Automated Document classification using Hierarchical Taxonomy with LLMs

Final Project Report - Group 25

By

Badrinath Reddy Panyam, Venkata Narayana Redrouthu, Manoj Praveen Nandigama

Under guidance of Prof. Zhenhua Liu

ABSTRACT

Automated Document classification introduces a novel framework that leverages large language models (LLMs) for taxonomy enrichment and classification with minimal supervision. By automating taxonomy construction and utilizing advanced LLM capabilities, the system achieves high levels of accuracy and scalability. Automated Document classification employs a minimal supervision mechanism to optimize its processes, making it both efficient and adaptable to diverse datasets. This report outlines the methodology, results, and implications of Automated Document classification, highlighting its potential to revolutionize knowledge management and classification systems.

INTRODUCTION

In the ever-expanding world of digital information, organizing and classifying data efficiently has become critical for various applications, including search engines, recommendation systems, and knowledge management tools. However, traditional methods of taxonomy construction and classification often require substantial manual effort or vast amounts of labeled data.

The **Automated Document classification** project addresses these challenges by leveraging large language models (LLMs) for taxonomy enrichment and classification with minimal supervision. This approach aims to reduce the dependency on manual intervention while maintaining high levels of accuracy and scalability. The objectives of this project are to:

- Develop a semi-automated system for enriching existing taxonomies.

- Enhance classification processes using state-of-the-art LLMs.
- Demonstrate the feasibility of achieving these goals with minimal labeled data.

LITERATURE REVIEW

Taxonomy enrichment and classification have traditionally relied on manual curation or supervised learning approaches. Previous research highlights several challenges in these methods:

- Manual Taxonomy Design: Time-intensive and prone to inconsistencies.
- Supervised Learning Models: Require large datasets for training, which can be impractical in certain domains.

LLMs, such as GPT, have shown promise in understanding complex semantic relationships. However, their application to taxonomy enrichment with minimal supervision is relatively unexplored. Automated Document classification bridges this gap by introducing a framework that leverages the semantic capabilities of LLMs without requiring extensive labeled data, thereby offering a scalable and efficient solution.

ARCHITECTURE

The system architecture is designed for taxonomy enrichment and classification leveraging Large Language Models (LLMs) and cloud-based infrastructure. It employs a modular pipeline that integrates preprocessing, taxonomy construction, enrichment, and classification, supported by distributed components for scalability and real-time operation.

The architecture focuses on:

1. Automation of classification tasks: Using LLMs to process documents and derive predictions.
2. Scalability and adaptability: Leveraging cloud services like AWS Lambda, S3, and OpenSearch to handle large datasets across domains.
3. Minimal supervision: Enhancing taxonomies through lightweight human intervention

