Jeong-Jin Kang
Edward J. Rothwell
Yang Hao
R. Young Chul Kim
Yong-seon Jang

Advanced and Applied Convergence Letters AACL 24

Advanced and Applied Convergence & Advanced Culture Technology

12th International Symposium, ISAAC 2024&ICACT 2024
in Conjunticon with ICKAI 2024

November 21-23, 2024, COEX, Seoul, Korea
Revised Selected Papers

IIBC The Institute of Internet, Broadcasting and Communication
IPACT The International Promotion Agency of Culture Technology
지식의 숲
FOREST OF KNOWLEDGE
VIDAK
Visual Information Design Association of Korea

Volume Editors:

Jeong-Jin Kang
Dong Seoul University, 76, Bokjeong-ro, Sujeong-gu, Seongnam, Gyeonggi, Korea
E-mail: jjkang@du.ac.kr

Edward J. Rothwell
Michigan State University, 2120 Engineering Building East Lansing, MI 48824-1226, USA
E-mail: rothwell@egr.msu.edu

Yang Hao
Queen Mary University of London, Mile End Road, London E1 4NS, UK
E-mail: y.hao@qmul.ac.uk

R. Young Chul Kim
Hongik University, 2639, Sejong-ro, Jochiwon-eup, Sejong special self-govering city, Republic of Korea
E-mail: bob@hongik.ac.kr

Yong-Seon Jang
Kunsan National University, 558, Daehang-ro, Kunsan-si, Jeollabuk-do, Republic of Korea
E-mail : marshill@hanmail.net

ISSN: 2288-6060

101 Jungdae-ro, Dongbu Sunville #610, Songpa-gu, Seoul, 05719 Korea
Tel: +82 2-407-7718 Fax: +82 2-407-7716
* Printed on acid-free paper

Please note that the papers in this proceeding book are neither reviewed by peer or professional editor nor accepted as official papers. The papers are working papers that the authors study their research recently.

CONTENTS

Automatic Textual Data Transformation for Enhancing F1-Score on Classification

Jinmo Yang[1], Janghwan Kim[2], Chaeyun Seo[3],
Deuk Young Hwang[4], Kidu Kim[5], and R. Young Chul Kim[6*]

[1,2,3]SE Lab, Hongik University, Korea,
[4]Dept. of AI & Software, Kangwon National University, Korea
[5]Telecommunications Technology Association, Korea
[6,]SE Lab, Hongik University, Korea*
[1]yjmd2222@g.hongik.ac.kr, {[2]lentoconstante, [3]chaeyun}@hongik.ac.kr,
[4]dyhwang@kangwon.ac.kr, [5]kdkim@tta.or.kr, [6,]bob@hongik.ac.kr*

Abstract

Nowadays, large language models (LLMs) are used in all domains and various tasks for fast and competent solutions as well as minor and supplementary tasks. These models have been trained on many kinds of data in large quantities, and further fine-tuned for better understanding in specific domains and tasks. There is much research on inventing and enhancing fine-tuning methods and datasets; however, there is little research on the generating datasets in various syntax for better language comprehension. Therefore, we propose an automatic textual data transformation for enhancing f1-score on classification. This mechanism augments topic classification (TC) data of 45,678 Korean news headlines from Korean Language Understanding Evaluation (KLUE) benchmark dataset. Our future work includes fine-tuning a few LLMs with the augmented sentences and conducting ablation study.

Keywords: *data augmentation, f1-Score, natural language processing, prompt engineering*

1. INTRODUCTION

Large language models (LLMs) are gaining popularity ever since the release of ChatGPT3 in 2022 [1]. LLMs are capable of answering simple questions as well as generating creative business ideas [2, 3]. This is possible because they have been trained on vast amount of textual data in semi-supervised learning and some amount in supervised and reinforcement learning [4]. And to provide most effective and competent responses in a specific language, the LLMs are fine-tuned with the specific language datasets [5]. Researchers mostly focus on using different techniques on fine tuning with the language datasets. Now, it is generally accepted that splitting compound and complex sentences would accurately summarize the meaning of these sentences and is therefore suitable for LLMs. However, constructing datasets on sentence types for the LLM to understand all syntax is mostly ignored. Therefore, we present automatic textual data transformation mechanism that transforms given Korean language passages into simple, compound, complex, and colloquial sentences. The resulting sentences are processed into datasets of different sentence types. This can be used to enhance the Korean language understanding performance of LLMs. In this paper, we focus on enhancing the f1-score on classification. Specifically, we augment a large benchmark dataset of 45,678 Korean news headlines from Korean Language Understanding Evaluation (KLUE) [6] into simple, compound, complex, and colloquial sentences, totaling 182,712 sentences. The rest of the paper is as follows. Section 2 mentions background research and related works. Section 3 presents our process. Finally, Section 4 mentions our conclusion.

