Upaya at ArabicNLU2024: Arabic Lexical Disambiguation using Large Language Models

Pawan Kumar Rajpoot
SCB DataX Thailand
pawan.rajpoot2411@gmail.com

Ashvini Kumar Jindal
LinkedIn AI USA
ajindal@linkedin.com

Ankur Parikh
UtilizeAI Research India
ankur.parikh85@gmail.com

Abstract

Disambiguating a word's intended meaning (sense) in a given context is important in Natural Language Understanding (NLU). WSD aims to determine the correct sense of ambiguous words in context. At the same time, LMD (a WSD variation) focuses on disambiguating location mention. Both tasks are vital in Natural Language Processing (NLP) and information retrieval, as they help correctly interpret and extract information from text. Arabic version is further challenging because of its morphological richness, encompassing a complex interplay of roots, stems, and affixes. This paper describes our solutions to both tasks, employing Llama3 and Cohere-based models under Zero-Shot Learning and Re-Ranking, respectively. Both the shared tasks were part of the second Arabic Natural Language Processing Conference co-located with ACL 2024. Overall, we achieved 1st rank in the WSD task (accuracy 78%) and 2nd rank in the LMD task (MRR@1 0.59).

1 Introduction

The First shared task is Word Sense Disambiguation in Arabic (Khalilia et al., 2024)(WSD), which aims to determine a word's intended meaning (sense) in a given context. Word Sense Disambiguation (WSD) is a long-standing challenge in Natural Language Processing (NLP), and it has an extended history of research. (Bevilacqua et al., 2021). In the shared task ArabicNLU2024 WSD task (Jarrar et al., 2023; Malaysha et al., 2023), Arabic language adds further challenges in WSD as Arabic exhibits morphological richness, encompassing a complex interplay of roots, stems, and affixes, rendering words susceptible to multiple interpretations based on their morphology.

Second task Location Mention Detection (LMD) is a special case of WSD. Disambiguating location/geolocation from social media posts is very useful in disaster management, as it helps response authorities, for example, locating incidents for planning rescue activities and affected people for evacuation. (Suwaileh et al., 2023), An LMD system aims at matching location mentions (LMs) appearing in microblogs toponyms, i.e., place or location names, in a geo-positioning database, i.e., gazetteer. (Bennett, 2010), The LMD problem can be typically decomposed into two sub-problems: 1) candidate retrieval (which aims to retrieve a list of candidate toponyms from gazetteer), 2) and candidate ranking (which aims to rank the list of retrieved candidates).

In this paper, we employed zero-shot learning on Llama3 (AI@Meta, 2024), to solve WSD task as training data wasn't provided. To solve LMD task, we employed OpenStreetMap API (Bennett, 2010), for candidate retrieval and we fine-tuned Cohere models [1] for re-ranking with the training data.

2 Literature Review

The task of resolving word sense ambiguity has attracted significant research efforts. These approaches can be broadly categorized into two main paradigms:

- Knowledge-Based Methods: These methods leverage external resources, such as BalkaNet (Tufis et al., 2004), BabelNet (Navigli and Ponzetto, 2013), IMS (Zhong and Ng, 2010), or manually constructed knowledge bases, to capture the semantic relationships between words and identify the intended sense in a given context (Banerjee et al., 2003; Basile et al., 2014; Wang et al., 2020; Kwon et al., 2021).

- Machine Learning Methods: Supervised machine learning approaches (Iacobacci et al., 2016; Papandrea et al., 2017; Zhang et al., 2021; Le et al., 2020), utilize pre-labeled data

[1] https://cohere.com/blog/rerank

Proceedings of The Second Arabic Natural Language Processing Conference, pages 377–382

to train models for predicting the most appropriate sense of a word. Advancements in deep learning yielded promising results in WSD tasks. While requiring significant computational resources, these methods often outperform knowledge-based approaches because they can learn complex patterns within language data.

For Arabic specifically (Al-Hajj and Jarrar, 2021), created a dataset of labeled Arabic context-gloss pairs (around 167k pairs) using Arabic Ontology and benchmarked this dataset using three fine-tuned Arabic BERT models.

