Bitext Processing: Translation, Alignment, and Word Sense Disambiguation

Anonymous ACL submission

1 Word sense disambiguation

1.1 Method description

Data Preparation We load sentences and tokens from Excel files, formatting as JSON with `'lang': 'EN'`.

AMUSE API Integration Data are sent to the AMUSE WSD API, returning tokenized text with BabelNet synset IDs.

Token Alignment and Output We apply character-level matching with hardcoded corrections for mismatches (e.g., "'s" → "s", number formatting), writing synset IDs for tokens with instance IDs to `amuse_output.key` .

Evaluation We compare output against `se13.key.txt` using `evaluate_wsd.py`.

1.2 Accuracy

The system achieved an accuracy of 0.621.

1.3 Examples of typical or interesting errors

Polysemous Word Disambiguation: Words with multiple senses (e.g., "plan", "world") contribute significantly to errors.

Domain-Specific Terminology: Professional vocabulary from specialized domains (e.g., 'Washington', 'Technology') poses significant disambiguation challenges.

1.4 Reflection

Strengths: Direct API integration with straightforward token matching achieves full coverage.

Weaknesses: Hardcoded corrections are brittle and non-generalizable, resulting in 62.1% accuracy.

Limitations: Manual tokenization fixes cannot scale to new data, and API dependency prevents domain-specific optimization.

2 Translation

Method Description We configure GoogleTranslator with `source='auto'` and `target='zh-CN'`, preprocess sentences by normalizing whitespace, and translate sequentially with error handling. Results are saved to `translations.txt` and evaluated with CometKiwi, producing `translation_scores.txt` with one score per line.

2.1 Evaluation results

The dataset contains 301 sentences with a system score of 0.749 (range: 0.172-0.887, SD: 0.117). Quality distribution: 130 high ($\geq$0.8, 43.2%), 86 good (0.7-0.8, 28.6%), 48 moderate (0.6-0.7, 15.9%), and 37 low (<0.6, 12.3%).

2.2 Error Analysis

Two main error categories were identified:

Complete Translation Failures: Sentences 41 and 264 scored below 0.2, returning empty or untranslated text due to complexity and API errors.

Proper Noun Translation Issues: Sentences scoring 0.4-0.6 left proper nouns untranslated or inconsistently transliterated.

2.3 Reflection

Strengths: Achieves 0.749 system score with simple sequential processing and robust error handling.

Weaknesses: Performance degrades on terminology-heavy content (12.3% scored below 0.6), with two complete failures.

Limitations: Lacks domain-specific tuning, relies on generic GoogleTranslator without customization, and sequential processing is slower than batch methods.

3 Word Alignment

3.1 Method description

Hyperparameter(s) for SimAlign we tried is only itermax for the matching method, and we used the default values for other hyperparameters.

3.2 Evaluation and Error Analysis

We found some successes and errors in the alignment output, here are some examples:

1. In the 1st sentence, “U.N.” and “联合国” are aligned correctly, “reduce” and “减少” are aligned correctly, and “emissions” and “排放” are aligned correctly. **2.** In the 13th sentence, “by” and “不是” are incorrectly aligned. **3.** In the 58th sentence, "oil" aligns to “石油” and "公司" at the same time. The alignment with “石油” is correct, but the alignment with “公司” is incorrect.

As mentioned above, two typical alignment errors are function words and one-to-many mappings. The former is because function words encode grammar, not meaning, so embedding-based aligners like SimAlign struggle to capture their true correspondences, especially when sentences are restructured across languages. The latter is because itermax’s purely distributional approach leads to over-alignment when multiple target tokens have similar embeddings to the source.

3.3 Reflection

To improve alignments, we could add function word filtering using POS tags to prevent incorrect alignments of articles and prepositions. We could also incorporate compound word detection to treat multi-word expressions as single units, reducing spurious one-to-many mappings.

4 Sense Projection

4.1 Method description

The system projects BabelNet synset IDs from English to Chinese by loading gold sense annotations from "se13.key.txt" and establishing correspondences between tokens in "se13_tokens.xlsx", "english_tokens.txt", and "chinese_tokens.txt" using case-insensitive string matching. It then follows word alignments from "alignments.txt" to transfer senses, only projecting when an English token has a unique sense annotation in the sentence to avoid ambiguity. The output is "senses.tsv", a tab-separated file pairing synset IDs with their aligned Chinese tokens.

Here’s a short example sentence: "The U.N. group drafts plan to reduce emissions," the tokens "group" (bn:00041942n), "plan" (bn:00062759n), and "reduce" (bn:00027473n) are aligned to Chinese tokens "小组", "计划", and "排放" respectively. The system successfully projects these senses, producing: bn:00041942n→小组, bn:00062759n→计划, and bn:00027473n→排放.

4.2 Reflection

The sense projection worked well for concrete nouns and verbs with clear one-to-one alignments. Main issues included missing alignments from structural divergences and incorrect alignments causing semantically mismatched projections. The uniqueness constraint prevented ambiguous transfers but reduced coverage by excluding repeated tokens with different senses.

5 Overall Report

5.1 Introduction

This work implements a bitext processing pipeline for word sense disambiguation, machine translation, word alignment, and sense projection using AMUSE API, GoogleTranslator, CometKiwi, and SimAlign on 301 English sentences from the SE13 dataset.

5.2 Discussion

The pipeline achieved moderate performance: WSD reached 62.1% accuracy, translation scored 0.749 (CometKiwi), and alignment successfully identified many correspondences. Error propagation across components impacted final sense projection quality, with alignment failures preventing accurate sense transfer.

5.3 Conclusion

Pipeline architectures face compounding errors. Key improvements needed: robust tokenization for API integration, function word filtering for alignment, and compound expression detection to reduce spurious mappings.

5.4 Citations

Tools used: AMUSE API (Bevilacqua et al., 2021) with BabelNet (Navigli and Ponzetto, 2012), deep-translator, CometKiwi (Rei et al., 2022), SimAlign (Sabet et al., 2020), and SE13 dataset (Navigli et al., 2013).

References

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent Trends in Word Sense Disambiguation: A Survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193:217–250.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643.