A Comparison of Classical and Modern Information Retrieval Approaches on Recipes

Group Number: 22

Members:
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

Project-link: https://github.com/Dowakiin/Advanced-Information-Retrival-WS24-25-Group22

1 Introduction

In recent years, sharing recipes on social media and other sites has become increasingly popular, and many people have traded their old, trusted cookbooks for the internet as source for new recipes and cooking ideas. Many recipe-sharing platforms exist, and the number of available recipes is ever-increasing. While the increase in available recipes offer greater variety and possibilities, it also makes it harder to find recipes catered to one's taste, preferences, eating habits, and cooking abilities.

Traditional information retrieval systems might be unable to handle the ever-increasing number of recipes. They might fail to capture specific attributes and categories without extensive preprocessing. Advanced information retrieval systems might be better suited to handle many recipes. Methods like *Word2Vec* and *BERT* can capture semantic relations between the ingredients, and might provide more relevant results than traditional approaches.

This project explores the application of both traditional and advanced information retrieval methods to recipe retrieval. It aims to evaluate the effectiveness by performing different queries that reflect how users might search for new recipes or general cooking ideas and assess the relevance of the retrieved recipes from the traditional and advanced approaches. The queries differ from particular queries to retrieve specific recipes to more general queries for broader recommendations. The main research questions are how well both approaches perform and whether advanced information retrieval systems provide better results than the traditional simple approach and which methods are better suited for which queries.

2 Related Work and Project Discussion

2.1 Used Methods

TF-IDF (Term Frequency - Inverse Document Frequency) is the classical information retrieval method that is used here as comparison to the ones that can capture semantic meaning. The idea behind it, is to measure the importance of a word in a document d compared to a corpus. This is done by measuring the *term frequency* $\mathrm{TF}(t, d)$ (the relative frequency of a term t within a document d), and the *inverse document frequency* $\mathrm{IDF}(t)$ (a measure of how much information a term provides). *TF-IDF* is then simply computed by multiplying TF and IDF.

TF-IDF is typically used in, information retrieval (to rank documents in response to a query), text mining and natural language processing (NLP) (to perform feature extraction), and search engines (to prioritize pages containing terms relevant to the users query). In 2015, about 83% of text-based recommender systems used *TF-IDF* .

Word2Vec [1] is a technique used in NLP. It finds a vector representation for words to capture information about the meaning of each word relative to the surrounding words. In more mathematical terms, it creates a word embedding by mapping words to a continuous vector space where similar words are closer to each other.

There are two main methods to get the word embeddings, *Continuous Bag Of Words (CBOW)* and *Skip-Gram*. *CBOW* tries to predict words based on the surrounding words. Thus, the computational objective is to maximize the probability of a target word given its context. *Skip-Gram* is the reverse of the *CBOW* method; it tries to predict the context given a word. The same goes for the objective, which is to maximize the probability of context words given a target word.

Typical usages of *Word2Vec* are quite similar to *TF-IDF* . They include NLP (for text classification, sentiment analysis and entity recognition), recommender systems (for finding similar items), and search and query expansions (to identify semantically related words for better search results).
BERT [2] (Bidirectional Encoder Representation from Transformers) is a state-of-the-art transformer-based model for NLP tasks. Similar to *Word2Vec* , it generates embeddings of words based on the surrounding words. Unlike *Word2Vec* , which has static embeddings, *BERT* has dynamic embeddings.

The embeddings are computed in four stages:
Input Representation: The given input gets split into sequences of words. Those then get tokenized into smaller units. Additionally, special tokens are added. Namely, [CLS] to mark the start of a sequence and [SEP] to separate two sentences or mark the end of a sentence.
Bidirectional Transformer: Unlike traditional models, *BERT* looks at the context from both the left and right sides of a token simultaneously to more deeply understand the meaning and context of a text. This approach is called bidirectional attention mechanism.
Pre-training Tasks: *BERT* gets pre-trained on large corpora using *Masked Language Modeling (MLM)* and *Next Sentence Prediction (NSP)*.
Fine-Tuning: *BERT* can be fine-tuned for specific tasks by adding a task-specific head on top.
As a more advanced and modern method, *BERT* has a broader spectrum of use-cases. Including, but not limited to Text classification (for sentiment analysis, spam detection, etc.), named entity recognition (for extracting entities like names, dates, or locations from text), question answering (by understanding questions and finding relevant answers), and machine translation and summarization (by generating context-aware translations and summaries).

