlwsd_se13.xml**: XML package of sentences, tokenization, and all sense annotations for input to WSD systems.

7.2 LLM Baseline Prompt Design

The LLM baseline used the following prompt :

"You are a bilingual lexicon expert. Given a dictionary definition: `[placeholder for BabelNet gloss]`*, produce the single word in Urdu that best matches this definition. Provide only the one Urdu word without explanations! DO NOT PROVIDE ANY OTHER OUPUT BUT THE URDU WORD!! Example (Do not include OUTPUT in your response, here INPUT and OUTPUT are only present to help you distinguish INPUT and OUTPUT, they should not be present in the your response), (Only the urdu word must be present in your response) Given INPUT prompt: You are a bilingual lexicon expert. Given a dictionary definition: "burden", produce the single word in Urdu that best matches this definition. Provide only the one Urdu word without explanations! DO NOT PROVIDE ANY OTHER OUTPUT BUT THE URDU WORD!! Expected OUTPUT response from you:* بوجھ- *DO NOT REPEAT THE INPUT PROMPT IN YOUR OUTPUT, ONLY GIVE THE URDU WORD!"*

The prompt design was iteratively refined to minimize off-target completions and suppress verbosity that LLMs might otherwise default to. The concise style and explicit example helped steer the model towards a consistent template-based output format. The prompt was carefully engineered to output only a single Urdu word as the response, avoiding extraneous text, but the model's response always contained the input prompt along with the Urdu word. The input prompt was the only extraneous information that the LLM included in it's output. Hence, the output from the LLM is saved in a temporary file and set to a postprocessing step where the model's response was cleaned by removing the input prompt from the output and storing only the Urdu word.

System(Sense)	**P**	**R**	**F1**
LLM Baseline	4.2	2.5	3.1
ExpandNet	**13.7**	**7.7**	**9.9**
System(Synset)	**P**	**R**	**F1**
LLM Baseline	4.2	4.1	4.2
ExpandNet	**15.8**	**12.4**	**13.9**

System(Synset)	**Core Coverage**
LLM Baseline	**6.1**
ExpandNet	5.1

Table 1: Performance comparison of LLM Baseline and ExpandNet systems on Urdu sense projection. The top two sections report precision(P), recall(R), and F1 at the sense and synset levels respectively. The bottom section reports core synset coverage in %.