XNLP: An Interactive Demonstration System for Universal Structured NLP

Hao Fei[1], Meishan Zhang[2], Min Zhang[2], Tat-Seng Chua[1]

[1] NExT++ Research Center, National University of Singapore

[2] Harbin Institute of Technology (Shenzhen), China

{haofei37,dcscts}@nus.edu.sg, {zhangmeishan,zhangmin2021}@hit.edu.cn

Abstract

Structured Natural Language Processing (XNLP) is an important subset of NLP that entails understanding the underlying semantic or syntactic structure of texts, which serves as a foundational component for many downstream applications. Despite certain recent efforts to explore universal solutions for specific categories of XNLP tasks, a comprehensive and effective approach for unifying all XNLP tasks long remains underdeveloped. Meanwhile, while XNLP demonstration systems are vital for researchers exploring various XNLP tasks, existing platforms can be limited to, e.g., supporting few XNLP tasks, lacking interactivity and universalness. To this end, we propose an advanced XNLP demonstration system, where we leverage LLM to achieve universal XNLP, *with one model for all* with high generalizability. Overall, our system advances in multiple aspects, including universal XNLP modeling, high performance, interpretability, scalability, and interactivity, offering a unified platform for exploring diverse XNLP tasks in the community.[1]

1 Introduction

XNLP has been referred to as a special form of NLP tasks that involves holistically analyzing and interpreting the underlying semantic or syntactic structure within a text, such as Syntactic Dependency Parsing (Nivre, 2003), Information Extraction (Wang and Cohen, 2015), Coreference Resolution (Lee et al., 2017), and Opinion Extraction (Pontiki et al., 2016), etc. Figure 1 (upper part) illustrates some representative XNLP tasks under different categories. XNLP has been infrastructural for a wide range of downstream NLP applications, such as Knowledge Graph Construction (Bosselut et al., 2019), Empathetic Dialogue (Rashkin et al., 2019), and more newly-emerging applications and techniques (Tang et al., 2020).
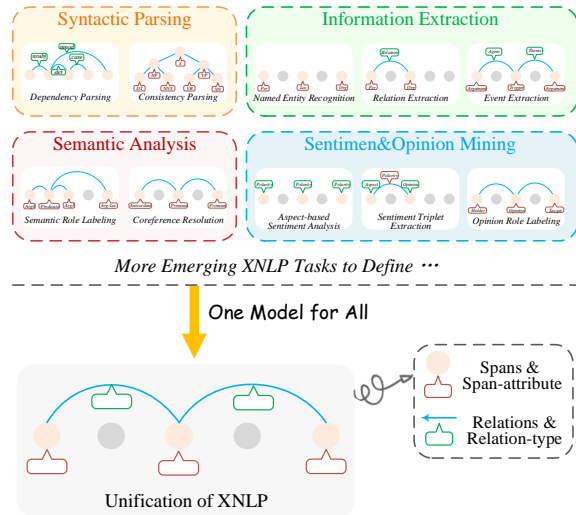

Figure 1: Illustration of the Structured NLP (XNLP) tasks, and the unification of XNLP by decomposing into the predictions of spans and relations.

As the key common characteristics, all the XNLP tasks have revolved around predicting two key elements from input: **1) textual spans** and **2) relations between spans** (Fei et al., 2022b), as depicted in the Figure 1 (lower part). Traditional efforts for XNLP have treated each task independently, which has led to limited utilization of shared features among XNLP tasks (He et al., 2019), and sub-optimal model generalization across different datasets (Chauhan et al., 2020), such as cross-language and cross-domain scenarios. In this paper, we emphasize the importance of Model Unification as a crucial topic in NLP. By unifying various NLP tasks under the XNLP framework, we can take advantage of the shared characteristics among tasks, leading to better model generalization and improved performance in realistic scenarios of product deployment.

Despite recent certain efforts in exploring universal solutions for some categories of XNLP tasks, such as Unified Sentiment Analysis (Chen and

---

[1]XNLP is online: https://xnlp.haofei.vip

Video demonstration at https://youtu.be/bOc-9HELEVw

Qian, 2020; Fei et al., 2022a), and Universal Information Extraction (UIE; Lu et al., 2022; Fei et al., 2022b), a comprehensive and effective approach for unifying all XNLP tasks has not been fully established. Fortunately, Large Language Models (LLMs) (Vaswani et al., 2017; Raffel et al., 2020) present a potential solution for unification across all XNLP tasks. There is a recent development in the form of LLMs, e.g., ChatGPT (Ouyang et al., 2022), LLaMA (Touvron et al., 2023) and Vicuna (Peng et al., 2023), that have shown promising advancements in NLP and other fields. LLMs, with sufficient sizes of model and data, have demonstrated impressive generalization capabilities, well supporting the idea of *"One model for all"* (OpenAI, 2023). In this work, we propose taking advantage of LLMs to achieve universal XNLP, addressing the lack of a well-defined and holistic approach.