The recent emergence of large language models (LLMs) such as GPT-3 represents a significant advancement in natural language processing (NLP) (Brown et al., 2020; Thoppilan et al., 2022). These models have expertise in a variety of domains, and hence, they can be used as it is in multiple NLP tasks. Traditional language models use separate pre-training and fine-tuning pipelines (Devlin et al., 2019; Maheshwari et al., 2021; Lan et al., 2019; Zhuang et al., 2021), where the fine-tuning stage follows pre-training. Models are fine-tuned on a task-specific dataset in a fully-supervised manner. The recent large language models such as Llama-3 and GPT-4 (Achiam et al., 2023), are improving at handling multiple languages and providing decent accuracy with zero-shot learning / only with prompt instructions (where we define the required task instruction in detail). These models have become the preferred choice when training data is not available.

3 Preliminary Background

3.1 WSD

Given the following inputs: A sentence S, a word w that needs to be disambiguate, and a set of possible definitions D = d_1, d_2, ... , d_n; where di is one possible definition of w, the WSD system tries to select the right definition di of w for the context S.

3.2 LMD

Given the following inputs: A post p (tweet in our dataset), a set of location mentions (LMs) Lp = l_i; i in [1,n_p] in post p, where l_i is the ith location mention, and n_p is the total number of location mentions in p and a geo-positioning database G (i.e., OSM) that consists of a set of toponyms T = t_j ; j in [1,k] where t_j is the j_th toponym, and k is the number of toponyms in G, the LMD system aims to match every location mention l_i in the post p to one of the toponyms tj in OSM that accurately represents it.

4 Data

4.1 WSD

SALMA corpus (Jarrar et al., 2023), is the first sense-annotated corpus for Arabic. SALMA contains 1,440 sentences and 34K tokens (8,760 unique tokens with 3,875 unique lemmas). All tokens are sense-annotated manually, with a total of 4,151 senses. The participants were provided with the development and test datasets in the shared task. The development set consists of 100 sentences randomly selected from SALMA, the set of candidate senses (glosses), and the target/correct sense for each word in each sentence. The rest of the SALMA corpus (1,340 sentences) was shared as a test set. The test set is similar to the development set but will not include the target/correct senses. No training dataset was provided.

4.2 LMD

IDRISI-DA (Suwaileh et al., 2023), is the first Arabic LMD dataset in the disaster domain. It encompasses 2,869 posts from diverse dialects, featuring 3,893 location mentions, of which 763 are unique, across seven countries. In the shared task, the data was shared with the standard 70:10:20 splits per event. For each post/tweet in the train and dev datasets, annotations of the location mentions were provided, where each location mention is accompanied by its correct toponym from the OpenStreetMap (Bennett, 2010), (OSM) gazetteer. Each toponym includes several attributes such as geo-coordinates, address, etc.

5 WSD with Zero-Shot Learning

As there is no training data present in WSD task, we leverage prompt based approaches to solve the task.

First, we manually convert the WSD problem statement to Natural language based task description and use it consistently for all prompt based experiments. In WSD task, each input consists of Arabic sentence, Arabic word, and list of definition choices. In our first experiment, we prompt models with WSD task description with input defined above to generate the correct option choice. We

refer to this approach as "Base Prompt". To enforce input format, we convert all parts of an input to a JSON formatted string and prompt the model to generate output in a fixed schema. We refer to this approach as "Structural Input and Output". We found 6% accuracy improvement for Llama-3-70B-Instruct model and 0.4% drop in accuracy for GPT-4-Turbo. This suggests that GPT-4-Turbo's performance is agnostic to input/output format and Llama-3-70B-Instruct understands structured inputs more accurately than raw inputs. To further handle the complexity of the Arabic language, we also explored In-Context learning based methodology. Specifically, we used Llama-3-70B-Instruct model to generate Arabic sentences for a set of words with their true definitions present in Arabic sentences. After getting Arabic sentences, we pass these sentences along with true definitions as few-shot examples for the model. We refer to this approach as "Structural Input and Output + In-Context Learning". However, we found that this approach hurt the accuracy of the dev set. This also suggests that more experiments are required for effective In-Context learning capability.

5.1 Results

Table 2 shows a 6% accuracy improvement for the Llama-3-70B-Instruct model with a "Structural Input and Output" based prompt approach. Due to the inference cost efficiency of Llama-3-70B-Instruct, we opted to use this prompting approach for final submission. Overall, our team achieved first rank on the test set with a test accuracy of 77.82%

6 LMD with LLM-based Re-Ranking

The approach consists of two stages: 1. Candidate Retrieval 2. Candidate Re-Ranking.

