Enhanced Classification of Complaints in a gaining information setting

Anonymous ACL submission

Abstract

Often systems like digitized grievance readressals in a setting of continuous chat between user and bot requires to rightly classify the user information. We propose <name>, a framework which addresses a two-fold problem of streamlining user responses to reveal further relevant information; and to enhance classification of total information till a particular point to the right label. The label classification is enhanced by joint learning of supporting features. We have demonstrated a realistic scenario of user complaint redirection to the right ministry in a narrowing information setting, and the superior performance of <name> over other classifiers.

1 Introduction

Many systems today are shifting from traditional classification to Large Language Model (LLM) based methods. Traditional techniques, like dropdown selection from a given set of labels, requires the user to have a pre-knowledge of the topic and setting. And often, such setup is single layered, i.e., no further supervisions are done if the information is rightly labelled or not.

On the other hand, the use of LLMs as a chat based approach allows systematic flow of information as a user-bot conversation. We propose <name>, a framework which allows continuous chat-based conversations with a streamline revelation of information, along with classification of the user information so far to the right label. We have demonstrated a Indian grievance re-addressing setting with the objective of iterative prompting to fill-in missing information along with routing the complaint to it's right ministry. In a typical setting, when a user types in complaint, the user may not be knowledgeable enough to select the right ministry; also the information provided as free-flow text may lack essential elements required. We group <name> as a two-fold approach, (1) iterative chat bot responses and prompting for predicted missing information; and (2) classification of collected complaint information at every user-bot level to a probable ministry. We have used the open-source, Llama-chat-7B model as the base LLM owing to it's ability to respond in a conversational fashion. The ministries follow a hierarchical structure consisting of parent ministry, intermediate ministries, and leaf-level ministries, or leaf nodes in a structured tree. The latter is a subset of the former ministry and each parent ministry can have a wide number of children. The hierarchy is followed by a semantic relationship, to say, the context of child nodes are a narrowed and specialized versions of the parent ministry. To quote an example, the parent ministry is "healthcare", of which a branch is "hospital", followed by "lack of beds" which is the leaf-node. <name> uses Retrieval Augmented Generation (RAG) for the generation of iterative prompts. We have generated synthetic directed chats between users and bot abiding to the 'thought-flow' of hierarchy of the ministries over which the model retrieves relevant information as the next prompt. This mimics a scenario where actual training data for a user-bot interaction is available. Next, at the level of each user response, we collect the conversation information including the current context and the previous history. With the use of SPEAR, the classification to the right ministry (predicted till that point) is leveraged by jointly learning classification from multiple labelling functions over the feature model used. A labelling function is a set of condition(s) that needs to be full filled for it to classify the input as a particular label.

In this work, we have demonstrated the ability of the <name> framework to stand as a chat bot which can iteratively prompt to gain relevant information from the user to complete a task (to complete a complaint in our work). This information is classified to the relevant label by jointly learning from the original feature model and its supporting labelling

functions.

1.1 Understanding <name> with example

1.2 Our Contribution

We propose a framework <name> in the chat bot based complaint re-direction setting, which has a dual functionality of prompting for relevant and specific information to narrow down the right prediction, and leveraged classification by jointly learning the labeling functions with the feature model. We have used a novel technique of RAG based prompting, wherein synthetic conversation documents are retrieved as a knowledge-base for the next prompt by bot. These prompts are able to fill complete the remaining required information with a successful complaint closure. Each complaint information is classified to its labels (ministries here) using joint learning by leveraging the feature model prediction and the supporting labeling functions.

2 Related Work

3 Dataset

The available data consists of whole hierarchy tree of ministry structure which includes parent, intermediate, and leaf-level ministries. There are a total of 24 parent ministries which serve as the most broad level contextual division. A total of <number> leaf-level ministries are present in the structure. We have in cooperated the hierarchy structure and mandatory fields to generate synthetic real-world complaints corresponding to each leaf-level ministry. The further sections explains the same in detail.

4 Our Approach

4.1 Problem Description

A complaint lodging systems based on chat bot working has two components: bot response to prompt for further collection of relevant information, and classification of the collected complaint information to a particular leaf-level ministry, which are our labels. We assume that the user doesn't have the required pre-knowledge for the right ministry and its required fields, and therefore needs to be prompted for it.

4.2 Data Synthesis and Prompting

4.3 Classification and Labeling functions

5 Results

These instructions are for authors submitting papers to ACL 2023 using LaTeX. They are not self-contained. All authors must follow the general instructions for *ACL proceedings,[1] as well as guidelines set forth in the ACL 2023 call for papers.[2] This document contains additional instructions for the LaTeX style files. The templates include the LaTeX source of this document (`acl2023.tex`), the LaTeX style file used to format it (`acl2023.sty`), an ACL bibliography style (`acl_natbib.bst`), an example bibliography (`custom.bib`), and the bibliography for the ACL Anthology (`anthology.bib`).

6 Engines

To produce a PDF file, pdfLaTeX is strongly recommended (over original LaTeX plus dvips+ps2pdf or dvipdf). XeLaTeX also produces PDF files, and is especially suitable for text in non-Latin scripts.