On the other hand, demonstration systems play a crucial role for researchers (especially beginners) exploring various XNLP tasks, providing a platform to analyze and understand the functionalities of different NLP components and their applications. While there are existing widely-used XNLP demo systems, such as *CoreNLP*[2], *AllenNLP*[3], we have observed several key issues with them: 1) limited to only a few specific tasks; 2) lacking interactive and extensible features, making it challenging to support dynamic growth in new XNLP tasks; 3) not universal systems, requiring separate models for each task, which can lead to increased overhead. To address these limitations, this work aims to build an advanced platform that provides superior XNLP demonstrations and benefits the broader NLP community.

In summary, our system advances in the following aspects.

1) **Universalness**
   - Our XNLP system takes the existing open-source LLMs as the backbone engine with excellent generalization capabilities, enabling unified prediction of various XNLP tasks, leading to a streamlined and cohesive XNLP ecosystem.
   - The LLM-based system supports end-to-end predictions for complex structured tasks, regardless of whether the spans are nested or discontinuous, making it versatile and adaptable to different linguistic structures.

2) **High Performance**
   - Our system is capable of few-shot or weakly-supervised learning. Having undergone extensive pre-training, LLMs do not require in-domain fine-tuning on specific task data.
   - Our system supports open-label and vocabulary predictions, utilizing LLM's generalization capabilities to discover new labels and vocabs with superior out-of-domain generalization.
   - Our approach naturally lends itself to cross-lingual, code-switching, and cross-domain settings.

3) **Scalability&Interpretability&Interactivity**
   - The system allows dynamic addition and definition of new tasks, requiring users only to provide demonstrations for the new tasks.
   - Predictions generated by our system are interpretable, as LLMs are able to provide rationales for their decisions, explaining why a specific result is produced.
   - The system enables user-machine interaction, empowering users to provide feedback, thereby allowing the system to refine its predictions based on user input.

2 Related Work

2.1 Structured NLP

Over the last few decades, XNLP has garnered significant research attention, with several works addressing specific aspects of XNLP tasks, spanning from linguistic/syntactic parsing (Kitaev and Klein, 2018), to information extraction (Mikheev et al., 1999), to semantic analysis (He et al., 2017) and to sentiment analysis & opinion mining (Wu et al., 2021). Prior studies and efforts have been paid and achieved notable developments for each of the XNLP tasks, such as Syntactic Dependency Parsing (Nivre, 2003), Information Extraction (Wang and Cohen, 2015), Coreference Resolution (Lee et al., 2017), and Opinion Extraction (Pontiki et al., 2016), etc. Different XNLP tasks may have different specific task definitions, while prediction formats of all the XNLP tasks can be reduced to the same prototype: the term extraction and relation detection (Lu et al., 2022; Fei et al., 2022b).

Demonstration for XNLP. The development of demonstration platforms has been crucial for educational and academic purposes, e.g., aiding researchers to explore various tasks and gaining hands-on experiences. Existing widely-employed

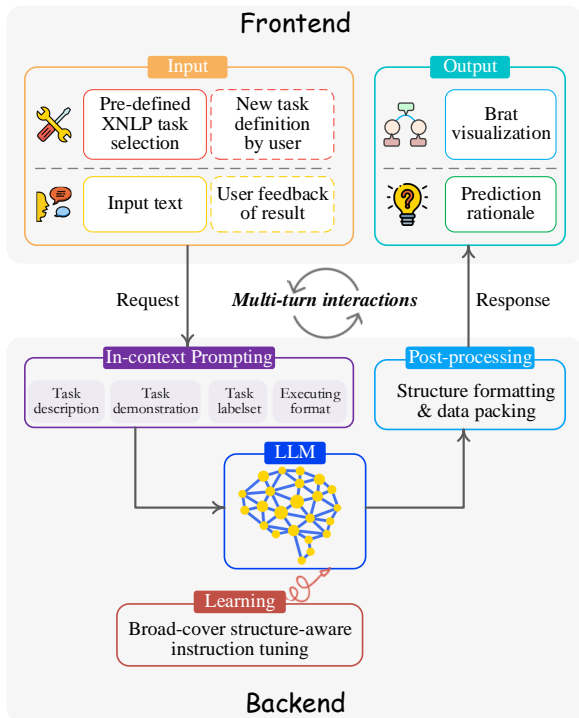---

[2]http://corenlp.run/
[3]https://demo.allennlp.org/

Figure 2: The overall architecture of our XNLP system includes the **frontend** module and the **backend** module.

open demo systems for XNLP include *CoreNLP*[4], *AllenNLP*[5] and *Explosion.ai*[6] etc. While offering user-friendly web interface for users to access a set of XNLP functionalities, there remain certain limitations, such as lacking flexibility for incorporating new tasks, non-universalness for model and cross-domain generalization.

2.2 Model Unification

There have been notable efforts to explore universal modeling for a type of NLP tasks (Chen and Qian, 2020; Fei et al., 2022a; Lu et al., 2022; Fei et al., 2022b), showcasing the benefit and potential of model unification, e.g., better leverage of shared characteristics and knowledge across tasks, simplified model maintenance, and enhanced system efficiency. However, a comprehensive and effective approach for unifying all XNLP tasks remains under-investigated. In this work, by capitalizing on LLM's robustness and broad applicability, we aim to pave the way for an advanced unified framework capable of handling diverse XNLP tasks effectively.