6.1 Candidate Retrieval

During this stage, for each location mention l_i of the post p, we query OSM to get the candidate toponyms. As per our analysis on the training data, for approximately 30% of the query results, either the OSM API didn't return the right toponym or returned a null value. The specific cause behind this behavior remains unclear and warrants further investigation. It is hypothesized that this limitation in the dataset and/or the OSM API's performance may have contributed to capping our performance on the test data. Figure 3 in Appendix gives the query result structure as opposed to the gold data. The example shows that the OSM API does not retrieve the correct toponym ID for the location mention 'France.

6.2 Candidate Re-Ranking

The high-level description of this stage is shown in Figure 1. Since the training data and development data were available for this task, we employed a Cohere rerank-multilingual-v2.0 model to re-rank candidate toponyms. The Cohere Rerank [2] model is a state-of-the-art neural network architecture designed to re-ranking candidate results (Cohere, 2023). This model has been successfully applied in various natural language processing tasks, such as document retrieval and question answering. The Cohere rerank-multilingual-v2.0 model leverages advanced techniques, including self-attention mechanisms and transformer-based architectures (Vaswani et al., 2017), to effectively capture the semantic relationships between the query and the candidate results. By learning to assign higher scores to the most relevant candidates, the model can significantly improve the accuracy of the re-ranking process.

6.3 Results

For the test data (791 tweets), our model achieved an accuracy/MRR@1 of 0.5994. The results of the test dataset are shown in Table 2. We stood second in the overall ranking. Model details can be accessed on the cohere fine-tuning portal [3]. The possible explanation behind the huge gap between the 1st and 2nd results could be the issues we mentioned in the Candidate Retrieval stage.

6.4 Future Work

We noticed that the OSM API doesn't return the right toponym or a null value for a significant portion of the location mentions. We do not run our rerank pipeline further for such cases. As mentioned before, this reduced our performance significantly. Assuming that annotations and the OSM API are correct, we believe that query expansion and/or using additional gazetteers can enhance our system's performance.

[2] https://cohere.com/blog/rerank

[3] https://dashboard.cohere.com/fine-tuning/custom/bf502547-df9f-4719-af4d-e7a70d640b4d

Model	Prompt Approach	Dev Accuracy
GPT-4-Turbo	Base Prompt	79.14%
Llama-3-70B-Instruct	Base Prompt	75.1%
GPT-4-Turbo	Structural Input and Output	78.74%
Llama-3-70B-Instruct	**Structural Input and Output**	**81.4%**
Llama-3-70B-Instruct	Structural Input and Output + In-Context Learning	76.11%

Table 1: Results on WSD dev set: we explore different prompting approaches. Our final submission for the WSD task is based on the approach marked in **bold**. Refer to Figure 2 (Appendix) for the exact prompt.

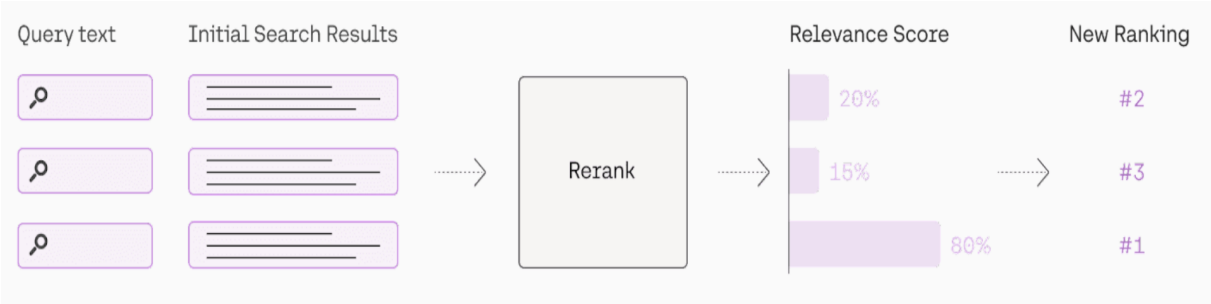

Figure 1: Re-ranking using Cohere Re-Rank model. Given the Query text: Mention + tweet, the candidate toponyms were returned by the OSM API. Those toponyms were finally re-ranked using the Cohere Re-Rank model.

Overall Rank	Accuracy
Team Upaya (ours)	77.82%
Team Pirates	70.78%
Team Rematchka	57.52%
Baseline(with train)	84.2%

Table 2: Results on WSD test dataset.

Top K	MRR 1
Team Rematchka	0.9497
Team Upaya(ours)	0.5994
Baseline(OSM)	0.572

Table 3: Results on LMD test dataset.