J.J. Kang et al. (Eds.): ISAAC 2024, AACL 24, pp. 2-6, 2024

2. BACKGROUND RESEARCH AND RELATED WORKS

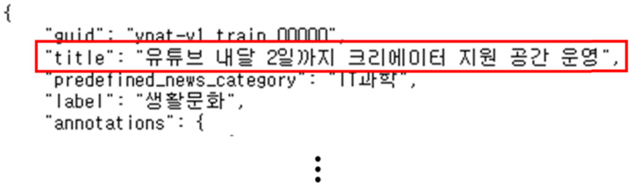
LLMs are outstanding in understanding user prompts and generating responses. There are various prompt-engineering methods for better understanding of the domains and tasks [7]. For example, prompt chaining is a technique for combining information from previous responses in the next prompts. With this, the LLM breaks down its thinking process at each prompt instead of trying to answer one whole question in a single prompt. This helps the user to debug the thinking process of the LLM. When all prompts are entered, the LLM can have a full understanding of the domain or the task, effectively producing a final response in high quality [8].

There is recent research in incorporating LLM in automation tasks. For example, Zhou et al. [9] proposed an LLM-enhanced data management system, where LLMs for different tasks were used in various places. The process is well-defined and provides effective way to process data; however, the collecting LLMs and tools for the system is too overwhelming. For a relatively simple task with a definitive goal, prompt engineering a single LLM may be enough.

Apart from prompt engineering, fine tuning gives necessary data for LLMs to update their parameters for specific tasks or domains [10]. For example, Korean language syntax data can be used in fine tuning so that the LLMs may be more competent in complex Korean grammar to better understand user input. Whereas assessment of Korean language syntax understanding [11] and preparing input in simple grammar [12] are widely practiced, there is little research on fine-tuning the LLM on diverse syntax.

3. AUTOMATIC TEXTUAL PROCESSING ALGORITHM

To prepare data for enhancing the performance (f1-score) of LLMs, topic classification (TC) dataset (ynat-v1.1_train.json) from the KLUE benchmark dataset was chosen [6]. This dataset consists of 45,678 datapoints of Korean news headlines. Figure 1 shows a sample datapoint with the title, "유튜브 내달 2일까지 크리에이터 지원 공간 운영," (which reads "YouTube runs a creator support space until the second day of the next month").



```
{
    "guid": "ynat-v1_train_00000",
    "title": "유튜브 내달 2일까지 크리에이터 지원 공간 운영",
    "predefined_news_category": "IT과학",
    "label": "생활문화",
    "annotations": {
                    ⋮
```

Figure 1. Sample KLUE TC datapoint

The automatic textual data transformation starts with the selection of the values corresponding to the "titles" keys of the key-value pair in the json data. Figure 2 shows the complete process of the automatic textual processing algorithm, divided into two flows. All steps are explained as follows:

1. Prompt-engineering flow: This step applies prompt-engineering on the LLM. This is done for the LLM to understand the characteristics of the provided data and the techniques to process the data:

a) Minimum exploratory data analysis: This sub-step is required for the user to understand the data. In transforming TC dataset, this is done with noting that the passages to be transformed are the values to the "titles" keys. The user also notes that the LLM needs knowledge on Korean language syntax for transforming given passages to sentences of different types.

b) Write / rewrite: The user writes or rewrites a prompt, with Korean language syntax information for the LLM to memorize.

c) Prompt-engineering: The prompt is processed by the LLM.

d) Prompt response: The LLM generates a response according to the input prompt. The user reads the response and then can write the next prompt or rewrite the previous one to better teach the LLM with syntax information.

e) Repeat sub-steps b)-d): This engineering flow is repeated until the LLM is prepared for data processing.