3 System Design

Architecture overview. We design our XNLP demo system into a web interface form. Built based

on the Django[7] framework, XNLP divides the functions into the **frontend** module and the **backend** module. As shown in Figure 2, the frontend takes user inputs and displays the visualization of outputs, and the backend provides task prediction services with LLM as its core engine, based on the in-context learning paradigm. Also, it is possible for multi-turn interactions between frontend and backend.

3.1 Backend

Backbone LLM. Among a list of open-source LLMs, we consider the Vicuna-13B[8] as our backbone. Trained by fine-tuning LLaMA (Touvron et al., 2023) on user-shared conversations collected from ShareGPT, Vicuna has achieved more than 90% of OpenAI ChatGPT's (Peng et al., 2023) quality in user preference tests.

In-context learning. To elicit LLM to induce task predictions, we build in-context prompts. We note that to ensure the support of universal XNLP for any potential tasks and inputs, the prompt template should cover rich and informative information from the user end. Thus, we design the prompt by mainly covering the **task name**, **task description**, **task demonstration**, **task label set**, **executing format**, **input text**, **language** and **domain**.

```
{Task-desc}
———————————————————————
For example, {Task-demo}
———————————————————————
Note the task output format should
be with this: {Exe-format}
———————————————————————
And generally, the desired predicted
labels should be within the following
given label set: {Task-label}
———————————————————————
Now, given a new test input:
"{Input-text}", please do the task
of {Task-name}.
———————————————————————
Note the input is with {Language}
language, and the text is from the
{Domain} domain.
———————————————————————
Please predict all possible results
strictly following the exact given
format, without any other output of
explanations.
```

Fed with the above prompt, the LLM is ex-

pected to output prediction in the provided format (**executing format**), with which, the post-process program further parses and polishes the structure result, and packs data to return to the frontend.

Broad-cover structure-aware instruction tuning. While LLM's outputs are sequential, XNLP tasks are highly structured. Thus, we expect the LLM to generate strictly structural results conditioned on sequence inputs. We consider further tuning the LLM with a *broad-cover structure-aware instruction tuning* mechanism. Instruction tuning is an emergent paradigm of LLM fine-tuning wherein natural language instructions are leveraged with LLM to induce the desired result more accurately. We write the XNLP predictions (outputs) for any input prompt by formatting the predictions into task-agnostic (i.e., task broad-covering) well-formed structure representations as in Fei et al. (2022b).

Structure formatting. As aforementioned, all the XNLP can be unified by predicting two key elements: the term extraction (with the span attribute) and relation detection (with the relation type), as illustrated in Figure 1. To unify all XNLP tasks, we follow Fei et al. (2022b) and design a structure formatter, where all the XNLP task outputs share the same structural representations. As shown in Figure 4, under the structure formatted, all XNLP tasks have been divided into the span extraction, pair extraction and hyper-pair extraction.

3.2 Frontend

As illustrated in Figure 2 (upper part), the frontend of XNLP receives inputs of 1) texts or user feedback or 2) task metadata (pre-defined or user-defined), and exhibits outputs from LLM. Following we mainly describe the key features of the frontend module as listed below.

Pre-defined XNLP tasks. To facilitate the user operation, we pre-defined total 22 XNLP tasks, covering four frequent categories, including Syntax Parsing, Information Extraction, Semantic Analysis and Sentiment/Opinion Mining.

New task definition. As there are rapidly-emergent XNLP tasks in the NLP community, it is impossible to cover it all in the pre-definition. We thus allow users to define their own XNLP tasks. This can be easily accomplished in our system without much effort, as the LLM has exceptional zero-shot performance and understanding ability. We require from the user only the *task name*, *task description*, *task demonstration*, *task label set*, *executing format*.

XNLP structure visualization. The key role of XNLP system is the visualization of the task output structure. We employ the open-source *brat* system[9] to realize this. *brat* has been shown very popular and effective in rendering structured data, with pretty visualization and stable functions.

Rationale for explainable task prediction. Besides the visualization of direct task results, we also display the rationale for each prediction, allowing *seeing what and knowing why*. This is especially meaningful for the beginners of the researchers for XNLP tasks. To enable this, we just ask LLM *"How and why do you make your decision?"* after each task prediction.

Enhancing prediction with user interaction. To take full advantage of the LLM, we further allow users to interact with our system by providing any feedbacks, so that users can revise the task predictions whenever they feel the results are not incorrect or coincident with their minds. To reach this, we also add another round of query to LLM, by asking *"The above prediction is not all right, because Feedback. Please do the task again by carefully taking the feedback here"*.

4 System Walkthrough

Figure 3 gives a comprehensive walkthrough of how the system can be operated by users.

▶ **Step-1.** *users select or define a task*;
▶ **Step-2.** *users go through (for pre-defined) or fill in (for user-defined) the task prompt*;
▶ **Step-3.** *users key in the text to analyze*;
▶ **Step-4.** *users submit the text & metadata and request result*;
▶ **Step-5.** *users can browse the visualization of task output*;
▶ **Step-6.** *users observe the rationale of this result*;
▶ **Step-7.** *users can further provide feedback for the system to re-generate result*;

Following we demonstrate XNLP system by walking readers through several important functions.

