A Comparison of Classical and Modern Information Retrieval Approaches on Recipes

Design Document

Group Number: 22

Markus Auer-Jammerbund
auer-jammerbund@student.tugraz.at

Thomas Knoll
thomas.knoll@student.tugraz.at

Jonas Pfisterer
jonas.pfisterer@student.tugraz.at

Thomas Puchleitner
thomas.puchleitner@student.tugraz.at

November 10, 2024

1 Abstract

The internet is an ever-expanding source for a variety of recipes. However, finding recipes which are catered to one's taste, personal preferences, eating habits, and own cooking abilities becomes increasingly harder as the number of recipes increases. Traditional information retrieval algorithms may not retrieve recipes in an optimal way. In this project, we aim to use advanced information retrieval methods to retrieve recipes and compare the results with those obtained from traditional algorithms. We will further assess the relevance of the retrieved recipes from the two approaches.

2 Idea and Main Task

For this project we will use 3 different approaches to retrieve a set of similar recipes for a given query recipe. We will employ *TF-IDF* , a classic information retrieval method, as a baseline, as well as *Word2Vec* [1], and *BERT* [2] embeddings, which represent more advanced information retrieval techniques.

When querying with a recipe, we will use the embeddings to find similar recipes to the one in the query. We will investigate the following questions:

- Do advanced information retrieval methods retrieve recipes that are more similar in terms of ingredients, recipe difficulty, and other metrics (such as flavor profile or region) than the baseline approach using *TF-IDF* ?
- Does the transformer-based approach with *BERT* yield better results than the *Word2Vec* and *TF-IDF* approaches?

The main tasks of the project are the following:

- **Project Implementation and Experiments.** The project implementation will include pre-processing the dataset, which involves modifying the dataset to include the embeddings created with *Word2Vec* and the *BERT* model. Additionally, the *TF-IDF* table has to be computed.
  When handling a query, the following steps are required:
  Before querying our dataset with the user-provided recipe, we need to extract the *Word2Vec* and *BERT* embeddings as well as the *TF-IDF* representation. We can then compute the *Cosine Similarity* between the query representation and the entries in the dataset and retrieve a certain number of the most similar recipes for each method. These results are displayed to the user.
  The retrieved recipes from the different methods will be compared and evaluated in terms of similarity and relevance.
- **Report and Presentation.** We will summarize our findings and results in a report and present them in the lecture. We aim to create plots and figures to display the results of our evaluation. Additionally, we will propose future work and discuss the limitations of our information retrieval setup.

3 Dataset and Processing

To find a proper dataset, we browsed the available datasets on HuggingFace. Specifically, we found a well-populated dataset of recipes with around one million entries: RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation[3].

To adapt the dataset for our use cases, we will annotate the database with the results of the *TF-IDF* , *Word2Vec* , and *BERT* embeddings for the use in *Cosine Similarity* computations.

Since we are not certain if we will be able to handle the full dataset, we might need to adjust the dataset size for the final project.

