

JSP3: Report for task WSD-C

Vira Yurchenko, Stanislav Lahno, Barbora Mahdalová
yurcvi00@upol.cz, lahnst00@upol.cz, mahdba00@upol.cz
veray15, LStandová, bMahdalová

January 2026

1 Introduction

The goal of this project is to automatically identifying the correct meaning (sense) of words in a text and give it a sentiment value.

The main task is word sense disambiguation and lexical sentiment analysis for a JSON corpus with sentences.

The assignment is:

- Look at the context around each important word.
- Choose which meaning of the word is most appropriate for that place.
- Give the word a sentiment score: from -100 (negative) to +100 (positive) or 0 (neutral).

Ollama, a small AI model used in the project, reads context and selects the appropriate meaning from the WordNet dictionary.

This project aims to automatically add a sentiment and tag words in a text corpus with their most appropriate sense.

2 Background

In this project we used mainly Open English WordNet (oewn:2024) and Ollama for running large language models locally. We tested two models: qwen2.5:7b version (the size is 7B parametres, this model is much slower) and the qwen2.5:3b version (the size is 3B parametres, this model is faster).

Resources

- Open English WordNet (oewn:2024)
- Ollama model qwen2.5:7b
- Ollama model qwen2.5:3b
- JSON file (twwtn-enhuman.json)

3 Approach

The full code is available here: [google colab](#).

First, we tried to test Ollama model locally on our computers. As we have regular notebooks, we couldn't test big models. So these models are quite small and don't work so fast as we expected.

In the Google Collab we installed GPU and ran our code there. There was used JSON file with texts in English.

Generally, we took each lemma and look at its context, used wn library for possible meanings and asked the model to give a sentiment score. We tested the size of context with such principle: 2 sentences before and after around the sentence.

We started with the bigger model (7b), but it was too slow, one small test took more than 3 hours (we took the range 110001:110005). So we switched to the smaller 3b model, which finished the same test in 2 hours (nearly). In result we took a little bigger range (110001:110006). With the first range model processed 3 sentences, with the second one it processed 4 sentences.

At the end we saved all the chosen meanings, scores and a summary: which meanings were chosen most often, what was the average sentiments of the words, etc. The results were saved in JSON files and simple statistics reports for each test.

4 Results

We tested four different context sizes (0, 1, 2, and 3 sentences before and after the current sentence). Here are the main results from the statistics files

(output):

Context size 0(only the current sentence):

- "processed sentences": 4,
- "total concepts": 11,
- "tag counts": "per": 4,
- "sentiment stats": "mean": -6.8
- "non zero sentiments": 1

Context size 1:

- "processed sentences": 4,
- "total concepts": 156,
- "tag counts": "per": 3, "w": 3, "x": 3
- "non zero sentiments": 5,
- "sentiment stats": "mean": -24.3

Context size 2:

- "processed sentences": 4,
- "total concepts": 241,
- "tag counts": "per": 3, "x": 6, "w": 1
- "non zero sentiments": 3,
- "sentiment stats": "mean": -33.0

Context size 3:

- "processed sentences": 4,
- "total concepts": 270,
- "tag counts": "x": 5, "per": 5, "w": 1
- "non zero sentiments": 6,
- "sentiment stats": "mean": -44.0

5 Discussion

From the statistics, we can see, that small context (size 0) gives the most neutral sentiment (-6.8) and more "per" (person) tags (which is correct for names). Big context (size 3) makes sentiment too negative (-44.0) and increases "x".

Therefore larger context increases the number of processed concepts, but also makes the model choose more "x" (closed-class words) and "w" (wrong sense) tags. Sentiment scores become more negative as context grows. It is too negative even for neutral words (names like "Karel", "Čapek", "David"...). Average sentiment drops from -6.8 (size 0) to -44.0 (size 3). This shows that the model qwen2.5:3b gets confused with long text.

With larger context (size 2 and 3), the number of concepts grows a lot (from 11 to 270).

This result shows that more context does not always help, the small model gets lost in long text.

We had some challenges during this work. In general, it was quite new kind of project for us, because we have never used Ollama language model. Despite this, it was really useful and interesting experience for us.

Problems we had in this project:

1. When we used the bigger model (qwen2.5:7b), one small test with only a few sentences took more than 3 hours. We chose the smaller one (qwen2.5:3b) and the same test was finished in 2 hours.
2. The model didn't always follow instructions. Sometimes it returned "None" (*Selected key: None*) instead of choosing a tag from the list. Other times it picked tags like "x" or "w". This happened more often when the context was long (context size 2 or 3). We think the reason is that the prompt became very big and the small model got confused.
3. Also there is a lot of negative sentiment even if a word is neutral.

For future work:

We should use a bigger model and better computer system to get better results. We could try to improve prompts, so the model never returns "None"

and gives more neutral sentiment for names. Absolutely we need to test on a bigger range (for example, 110001:110031) to see if context helps with specific terms.

6 Conclusions

We built and tested the code, that uses a JSON corpus to determine a word's correct sense and assign a sentiment score. There was used small Ollama model for reading the context and choosing the meaning from WordNet. We ran tests with different context sizes: 0, 1, 2, and 3 sentences before and after the current sentence.

The result showed that for this model, it is often better to use a small context, because long text makes it lose focus.

Finally the code worked well. It processed the text, saved tagged JSON files and created statistics files for each context size. This helped us to easily compare the results. In the future we could use a bigger model, better prompts or test on more parts of text to get better results.

* The main part of the work was done by Vira Yurchenko and Stanislav Lahno. We wrote the code from the beginning to the end. From the middle of the project Barbora Mahdalová helped by suggesting us her own ideas on how to improve the code (as she joined the group in the middle of the semester).