4.1 User-allowed Operations

Pre-defined XNLP task selection. For the first step, users should select an XNLP task template
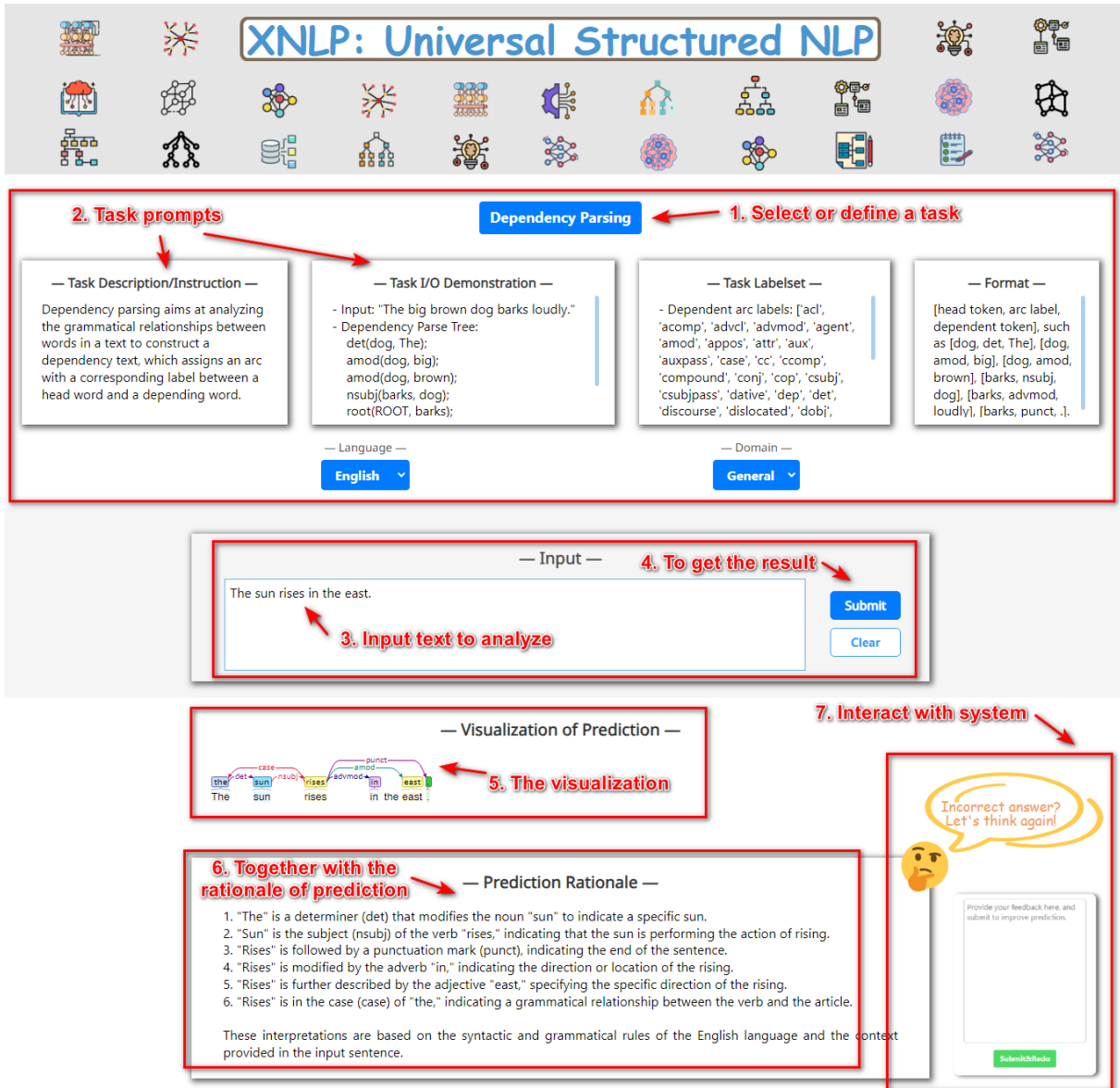
---

[9]https://brat.nlplab.org/

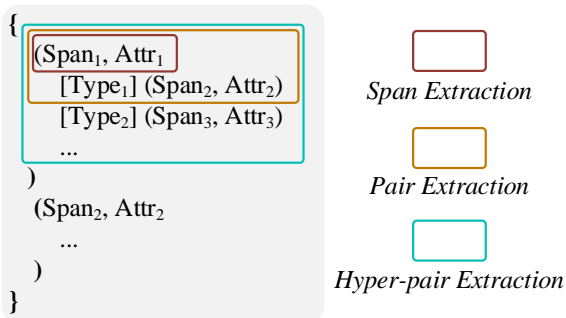Figure 3: Screenshot of the XNLP web application, where key functions are annotated.

Figure 4: Structure formatter for universal XNLP.

from the 22 system pre-defined pools. The operation is shown in Figure 7 in Appendix §A.

New task definition. Or, user can define their own tasks. As shown in Figure 8 in Appendix §B, users should decide the task name, and fill in the task metadata (task prompt) as shown in Figure 3, and also select a pre-defined task with which the new task shares most similarity.