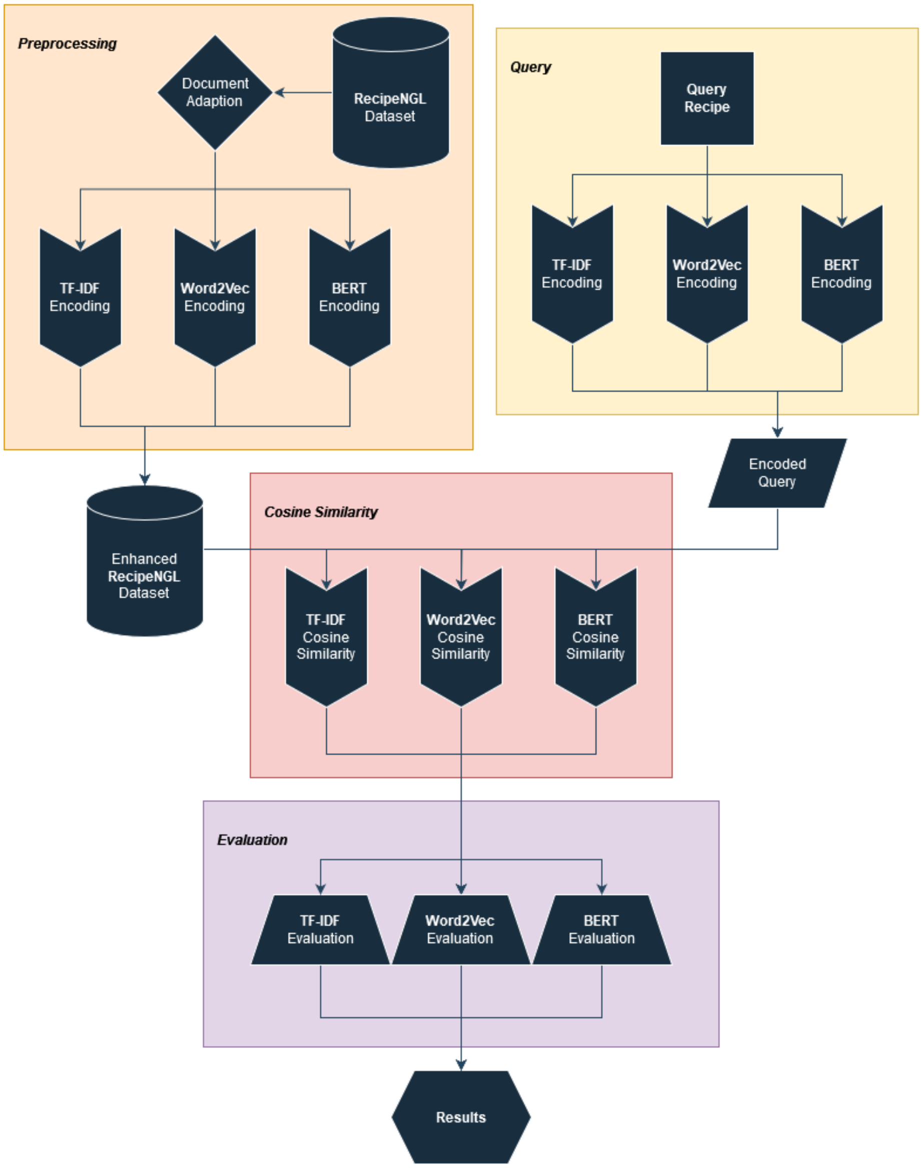

Figure 1: Shows the pipeline of our AIR project.

4 Methods and Models

- **TF-IDF.** We use *TF-IDF* as a traditional information retrieval method and as a baseline for comparison with the advanced information retrieval methods. We create a *TF-IDF* table for our preprocessed dataset and use *Cosine Similarity* to find similar recipes based on our query recipe.

- **Word2Vec.** We apply *Word2Vec* to extend the dataset with vector representation embeddings.
- **Bert-Embeddings.** We use *BERT* (Bidirectional encoder representations from transformers) to enhance our dataset by representing text as a sequence of vectors using self-supervised learning. The model we use is a pre-trained one. We use it to add embeddings to our dataset and computing the *Cosine Similarity* between the query recipe and recipes in the dataset to retrieve the most relevant ones.
- ***Cosine Similarity* .** We use cosine similarity to measure the similarity between recipes across all three types of embeddings. Then, we rank the recipes by similarity and return the most relevant ones to the user.

Our *BERT* model is also available on HuggingFace. Specifically, we would selected RecipeBERTto produce *BERT* embeddings. For *TF-IDF* and *Word2Vec* we will use implementations from existing python libraries.

5 Evaluation

We will individually judge the precision score for the retrieved recipes. This results in four precision scores per query for each method. We will take the mean of these scores to arrive at a single score per method per query. Furthermore, we will compute an overall mean for each method, resulting in three scores that represent the strength of each method. For example, assume we have n query recipes and k persons ranking each result. Then the overall score S_m for method m would be defined as follows (where $I_{i,j}$ represents the individual precision rating of person j for the given query recipe i):

$$S_m = \frac{1}{n \cdot k} \sum_{i=1}^{n} \sum_{j=1}^{k} I_{i,j}$$

Finally, we will use these results to address our research questions and compare the different retrieval methods.

6 Member Roles

We plan to distribute the tasks among the members as follows:

- **Markus Auer-Jammerbund:** Report, Query pipeline setup, Evaluation
- **Jonas Pfisterer:** Report, *Word2Vec* embeddings, Evaluation
- **Thomas Knoll:** Design Document, *BERT* embeddings, Evaluation
- **Thomas Puchleitner:** Design Document, *TF-IDF* handling, Result processing, Evaluation

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: https://arxiv.org/abs/1301.3781

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[3] M. Bień, M. Gilski, M. Maciejewska, W. Taisner, D. Wisniewski, and A. Lawrynowicz, "RecipeNLG: A cooking recipes dataset for semi-structured text generation," in *Proceedings of the 13th International Conference on Natural Language Generation.* Dublin, Ireland: Association for Computational Linguistics, Dec. 2020, pp. 22–28. [Online]. Available: https://www.aclweb.org/anthology/2020.inlg-1.4