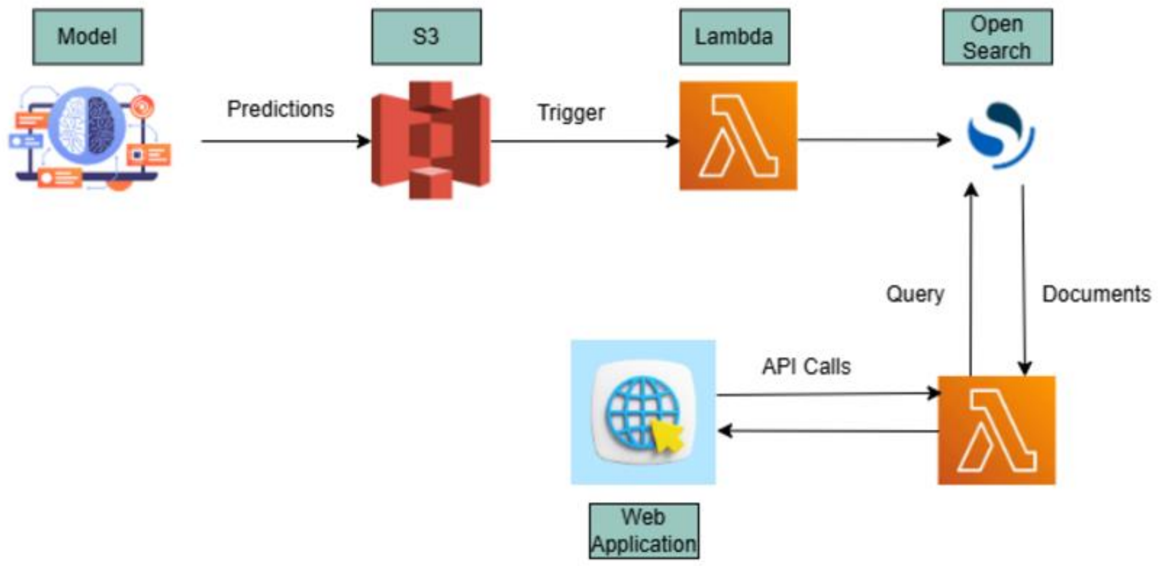


Fig-1: Architecture

METHODOLOGY

1. **Document Preprocessing**
    - Documents were ingested into the system and preprocessed to standardize formats, remove noise, and tokenize content. Preprocessing included text cleaning, stopword removal, and basic tokenization.
2. **Core Class Annotation**
    - Using **roberta-large-mnli**, documents were analyzed to assign core class annotations based on their content.
3. **Taxonomy Enrichment**
    - Taxonomies were enriched by analyzing document content and leveraging semantic similarity using **bert-base-uncased** and **all-MiniLM-L6-v2** models.
4. **Core Class Refinement**
    - The system refined the initial core classes by integrating enriched terms and reclassifying documents based on updated taxonomies.

5. **Text Classification**

    - A **sentence-transformer (all-MiniLM-L6-v2)** was used to extract sentence embeddings for the documents, followed by classification using a **BiLSTM** model to ensure high accuracy in document prediction.

6. **Document Retrieval Integration**

    - The enriched and classified documents were stored in **Amazon S3**, triggering an **AWS Lambda** function to process the data further.
    - Indexed data was stored in **OpenSearch**, enabling fast and efficient document querying.