Language and domain notifications. To enable more accurate predictions, it is better to explicitly notify LLM what language and domain the input has. Figure 10 in Appendix §D and Figure 11 in Appendix §E illustrate the operations, respectively.

Improving/Revising Prediction with User Feedback Figure 9 in Appendix §C showcase the operation for the multi-turn user interaction.

4.2 Task Visualization

Here we showcase the XNLP task visualizations of real examples via our system. Figure 5 renders the outputs for the four task clusters, with each
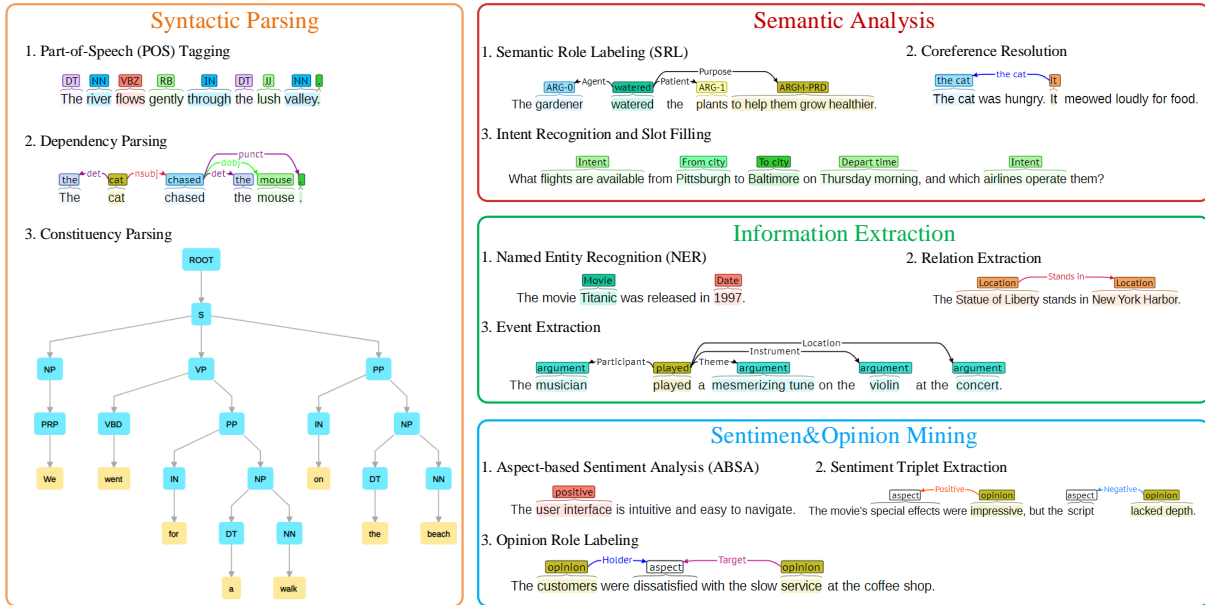
Figure 5: Screenshots of the visualizations of 12 representative XNLP tasks. Best viewing with zooming in.

showing three representative task results, such as:

Syntax parsing, including Part-of-Speech (POS) Tagging, Dependency Parsing and Constituency Parsing.

Semantic analysis, including Semantic Role Labeling (SRL), Coreference Resolution, and Intent Recognition and Slot Filling.

Information extraction, including Named Entity Recognition (NER), Relation Extraction, and Event Extraction.

Sentiment/opinion mining, including Aspect-based Sentiment Analysis (ABSA), Sentiment Triplet Extraction and Opinion Role Labeling.

We can observe from the visualizations that, 1) the structure visualizations are pretty, owing to the use of the brat system; 2) the results of tasks are correct, for which we give the credit to the integration of LLM, and also the *broad-cover structure-aware instruction tuning* mechanism.

5   Performance Evaluation

To quantitatively verify the performance of the backbone LLM on XNLP tasks, we now perform evaluations. We compare the Vicuna (13B) with the ChatGPT over 100 randomly selected test instances of 6 XNLP tasks. The experiments are based on one-shot in-context learning, i.e., with one demonstration as input. Figure 6 shows the comparisons. We see Vicuna has a slightly lower performance than ChatGPT, while Vicuna after broad-cover structure-aware instruction tuning (Vi-
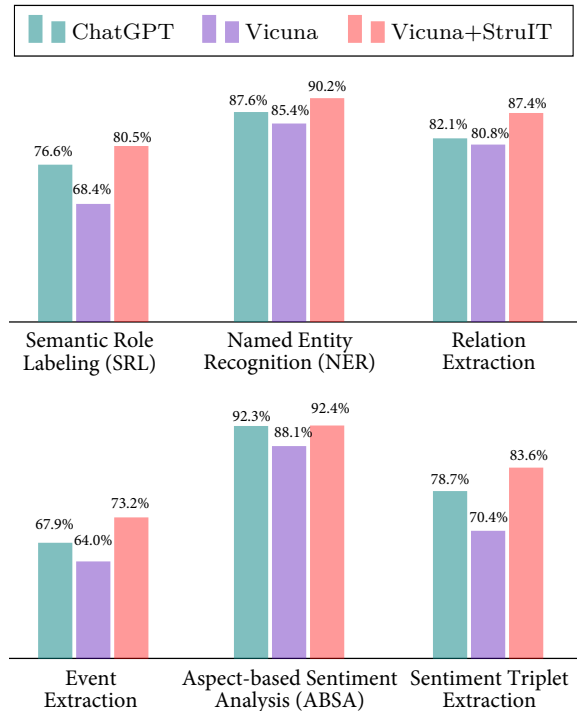

Figure 6: Comparisons (end-to-end prediction, in accuracy) between ChatGPT and Vicuna on XNLP tasks.

cuna+StruIT) shows results even much better than ChatGPT, with smaller model size (13B vs. 175B).

