Smart Knowledge Engine based on WebKB

Yuxi.Guo Hongyu.Shuai Fenglin.Yang
SWUFE HKU CUFE

1 Data set

WebKB:

- Main dataset: collecting 8000+ data regarding universities' websites, classified into eight categories with unbalanced distribution.
- A relational data set describing both pages and hyperlinks.
- A subset of the 4 Universities dataset containing web pages and hyperlink data.

2 Project idea

We will start by crawling content from a set of predefined webpages using web scraping techniques. The crawled data will be cleaned and normalized, including tasks such as removing HTML tags, handling missing data, and standardizing text formatting.

Data Collection (Spider System):

- Simple Spider: This module is responsible for crawling the web and collecting HTML content from specified pages. The crawled HTML files are sent to the spider system for processing.
- Agent Crawler: This is an auxiliary component that handles complex issues during the crawling process, such as dealing with anti-crawling mechanisms and handling IP proxy switching.

Data Cleaning and Feature Extraction (NLP):

- Feature Extraction: After obtaining the HTML content, this step involves cleaning the data and normalizing the text. Natural Language Processing (NLP) techniques are applied to extract relevant features from the web page content, transforming it into a suitable format for machine learning models.
- Webpage Classification: After feature extraction, machine learning or deep learning algorithms are used to classify the web pages. These algorithms can include traditional models like SVM, Naive Bayes, and Random Forest, as well as advanced deep learning models like BERT, GPT, etc.

Output (New Content): After webpage classification, the system outputs processed content, which may include classification results or further analysis based on the classification model.

3 Methods & Software

Stage 1: Data Cleaning

Libraries Used: Requests, BeautifulSoup / Scrapy

Preprint. Under review.

Stage 2: Extract Information, Entities and Features

Libraries Used: NLTK, Spacy, Matplotlib/Seaborn

Stage 3: Deep Learning Models

Libraries Used: PyTorch, Transformers (by Hugging Face): For pre-trained models like BERT and GPT, and fine-tuning them for our classification task. DGL (Deep Graph Library) or PyTorch Geometric: For implementing GNNs, GCNs, and GATs.

4 Papers to read

- Learning to Construct Knowledge Bases from the World Wide Web. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery. To appear in Artificial Intelligence .
- Zhang, J., Gao, C., Zhang, L., Li, Y., and Yin, H. (2024). SmartAgent: Chain-of-user-thought for embodied personalized agent in cyber world. arXiv. https://arxiv.org/abs/2412.07472
- Debruyne, C., and Crotti Junior, A. (2024). Apples to apples: Establishing comparability in knowledge generation tasks involving users. arXiv. https://arxiv.org/abs/2412.16766

5 Work division

We will read the papers above ahead of beginning our project. And then we will work according to the following division of labor:

- Guo Yuxi Manage the project process and design structure and flow chart on github, code spiders and text cleaning components, extract entities within webpages
- Shuai Hongyu Develop and implement machine learning and deep learning models for webpage classification, including fine-tuning pre-trained models and research and apply GNNs, GCNs, and GATs using libraries like DGL or PyTorch Geometric for relational dataset analysis.
- Yang Fenglin will work on the classification based on basic models for further selections and visualizations works.

Finally we select some DL models with relation datasets for better accuracy and finish the poster with collaboration.

6 Weekly Milestones

Week 2: Literature review: Highlight potential feasible models like Machine Learning, GNN, and Transfer Learning (each with 1-2 sentences of details). Meanwhile, complete and implicate the crawling program and pre-process the gathered data.

Week 3: Algorithm Exploration: Try different algorithms to identify models with better accuracy and evaluate each model based on results.

Week 4: Code Polishing and Poster Draft: Rerun the codebase, refine it, and complete the initial draft of the poster.

Week 5: Final Adjustments: Complete remaining tasks, such as visualizing progress, and further improve the poster.