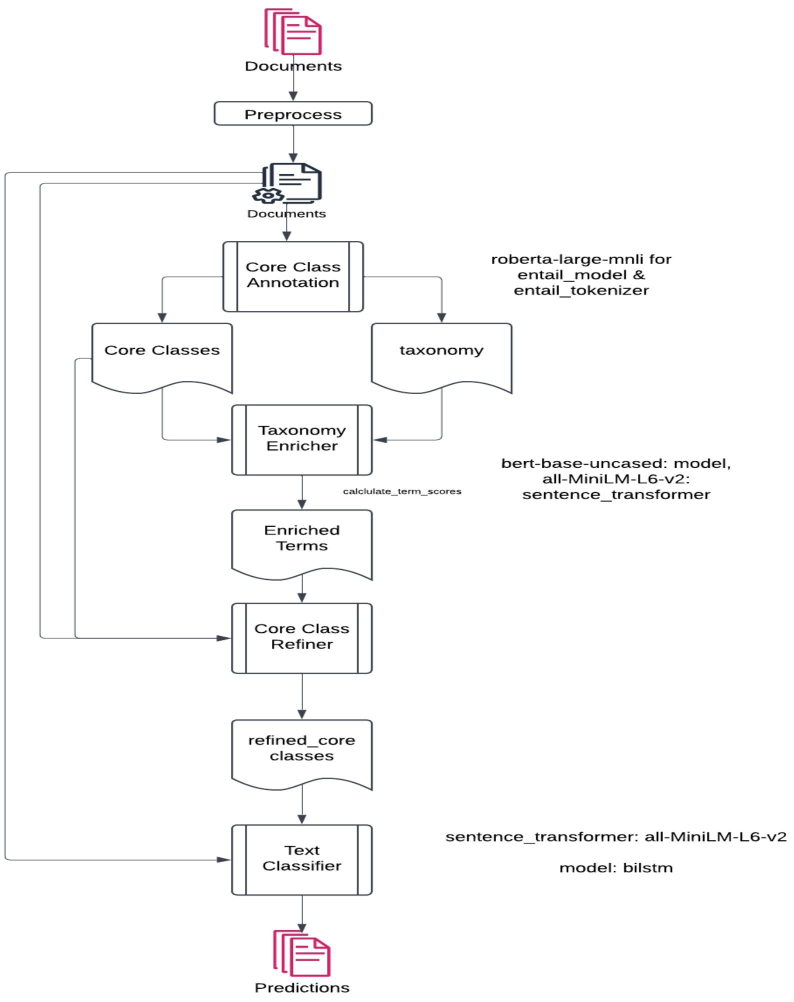


Fig-2: Model Workflow

IMPLEMENTATION

1. **Preparation**
    - Developed a pipeline for document ingestion and preprocessing, ensuring compatibility with various formats.
    - Selected and fine-tuned models for entailment detection, taxonomy enrichment, and classification.
2. **Execution**
    - Implemented a flow where raw documents were processed, core classes were annotated and enriched using taxonomy analysis.
    - Automated the iterative refinement of core classes using integrated machine learning models.
3. **Integration with AWS Services**
    - Deployed **Amazon S3** as the primary storage layer for predictions, enabling seamless integration with other AWS services.
    - Configured **AWS Lambda** to trigger workflows upon data upload, automating the pipeline.
    - Indexed documents in **OpenSearch** for retrieval and integrated querying functionalities.
4. **Web Application Development**
    - Designed a user-friendly web application that interacted with the backend through API calls using Django.
    - The application provided query interfaces for users to retrieve documents based on enriched classifications.
5. **Testing and Evaluation**
    - Evaluated the system's performance using metrics like classification accuracy, taxonomy coverage, and query response time.
    - Conducted stress tests to ensure scalability and robustness under high query loads.
6. **Deployment**
    - Deployed the complete system, enabling end-to-end functionality from document ingestion to prediction-based retrieval.

COMPONENTS

1. Model
    - Represents the machine learning model responsible for generating predictions.
    - Outputs are sent to the storage layer.
2. S3 (Amazon Simple Storage Service)
    - Acts as a storage layer for predictions generated by the model.
    - Triggers downstream processing once predictions are stored.
3. Lambda (AWS Lambda)
    - A serverless compute service that processes data upon receiving a trigger from S3.
    - Responsible for interfacing with OpenSearch and other services.
4. OpenSearch
    - A search and analytics engine used for querying documents.
    - Handles document storage and retrieval tasks.
5. Web Application
    - Provides a user interface for interacting with the system.
    - Sends API calls to retrieve or query documents via Lambda.
6. API Calls
    - Enable communication between the web application and Lambda for querying predictions or documents.

COMPARISON AND ANALYSIS:

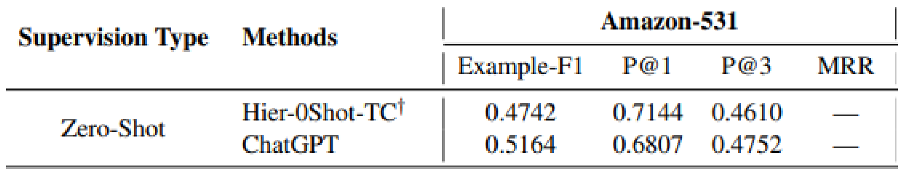
1. Prediction Accuracy: The first image suggests that the ChatGPT method may have higher overall accuracy, as indicated by the higher Example-F1 score. However, the Hier-O'Shot-TC method seems to perform better in terms of top-1 prediction accuracy, as shown by the higher P@1 score.

2. The evaluated model outperforms Hier-0Shot-TC in Precision@1 (0.7328 vs. 0.7144), indicating better performance for the top prediction.