6   Conclusion

We present XNLP, an advanced online demonstration system for interaction and visualization of XNLP tasks. XNLP, built upon LLM, effectively models all the XNLP tasks universally, achieving *one model for all* in zero-shot or weak supervision. XNLP not only renders the output structures with

delicate visualizations, but also provides rationales for interpretable predictions. Also, XNLP allows the users to define newly emergent XNLP tasks; and enables users to dynamically revise the output with multi-turn interactions. Our XNLP contributes to the community by paving the way for a unified, scalable, and interactive demonstration platform.

Limitations

The focus of this paper was introducing an open online web application (demonstration system) to make the interaction of XNLP tasks available to as many practitioners as possible, but there are a couple of limitations in the system and the model we proposed. First, our system is based on the web service form, with the LLM running at the backend deployed at the online server, where sometimes when the Internet traffic is bad, the user may wait for too long to get the response. Second, as the LLM essentially generates sequential texts of any inputs, there are chances that the output texts include problematic structured formatter (i.e., structural representations, cf. Figure 4). With ill-formed structural representations, it is problematic to parse them into correct data used for rendering into brat visualization, i.e., causing failure prediction. Third, as one of the nature characteristics, LLM may sometimes generate false output, or do not obey the input instructions, which has been called the *Hallucination* phenomenon (Varshney et al., 2023). In such case, the user experience will be affected. Lastly, the current version of the system is still at a basic stage, and there are functionalities at the user interface level that need further polishing and improvement in subsequent updates.

Ethics Statement

Our XNLP system uses the LLM as backbone. While the Vacuna model is fine-tuned on the pre-trained LLaMA model, which is known to contain some toxic contents (Schick et al., 2021), an internal check does not reveal any toxic generation. However, there is a potential risk that the Vacuna could generate toxic text for users due to the underlying black-box LLM.

References

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th ACL*, pages 4762–4779.

Dushyant Singh Chauhan, Dhanush S R, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Sentiment and emotion help sarcasm? a multi-task learning framework for multi-modal sarcasm, sentiment and emotion analysis. In *Proceedings of the 58th ACL*, pages 4351–4360.

Zhuang Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th ACL*, pages 3685–3694.

Hao Fei, Fei Li, Chenliang Li, Shengqiong Wu, Jingye Li, and Donghong Ji. 2022a. Inheriting the wisdom of predecessors: A multiplex cascade framework for unified aspect-based sentiment analysis. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, pages 4121–4128.

Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2022b. Lasuie: Unifying information extraction with latent adaptive structure-aware generative language model. In *Proceedings of the Advances in Neural Information Processing Systems, NeurIPS 2022*, pages 15460–15475.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th ACL (Volume 1: Long Papers)*, pages 473–483.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th ACL*, pages 504–515.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. pages 2676–2686.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 EMNLP*, pages 188–197.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th ACL (Volume 1: Long Papers)*, pages 5755–5772.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *CoRR*, abs/2304.03277.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th ACL*, pages 5370–5381.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Trans. Assoc. Comput. Linguistics*, 9:1408–1424.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In *Proceedings of the 58th ACL*, pages 6578–6588.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *CoRR*, abs/2307.03987.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5998–6008.

William Yang Wang and William W. Cohen. 2015. Joint information extraction and reasoning: A scalable statistical relational learning approach. In *Proceedings of the 53rd ACL and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 355–364.

Shengqiong Wu, Hao Fei, Yafeng Ren, Donghong Ji, and Jingye Li. 2021. Learn from syntax: Improving pair-wise aspect and opinion terms extraction with rich syntactic knowledge. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 3957–3963.

A   Selection of Pre-defined XNLP Tasks
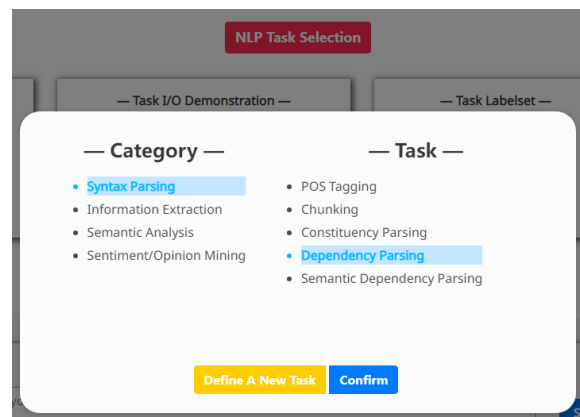
See Figure 7.