2. LLM integration: The LLM to which the user applied prompt-engineering is integrated into the transformation flow

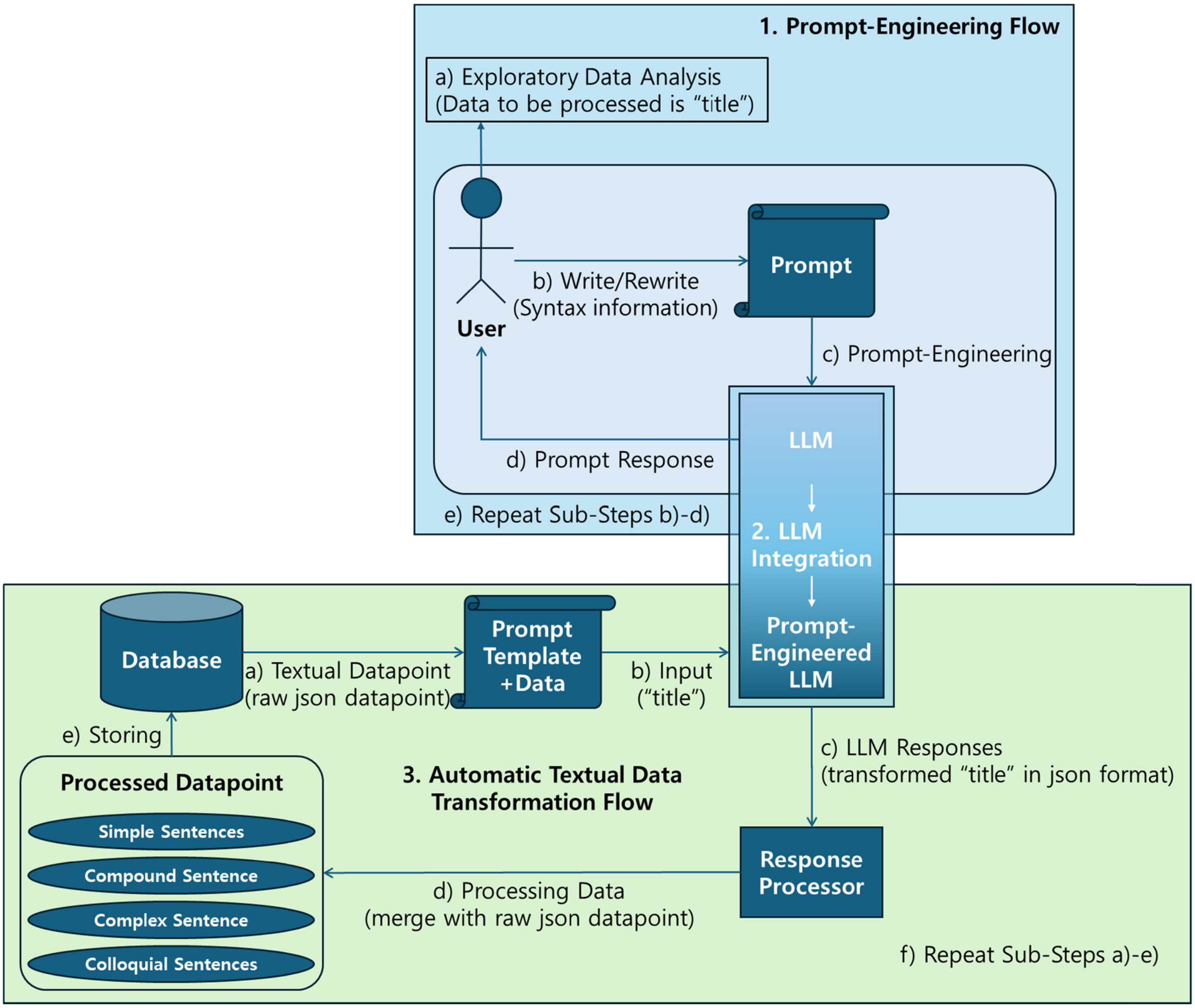


Figure 2. Automatic Textual Data Transformation

3. Automatic textual data transformation flow: This step uses the integrated LLM to transform given passages to different types of sentences, i.e. simple, compound, complex, and colloquial.

a) Textual datapoint: The first textual datapoint in raw json format is selected. This is combined into the prompt template to help LLM process the datapoint and examine the validity of the response. This is done with the **A) retrieval algorithm**.

b) Input: The template with the necessary information, i.e. "title" is entered into the engineered LLM. This is done with the **B) enter prompt algorithm**.