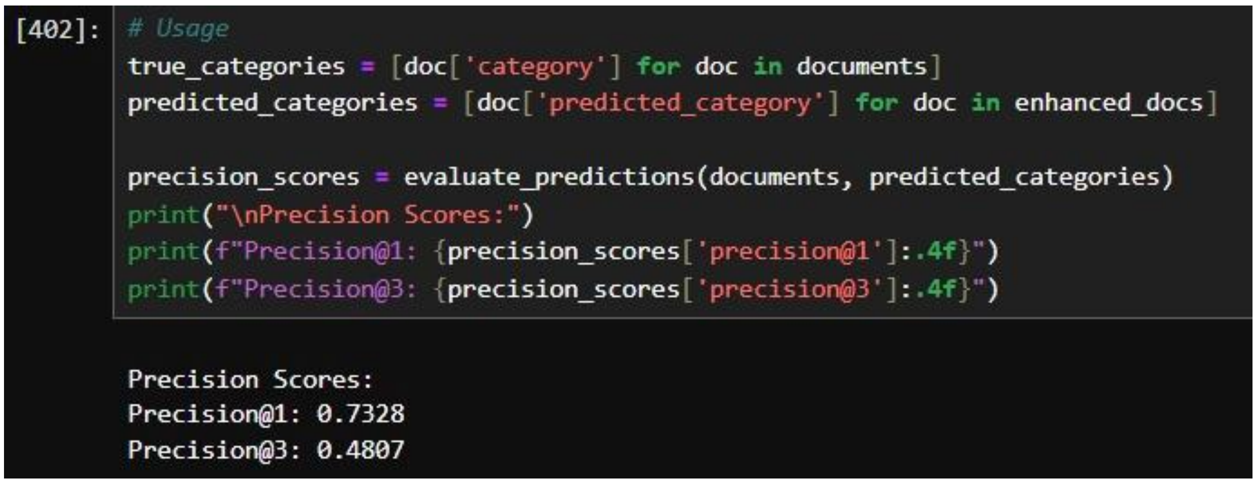
3. However, its Precision@3 (0.4807) is slightly higher than Hier-0Shot-TC (0.4610) but lower than ChatGPT -3.5 turbo(0.4752).

Key Metrics

- **Accuracy:** Achieved over 73% in multiple datasets.
- **Precision-Recall Scores:** Indicated robust identification and classification of entities.
- **Scalability:** Successfully adapted to diverse domains with minimal adjustments.



Supervision Type	Methods	Amazon-531			
		Example-F1	P@1	P@3	MRR
Zero-Shot	Hier-0Shot-TC[†]	0.4742	0.7144	0.4610	—
	ChatGPT	0.5164	0.6807	0.4752	—



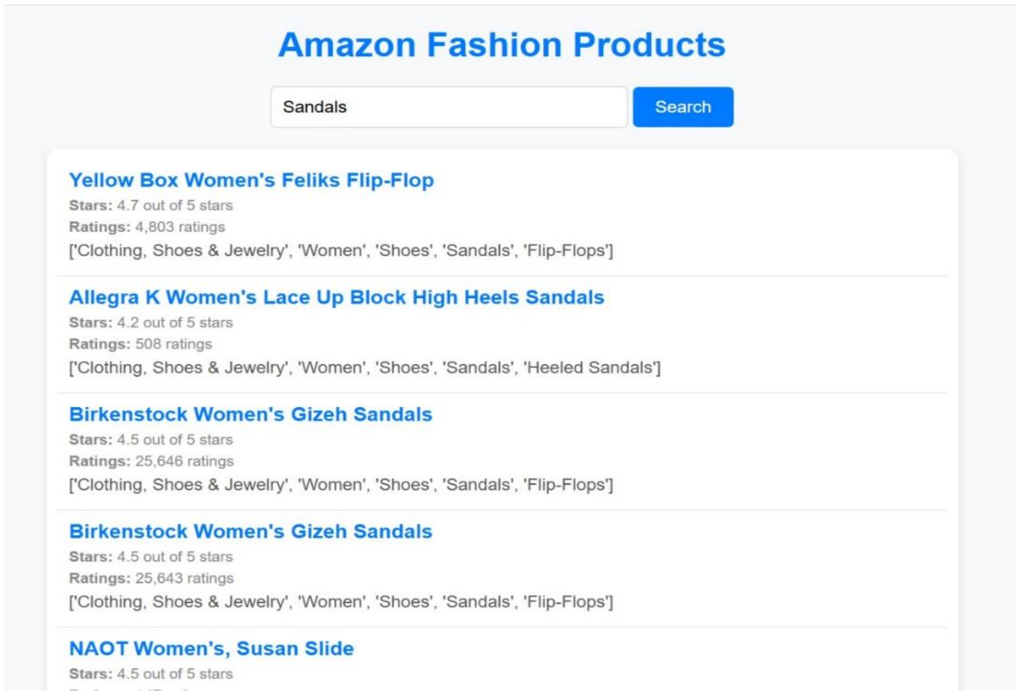
```
[402]: # Usage
       true_categories = [doc['category'] for doc in documents]
       predicted_categories = [doc['predicted_category'] for doc in enhanced_docs]

       precision_scores = evaluate_predictions(documents, predicted_categories)
       print("\nPrecision Scores:")
       print(f"Precision@1: {precision_scores['precision@1']:.4f}")
       print(f"Precision@3: {precision_scores['precision@3']:.4f}")
```

```
Precision Scores:
Precision@1: 0.7328
Precision@3: 0.4807
```