Figure 7: Screenshot of the selection panel of pre-defined XNLP tasks.

B   New XNLP Task Definition

See Figure 8.
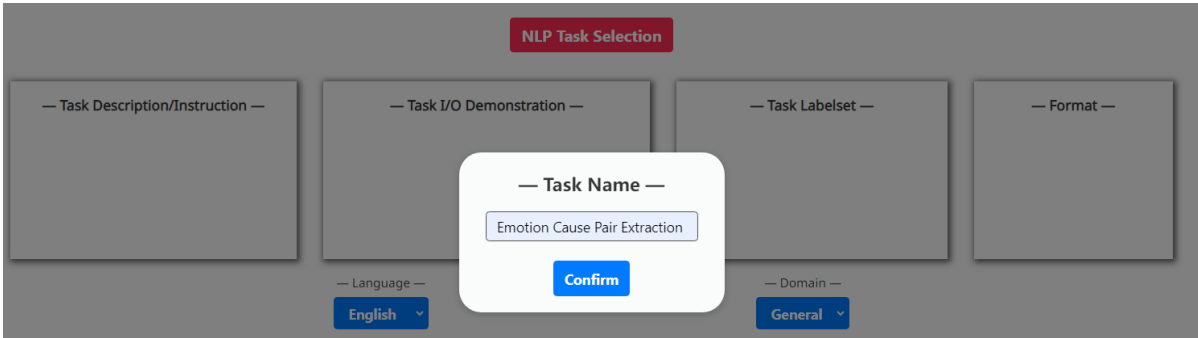
C   Multi-turn Interactions with User Feedback

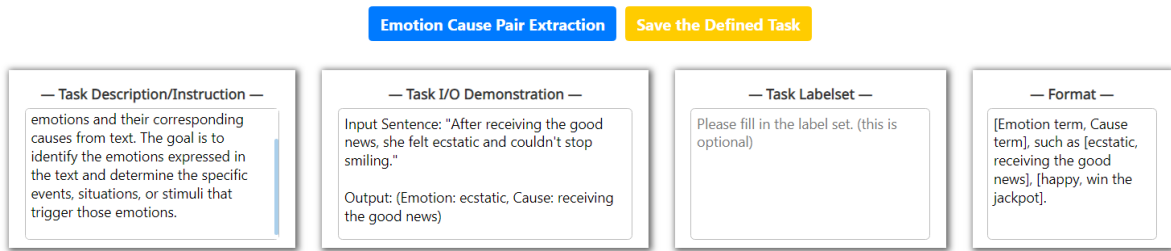See Figure 9.

D   Text in Different Language

See Figure 10.
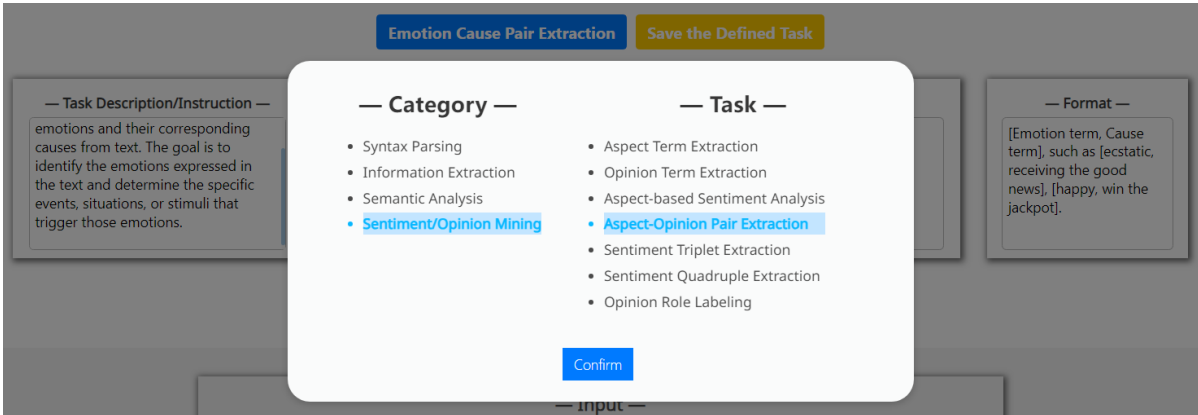
E   Text in Different Domain
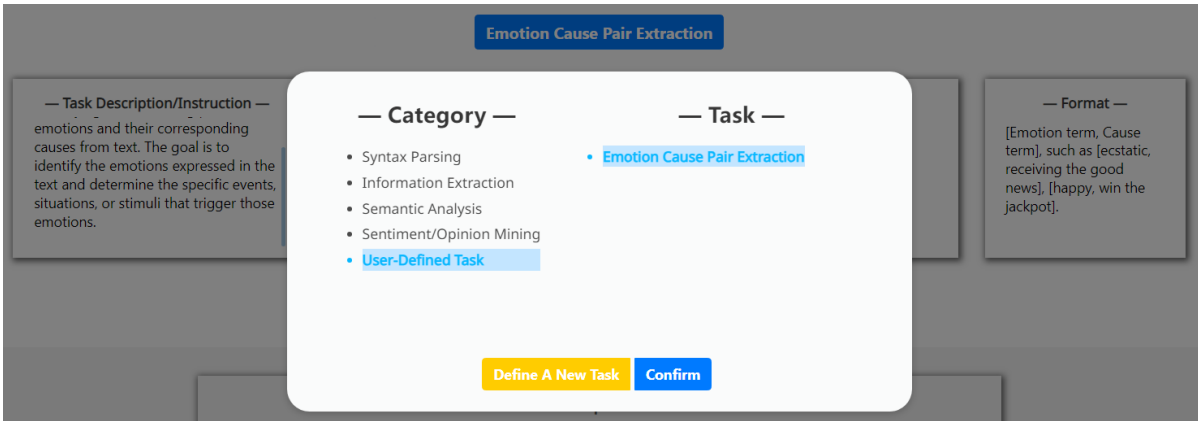
See Figure 11.