7 Conclusion

This work explores the use of large language models for resolving lexical ambiguity in the Arabic language. Two lexical ambiguities are addressed here: 1. Word Sense Disambiguation (WSD) and 2. Location Mention Disambiguation (LMD). Since no training data was available for the WSD task, we employed the Llama-3 model with Zero-Shot Learning. Our WSD system achieved the 1st rank with an accuracy of 0.7782. For the LMD task, we used Cohere models to re-rank the toponym candidates retrieved using OpenStreetMap API. Our LMD system secured the 2nd spot with MRR@1 of 0.599.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Moustafa Al-Hajj and Mustafa Jarrar. 2021. Arabgloss-bert: Fine-tuning bert on context-gloss pairs for wsd. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 40–48, Online. INCOMA Ltd.

Satanjeev Banerjee, Ted Pedersen, et al. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Ijcai*, volume 3, pages 805–810.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600.

Jonathan Bennett. 2010. *OpenStreetMap*. Packt Publishing Ltd.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. In *International Joint Conference on Artificial Intelligence*, pages 4330–4338. International Joint Conference on Artificial Intelligence, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

IGNACIO JAVIER Iacobacci, MOHAMMED TAHER Pilehvar, Roberto Navigli, et al. 2016. Embeddings for word sense disambiguation: An evaluation study. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016-Long Papers*, volume 2, pages 897–907. Association for Computational Linguistics (ACL).

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammed Khalilia. 2023. SALMA: Arabic sense-annotated corpus and WSD benchmarks. In *Proceedings of ArabicNLP 2023*, pages 359–369, Singapore (Hybrid). Association for Computational Linguistics.

Mohammed Khalilia, Sanad Malaysha, Reem Suwaileh, Mustafa Jarrar, Alaa Aljabari, Tamer Elsayed, and Imed Zitouni. 2024. Arabicnlu 2024: The first arabic natural language understanding shared task. In *Proceedings of the 2nd Arabic Natural Language Processing Conference (Arabic-NLP), Part of the ACL 2024*. Association for Computational Linguistics.

Sunjae Kwon, Dongsuk Oh, and Youngjoong Ko. 2021. Word sense disambiguation based on context selection using knowledge-based word similarity. *Information Processing & Management*, 58(4):102551.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Duong Le, My Thai, and Thien Nguyen. 2020. Multi-task learning for metaphor detection with graph convolutional neural networks and word sense disambiguation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8139–8146.

Himanshu Maheshwari, Bhavyajeet Singh, and Vasudeva Varma. 2021. SciBERT sentence representation for citation context classification. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 130–133, Online. Association for Computational Linguistics.

Sanad Malaysha, Mustafa Jarrar, and Mohammed Khalilia. 2023. Context-gloss augmentation for improving Arabic target sense verification. In *Proceedings of the 12th Global Wordnet Conference*, pages 254–262, University of the Basque Country, Donostia - San Sebastian, Basque Country. Global Wordnet Association.

Roberto Navigli and Simone Paolo Ponzetto. 2013. An overview of babelnet and its api for multilingual language processing. *The People's Web Meets NLP: Collaboratively Constructed Language Resources*, pages 177–197.

Simone Papandrea, Alessandro Raganato, and Claudio Delli Bovi. 2017. Supwsd: A flexible toolkit for supervised word sense disambiguation. In *Proceedings of the 2017 conference on empirical methods in natural language processing: system demonstrations*, pages 103–108.

Reem Suwaileh, Tamer Elsayed, and Muhammad Imran. 2023. Idrisi-d: Arabic and english datasets and benchmarks for location mention disambiguation over disaster microblogs. In *Proceedings of ArabicNLP 2023*, pages 158–169.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Dan Tufis, Dan Cristea, and Sofia Stamou. 2004. Balkanet: Aims, methods, results and perspectives. a general overview. *Romanian Journal of Information science and technology*, 7(1-2):9–43.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Yinglin Wang, Ming Wang, and Hamido Fujita. 2020. Word sense disambiguation: A comprehensive knowledge exploitation framework. *Knowledge-Based Systems*, 190:105030.