The first line of the file must be

```
\documentclass[11pt]{article}
```

To load the style file in the review version:

```
\usepackage[review]{ACL2023}
```

For the final version, omit the `review` option:

```
\usepackage{ACL2023}
```

To use Times Roman, put the following in the preamble:

```
\usepackage{times}
```

(Alternatives like txfonts or newtx are also acceptable.) Please see the LaTeX source of this document for comments on other packages that may be useful. Set the title and author using `\title` and `\author`. Within the author list, format multiple authors using `\and` and `\And` and `\AND`; please see the LaTeX source for examples. By default, the box containing the title and author names is set to the minimum of 5 cm. If you need more space, include the following in the preamble:

```
\setlength\titlebox{<dim>}
```

where <dim> is replaced with a length. Do not set this length smaller than 5 cm.

[1] http://acl-org.github.io/ACLPUB/formatting.html

[2] https://2023.aclweb.org/calls/main_conference/

7 Document Body

7.1 Footnotes

Footnotes are inserted with the `\footnote` command.[3]

7.2 Tables and figures

See Table **??** for an example of a table and its caption. **Do not override the default caption sizes.**

7.3 Hyperlinks

Users of older versions of LaTeX may encounter the following error during compilation:

```
\pdfendlink ended up in different
nesting level than \pdfstartlink.
```

This happens when pdfLaTeX is used and a citation splits across a page boundary. The best way to fix this is to upgrade LaTeX to 2018-12-01 or later.

7.4 Citations

Table **??** shows the syntax supported by the style files. We encourage you to use the natbib styles. You can use the command `\citet` (cite in text) to get "author (year)" citations, like this citation to a paper by Gusfield (1997). You can use the command `\citep` (cite in parentheses) to get "(author, year)" citations (Gusfield, 1997). You can use the command `\citealp` (alternative cite without parentheses) to get "author, year" citations, which is useful for using citations within parentheses (e.g. Gusfield, 1997).

7.5 References

The LaTeX and BibTeX style files provided roughly follow the American Psychological Association format. If your own bib file is named `custom.bib`, then placing the following before any appendices in your LaTeX file will generate the references section for you:

```
\bibliographystyle{acl_natbib}
\bibliography{custom}
```

You can obtain the complete ACL Anthology as a BibTeX file from https://aclweb.org/anthology/anthology.bib.gz. To include both the Anthology and your own .bib file, use the following instead of the above.

```
\bibliographystyle{acl_natbib}
\bibliography{anthology,custom}
```

Please see Section 8 for information on preparing BibTeX files.

7.6 Appendices

Use `\appendix` before any appendix section to switch the section numbering over to letters. See Appendix A for an example.

8 BibTeX Files

Unicode cannot be used in BibTeX entries, and some ways of typing special characters can disrupt BibTeX's alphabetization. The recommended way of typing special characters is shown in Table **??**.

Please ensure that BibTeX records contain DOIs or URLs when possible, and for all the ACL materials that you reference. Use the `doi` field for DOIs and the `url` field for URLs. If a BibTeX entry has a URL or DOI field, the paper title in the references section will appear as a hyperlink to the paper, using the hyperref LaTeX package.

Limitations

ACL 2023 requires all submissions to have a section titled "Limitations", for discussing the limitations of the paper as a complement to the discussion of strengths in the main text. This section should occur after the conclusion, but before the references. It will not count towards the page limit. The discussion of limitations is mandatory. Papers without a limitation section will be desk-rejected without review.

While we are open to different types of limitations, just mentioning that a set of results have been shown for English only probably does not reflect what we expect. Mentioning that the method works mostly for languages with limited morphology, like English, is a much better alternative. In addition, limitations such as low scalability to long text, the requirement of large GPU resources, or other things that inspire crucial further investigation are welcome.

Ethics Statement

Scientific work published at ACL 2023 must comply with the ACL Ethics Policy.[4] We encourage all authors to include an explicit ethics statement on the broader impact of the work, or other ethical considerations after the conclusion but before the

[3] This is a footnote.

[4] https://www.aclweb.org/portal/content/acl-code-ethics

references. The ethics statement will not count toward the page limit (8 pages for long, 4 pages for short papers).

Acknowledgements

This document has been adapted by Jordan Boyd-Graber, Naoaki Okazaki, Anna Rogers from the style files used for earlier ACL, EMNLP and NAACL proceedings, including those for EACL 2023 by Isabelle Augenstein and Andreas Vlachos, EMNLP 2022 by Yue Zhang, Ryan Cotterell and Lea Frermann, ACL 2020 by Steven Bethard, Ryan Cotterell and Rui Yan, ACL 2019 by Douwe Kiela and Ivan Vulić, NAACL 2019 by Stephanie Lukin and Alla Roskovskaya, ACL 2018 by Shay Cohen, Kevin Gimpel, and Wei Lu, NAACL 2018 by Margaret Mitchell and Stephanie Lukin, BibTEX suggestions for (NA)ACL 2017/2018 from Jason Eisner, ACL 2017 by Dan Gildea and Min-Yen Kan, NAACL 2017 by Margaret Mitchell, ACL 2012 by Maggie Li and Michael White, ACL 2010 by Jing-Shin Chang and Philipp Koehn, ACL 2008 by Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, ACL 2005 by Hwee Tou Ng and Kemal Oflazer, ACL 2002 by Eugene Charniak and Dekang Lin, and earlier ACL and EACL formats written by several people, including John Chen, Henry S. Thompson and Donald Walker. Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence* and the *Conference on Computer Vision and Pattern Recognition*.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

A Example Appendix

This is a section in the appendix.