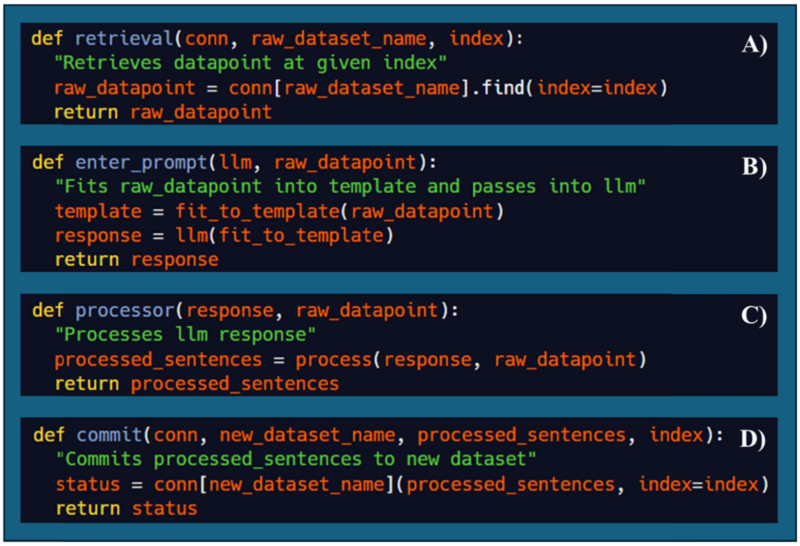
c) LLM responses: The LLM transforms input “title” into different types of sentences in a basic json format. This is passed to the Response Processor.

d) Processing data: The Response Processor merges the transformed sentences with the raw json datapoint. This is done with the **C) processor algorithm**.

e) Storing: The processed datapoint is committed to the database. This is done with the **D) commit algorithm**.

f) Repeat sub-steps a)-e): This flow is repeated until all original datapoints are processed.

The algorithms mentioned above are shown in Figure 3, marked with uppercase letters on top right.



```
def retrieval(conn, raw_dataset_name, index):                          A)
  "Retrieves datapoint at given index"
  raw_datapoint = conn[raw_dataset_name].find(index=index)
  return raw_datapoint
```

```
def enter_prompt(llm, raw_datapoint):                                  B)
  "Fits raw_datapoint into template and passes into llm"
  template = fit_to_template(raw_datapoint)
  response = llm(fit_to_template)
  return response
```

```
def processor(response, raw_datapoint):                                C)
  "Processes llm response"
  processed_sentences = process(response, raw_datapoint)
  return processed_sentences
```

```
def commit(conn, new_dataset_name, processed_sentences, index):  D)
  "Commits processed_sentences to new dataset"
  status = conn[new_dataset_name](processed_sentences, index=index)
  return status
```

Figure 3. Algorithms used in automatic textual data transformation

Choosing KONI-Llama3-8B-Instruct-20240729 [13] as the LLM, we augmented 45,678 news headlines and obtained 182,712 sentences, with each sentence type having 45,678 sentences. This will be used as part of the fine-tuning dataset to enhance f1-score on Korean language topic classification. The complete workflow is shown in Figure 4.