Chun-Xiang Zhang, Rui Liu, Xue-Yao Gao, and Bo Yu. 2021. Graph convolutional network for word sense disambiguation. *Discrete Dynamics in Nature and Society*, 2021(1):2822126.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 system demonstrations*, pages 78–83.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Appendix

```
[
  {
    "role": "system",
    "content": "You are an Arabic linguist and have deep expertise in analyzing Arabic text."
  }

  {
    "role": "user",
    "content": #Task:
               You are given a JSON string that contains an Arabic sentence, a word present in the sentence, and a list
of choices for word's definition. Each definition choice contains an option choice (eg: 'A', 'B') as key and potential
word definition as value. Your job is to disambiguate provided Arabic word in the context by picking the right
definition of the word in a given sentence. Only one out of all provided definitions is correct. Answer with option
choice such as 'A', 'B', 'E', etc.

               #Input:

               {
                 "arabic_sentence": "كيف ساهمت السياسة الأميركية المستندة إلى رؤية سعودية في سوء فهم أمريكا لليمن
وتعثرها هناك ؟ \" ليس لدى الولايات المتحدة سياسة تجاه اليمن \" ولكن هذا ليس مفاجئا : الولايات المتحدة لم يكن لديها سياسة
خاصة تجاه اليمن قط.",
                 "arabic_word": "ساهمت",
                 "multiple_choice_definition_options": [
                   {
                     "A": {
                       "definition": "/فَسَاهَمَ فَكَانَ مِنَ الْمُدْحَضِينَ<\ الصافات"\ ساهَمَه:- قارعه وغالبه وباراه في الفوز بالسّهام
141 \"."
                     }
                   },
                   {
                     "B": {
                       "definition": "ساهَمَه:- قاسمه وأخذ سهمًا أي نصيبًا معه"
                     }
                   },
                   {
                     "C": {
                       "definition": "ساهم في حلّ مشاكل صديقه - ساهم في"\ ساهم في الأمر:- أسهم فيه، شارك فيه، عاون، ساعد
لتنظيم مظاهرة احتجاج - ساهمت الحكومة بمبلغ من المال للمشروع \"."
                     }
                   }
                 ]
               }

               [RESPONSE FORMAT]
               Generate response as valid JSON with following schema. JSON must parse accurately without any formatting
issues like missing commas, etc.
               {
                   "correct_option": <correct_option>,
                   "correct_option_definition": <correct_option_definition>
               }

               Here is an example response with valid JSON format:
               {
                   "correct_option": "A",
                   "correct_option_definition": "كيف زيد؟ : كيف"
               }

               [JSON RESPONSE]
  }
]
```

Figure 2: Prompt used for WSD task submission

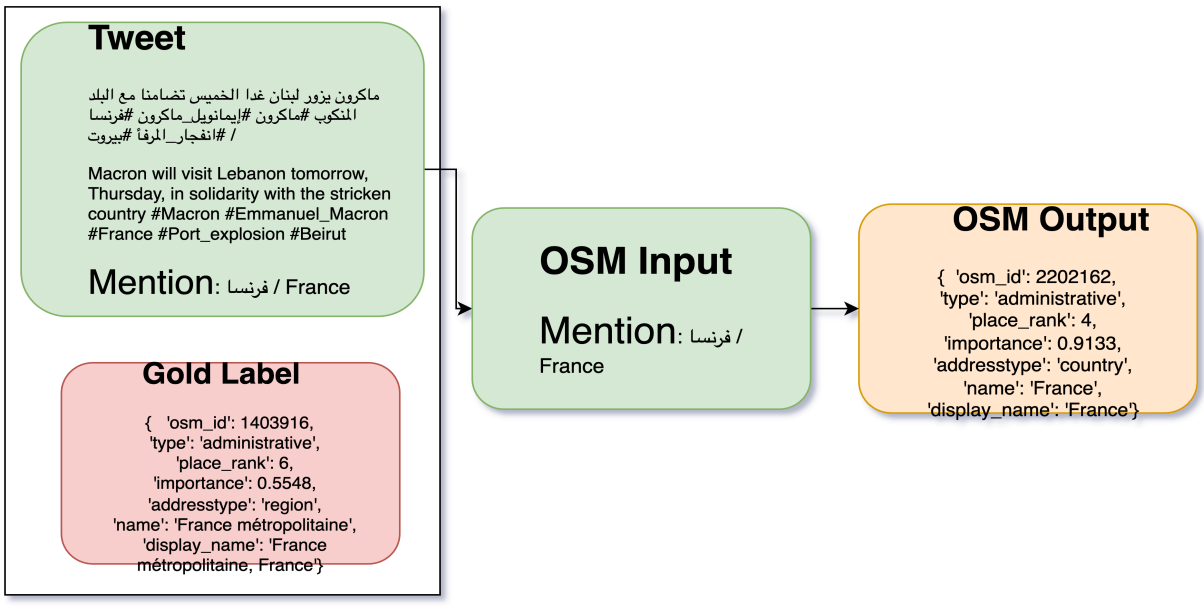

Figure 3: Input Output to OSM API