(a) Step-1, name the task.

(b) Step-2, fill in the task metadata.

(c) Step-3, select an executing format with a similar task,

(d) Step-4, confirm to define.

Figure 8: Screenshot of defining new XNLP task by the user.

— Visualization of Prediction —

the children ←—the children, they, themselves→ they ←—the children, they, themselves→ themselves

The children were building sandcastles. They were enjoying   themselves. Their parents took pictures of these kids.

you lost the last "them" (end of the sentence) of the chain, should add "them" into the chain, i.e,."the children"-"they"-"themeselves"-"these kids".

Submit&Redo

— Prediction Rationale —

The rationale for producing the results is as follows:

1. "The children" refers to a group of individuals being mentioned for the first time in the sentence. This is the antecedent for the following mentions.

2. "They" is a pronoun that refers to the group of children mentioned earlier. It is used to maintain coherence and avoid repetition.

3. "Themselves" is another pronoun that refers back to the group of children. It emphasizes that the children are enjoying their own company.

4. The common referent for all the coreferred terms is "The children". This is the entity that all the mentions are linked to.

(a) Before user feedback.

— Visualization of Prediction —

the children ←the children, they, themselves, these kids→ they ←the children, they, themselves, these kids→ themselves ←the children, they, themselves, these kids→ these kids

The children were building sandcastles.   They were enjoying   themselves. Their parents took pictures of   these kids.

Incorrect answer? Let's think again!

— Prediction Rationale —

The rationale for producing the updated results is as follows:

1. "The children" refers to a group of individuals introduced in the sentence. It serves as the antecedent for subsequent mentions.

2. "They" is a pronoun that refers back to the group of children mentioned earlier. It is used to avoid repetition and maintain coherence in the text.

3. "Themselves" is another pronoun that refers to the group of children. It emphasizes that the children are enjoying their own company.

4. "These kids" is a noun phrase that also refers to the same group of children. It provides an alternative way of referring to the children mentioned earlier.

The coreference chain consists of the coreferred terms "The children," "They," "themselves," and "these kids." All these terms are in the exact original form copied from the input text.

The common referent for all the coreferred terms is "The children." This is the entity that all the mentions are linked to, establishing coherence and understanding the relationships between different mentions of the same entity.

(b) After user feedback.

Figure 9: System output before user feedback and after user feedback.

Figure 10: Input text in different languages (Chinese).

Relation Extraction

— Task Description/Instruction —

Relation Extraction involves identifying semantic relationships between two entities mentioned in a text.

— Task I/O Demonstration —

- Input Text: "Apple Inc. was founded by Steve Jobs and Steve Wozniak in 1976."
- Output [subject (category), relation, object (category)]:
    [Apple Inc. (Organization), founder, Steve Jobs (Person)],
    [Apple Inc. (Organization), founder, Steve

— Task Labelset —

- Entity labels: ['person', 'location', 'organization'], where the labels can be flexible depending on the input;
- Relation labels: ['founder', 'born_in', 'located_in', 'CEO', 'works_at', 'parent_company', 'founded_year', 'headquarters_in',

— Format —

[entity-subject (category), relation, entity-object (category)], such as [Barack Obama (person), President, the United States (GPE)], [Barack Obama (person),

— Language —
English ⌄

— Domain —
Finance ⌄

— Input —

The company's revenue increased by 20% in the last quarter, leading to a rise in its stock price.

Submit

Clear

— Visualization of Prediction —

Organization ---Revenue increased--- None
The company's revenue increased by 20% in the last quarter, leading to a rise in its stock price.

Incorrect answer? Let's think again!

— Prediction Rationale —

1. The input text mentions "The company's revenue increased by 20% in the last quarter." This indicates that there is a relationship between the company and its revenue.

2. The verb "increased" suggests a change or action, indicating a possible relation between the company and the increase in revenue.

3. The percentage "20%" is mentioned as the extent of the increase in revenue. While it doesn't fit into any pre-defined entity category, it provides additional information related to the revenue increase.

4. The phrase "leading to a rise in its stock price" indicates a causal relationship between the revenue increase and the change in stock price. However, the specific relationship between the company, revenue increase, and stock price is not explicitly mentioned in the input text, so it is not possible to determine a specific relation label.

Based on these reasons, the prediction for the relation extraction is [company (organization), revenue increased, 20% (none)].

Figure 11: Input text in different domains (financial domain).