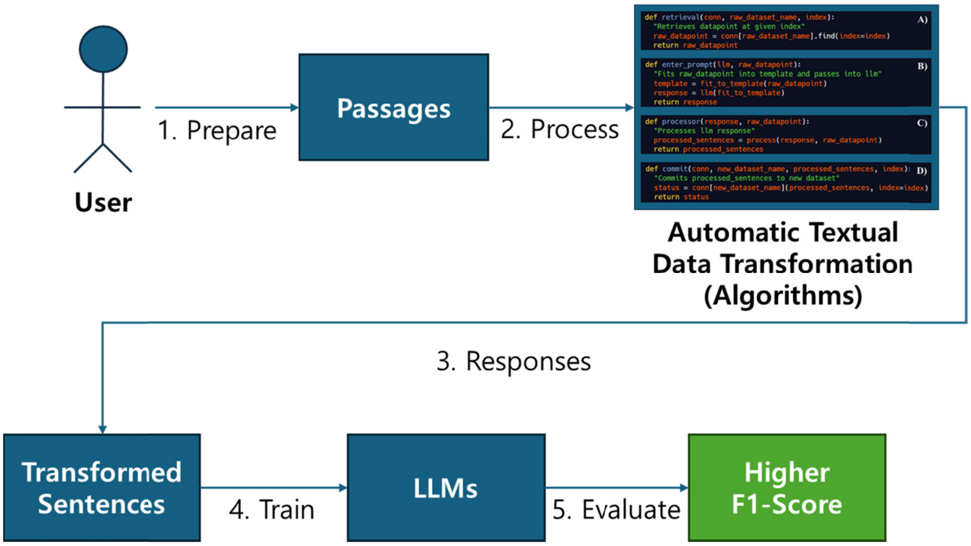


Figure 4. Complete workflow for automatic textual data transformation for enhancing f1-score

4. CONCLUSION

We mention the mechanism for enhancing f1-score of LLMs on the classification task. Using the automatic textual data transformation, we augment 45,678 Korean news headlines into 182,712 sentences of the types simple, compound, complex, and colloquial. This can be used to train LLMs to understand diverse Korean language syntax, in the hope that their responses match the user's expectation. For the future work, we will select a few LLMs to apply the augmented dataset and conduct comprehensive evaluation.

ACKNOWLEDGEMENT

This research was supported by Korea Creative Content Agency (KOCCA) grant funded by the Ministry of Culture, Sports and Tourism (MCST) in 2024 (Project Name: Artificial Intelligence-based User Interactive Storytelling 3D Scene Authoring Technology Development, Project Number: RS-2023-0022791730782087050201) and National Research Foundation (NRF), Korea, under project BK21 Four.

REFERENCES

[1] OpenAI, "Introducing ChatGPT," OpenAI, https://openai.com/index/chatgpt/, Accessed October 31, 2024.

[2] M. Abdullah, A. Madain, and Y. Jararweh, "ChatGPT: Fundamentals, Applications and Social Impacts," In *Proc. of 2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2022.

[3] C. Lee, "Design to Improve Educational Competency Using ChatGPT," *IJIBC*, vol. 16, no. 1, pp. 182-190, 2024.

[4] S. Balasubramaniam, S. Kadry, A. Prasanth, and R. Dhanaraj, *Generative AI and LLMs: Natural Language Processing and Generative Adversarial Networks*, Berlin, Boston: De Gruyter, 2024.

[5] D. Jang, S. Byun, H. Jo, and H. Shin, "A Comprehensive Korean Instruction Toolkit on 19 Tasks for Fine-Tuning Korean Large Language Models," *arXiv*, 2024.

[6] S. Park et al., "KLUE: Korean Language Understanding Evaluation," *arXiv*, 2021.

[7] S. Schulhoff et al., "The Prompt Report: A Systematic Survey of Prompting Techniques," *arXiv*, 2024.

[8] T. Wu, M. Terry, and C. Cai, "AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts," in *Proc. of 2022 CHI Conference on Human Factors in Computing Systems*, no. 385, pp. 1-22, 2022.

[9] X. Zhou, X. Zhao, and G. Li, "LLM-Enhanced Data Management," *arXiv*, 2024.

[10] S. Zhang et al., "Instruction Tuning for Large Language Models: A Survey," *arXiv*, 2024.

[11] J. Kim, Y. Lee, Y. Han, S. Jung, and H.-J. Choi, "Does Incomplete Syntax Influence Korean Language Model? Focusing on Word Order and Case Markers," *arXiv*, 2024.

[12] bigwhitebird, "How to write simple with ChatGPT (and why it works)," Reddit, https://www.reddit.com/r/ChatGPTPro/comments/1bnlxke/how_to_write_simple_with_chatgpt_and_why_it_works/, Accessed November 5, 2024.

[13] KISTI, "KIST-KONI/KONI-Llama3-8B-Instruct-20240729," KISTI, https://huggingface.co/KISTI-KONI/KONI-Llama3-8B-Instruct-20240729.