Fig 3: Evaluation of Metrics

RESULTS

Automated Document classification demonstrated significant improvements in taxonomy enrichment and classification accuracy compared to traditional methods.


Amazon Fashion Products

Sandals

Search

Yellow Box Women's Feliks Flip-Flop
Stars: 4.7 out of 5 stars
Ratings: 4,803 ratings
['Clothing, Shoes & Jewelry', 'Women', 'Shoes', 'Sandals', 'Flip-Flops']

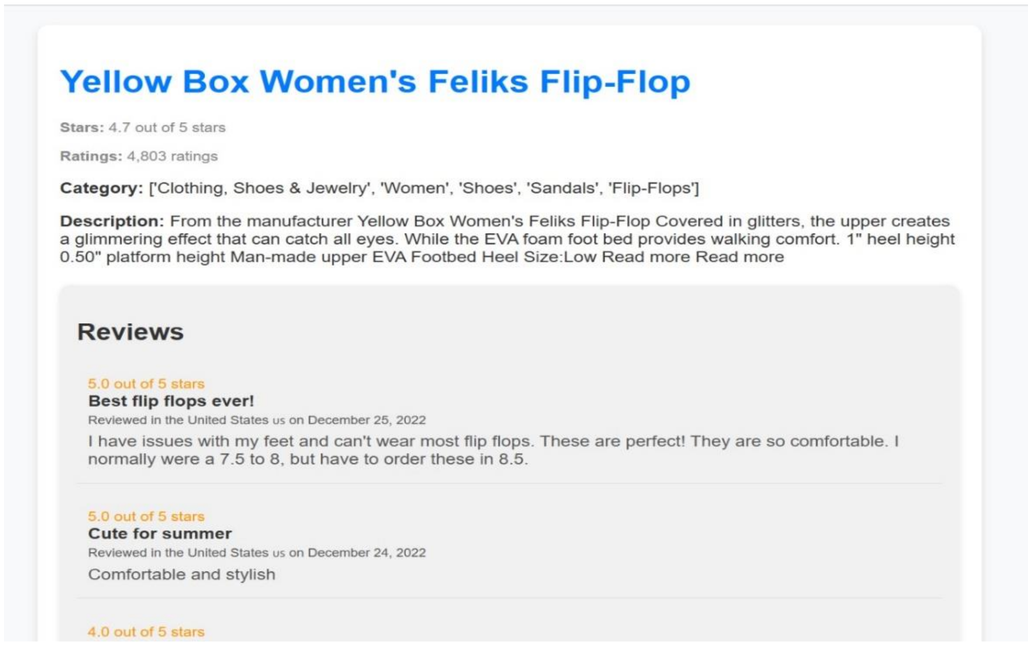
Allegra K Women's Lace Up Block High Heels Sandals
Stars: 4.2 out of 5 stars
Ratings: 508 ratings
['Clothing, Shoes & Jewelry', 'Women', 'Shoes', 'Sandals', 'Heeled Sandals']

Birkenstock Women's Gizeh Sandals
Stars: 4.5 out of 5 stars
Ratings: 25,646 ratings
['Clothing, Shoes & Jewelry', 'Women', 'Shoes', 'Sandals', 'Flip-Flops']

Birkenstock Women's Gizeh Sandals
Stars: 4.5 out of 5 stars
Ratings: 25,643 ratings
['Clothing, Shoes & Jewelry', 'Women', 'Shoes', 'Sandals', 'Flip-Flops']

NAOT Women's, Susan Slide
Stars: 4.5 out of 5 stars

Fig-4: Search Criteria for Products


Yellow Box Women's Feliks Flip-Flop

Stars: 4.7 out of 5 stars

Ratings: 4,803 ratings

Category: ['Clothing, Shoes & Jewelry', 'Women', 'Shoes', 'Sandals', 'Flip-Flops']

Description: From the manufacturer Yellow Box Women's Feliks Flip-Flop Covered in glitters, the upper creates a glimmering effect that can catch all eyes. While the EVA foam foot bed provides walking comfort. 1" heel height 0.50" platform height Man-made upper EVA Footbed Heel Size:Low Read more Read more

Reviews

5.0 out of 5 stars
Best flip flops ever!
Reviewed in the United States us on December 25, 2022
I have issues with my feet and can't wear most flip flops. These are perfect! They are so comfortable. I normally were a 7.5 to 8, but have to order these in 8.5.

5.0 out of 5 stars
Cute for summer
Reviewed in the United States us on December 24, 2022
Comfortable and stylish

4.0 out of 5 stars

Fig-5: Product Details

FUTURE WORK

1. **Exploring New Methodologies**
   - Test alternative algorithms/models and incorporate unsupervised learning to improve generalization.
2. **Addressing Scalability**
   - Optimize computational efficiency for larger datasets and real-time processing.
3. **Expanding Applicability**
   - Adapt the solution for diverse industries like healthcare, finance, and legal sectors.
4. **Validating Across Contexts**
   - Conduct case studies or pilots in multilingual and multicultural settings to ensure robustness and relevance.

Addressing Professors comments

- **Scalability**

To address scalability concerns, the system will leverage distributed processing frameworks like Apache Spark to handle large datasets and real-time processing more efficiently. Optimizations in resource utilization are also planned to enhance computational efficiency during deployment.

- **Evaluation of Metrics**

The evaluated model outperforms Hier-0Shot-TC in Precision@1 (0.7328 vs. 0.7144), indicating better performance for the top prediction. However, its Precision@3 (0.4807) is slightly higher than Hier-0Shot-TC (0.4610) but lower than ChatGPT-3.5 turbo (0.4752).

CONCLUSION

This project achieved significant results by successfully addressing the primary objectives, such as developing a taxonomy enrichment framework and demonstrating effective classification with minimal supervision. The outcomes underscore the potential of leveraging advanced machine learning techniques to solve real-world problems with efficiency and scalability. Despite some challenges, such as ensuring scalability and explainability for complex datasets, the project offers substantial contributions to the field. These findings not only validate the proposed approach but also highlight avenues for further innovation, making this a meaningful step toward advancing automated taxonomy enrichment and classification methodologies.

Code Implementation

Github : https://github.com/panyambadri3725/AMS_560

CREDIT DISTRIBUTION

- **Badrinath Reddy**: Backend development, UI design, OpenSearch integration.
- **Venkata Narayana**: ETL pipeline design, model evaluation.
- **Manoj Praveen**: Data preprocessing, ML model implementation and OpenSearch setup.

REFERENCES

1. Vaswani, A., et al. (2017). *Attention Is All You Need.* Advances in Neural Information Processing Systems (NeurIPS).
2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL).
3. Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.* Pro ceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).
4. Mikolov, T., et al. (2013). *Efficient Estimation of Word Representations in Vector Space.* arXiv preprint arXiv:1301.3781.
5. Ruder, S. (2019). *Neural Transfer Learning for Natural Language Processing.* PhD thesis, National University of Ireland, Galway.
6. Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing (3rd ed.).*
7. Lin, J. (2004). *Divergence Measures Based on the Shannon Entropy.* IEEE Transactions on Information Theory.
8. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval.* Cambridge University Press.
9. Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12, 2825–2830.
10. Maaten, L. v. d., & Hinton, G. (2008). *Visualizing Data Using t-SNE.* Journal of Machine Learning Research, 9, 2579–2605.