2.2 Comparison

The main drawback of *TF-IDF* is that it only takes the words themselves into account. It is completely context-unaware. On the other hand, it is pretty fast and light weight. That is also the main advantage that *Word2Vec* has over *BERT* . It also has limited context awareness, which based on the surrounding words. Thus, the context awareness is heavily influenced by the window size of surrounding words that are taken into account. Meanwhile, *BERT* is by far the most computational expensive method All in all, *TF-IDF* works best for simple tasks such as ranking in search engines or spam detection. It also needs relatively little data to work well. Moving on, *Word2Vec* shines on semantic tasks and clustering, for example Recommendation and similarity. While it still needs more datasets, it still works with less data then *BERT* . Speaking of which, for the heavy computational costs and large amount of data needed for training, *BERT* can handle

complex tasks such as Question answering and translation.

2.3 Related Research

There already exists some research comparing those IR methods. For example, the paper "Performance comparison of tf-idf and word2vec models for emotion text classification" [3] compares two versions of *TF-IDF* and one *Word2Vec* version. The task was to, first, distinguish between texts that contain emotion and texts that do not, and further classify five different emotions. The result was that both *TF-IDF* versions yielded better results than *Word2Vec* . Another one is "Tweet recommendation using Clustered Bert and Word2vec Models" [4]. Very similar to our project, the overall goal of that paper was to compare *Word2Vec* and *BERT* in recommending tweets to a user. It found out, that *BERT* way outperformed *Word2Vec* . In a third one, "Comparative Analysis of Machine Learning Algorithms for Email Phishing Detection Using TF-IDF, Word2Vec, and BERT" [5], compared all three methods for their ability to identify phishing emails. The results were, that *TF-IDF* and *Word2Vec* performed about the same while *BERT* was the clear winner.

2.4 Project Discussion

Regarding this project, and taking into account how the different methods word as well as what related projects have already shown, it is to be expected that *TF-IDF* will perform best for easy queries, because linking specific words to documents associated with that term is literally what it was made for. But the more complicated the queries get, it will perform increasingly worse because it does not take the context into account. *BERT* has exactly the same problem in reverse. While it will stomp the competition in the harder queries, it needs context to work well. Due to that, the short, easy queries could be misinterpreted. Last but not least, one could assume that *Word2Vec* will perform somewhere in-between. But, as some related research has shown, *Word2Vec* actually performs closer to *TF-IDF* and is sometimes even outperformed by the simpler approach.

3 Experiments and Results

Table 1: *Accuracy of Methods*

Difficulty\Method	*TF-IDF*	*Word2Vec*	*BERT*
Easy	82.14	53.57	64.29
Medium	51.19	55.95	70.24
Hard	11.90	22.62	50.00
Overall	48.41	44.05	61.51

3.1 Dataset

The dataset for the following experiments is RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation[6]. To keep runtime manageable, it was reduced to a smaller subset. This subset was selected by choosing the first 100.000 recipes and preprocessed to enable cosine similarity computation.

3.2 Queries

For the queries, three levels of difficulty (*easy, medium,* and *hard*) were defined, with each category containing three queries. For each query, seven recipes were retrieved using their respective cosine similarity method. Example queries are "pizza" for *easy*, "Salad, Ingredients: onion, tomato" for *medium*, and "no rice" for *hard*.

3.3 Methods (i.e. *TF-IDF*, *Word2Vec*, and *BERT*)

For *TF-IDF*, stop-words were removed and a vocabulary limit of 50,000 words was imposed, which was not reached in this dataset. For *Word2Vec*, a vector size of 256, a window size of 8, and a minimum count of 1 has been used, which are commonly used for *Word2Vec* embeddings. The *BERT* model RecipeBERT, however, is already pre-trained and fine-tuned on the used dataset.

3.4 Evaluation

After compiling the results for all queries, four evaluators independently rated each recipe against its corresponding query in a binary manner (e.g. Yes/No for relevance). The overall accuracies are presented in Table 1. The evaluation can be accessed at AIR Evaluation.

4 Conclusion

We compared advanced information retrieval methods with traditional methods on the task to retrieve recipes. The models where given queries in 3 different difficulties respectively.

Our findings revealed that *TF-IDF* performed reasonably well for simpler queries. However, as expected it struggled with the hard queries chosen by us, due to its lack of context awareness. *Word2Vec* underperformed compared to expectations, particularly for the hard queries, since it should have some limited context awareness. *BERT* delivered the best results overall, but in particularly for the medium and hard queries, which conforms to our expectation.

In direct comparison, advanced information retrieval methods outperformed traditional methods. However, the rather bad results of *Word2Vec* raise questions about its general suitability for this specific domain.

To build on these findings for future work, we suggest a deeper analysis of the weak performance of *Word2Vec*, focusing on its parameter settings, training conditions, and overall suitability for recipe retrieval tasks. Also, using the whole dataset (2 million recipes) instead should improve quality of the returned recipes across all models. However, we expect that the *BERT* model should benefit the most from using a larger dataset. To enhance the performance of the *BERT* model even more, in depth fine-tuning of the used RecipeBERT model can be investigated. Lastly, an evaluation on more than seven retrieved recipes per query will give better insights on the capabilities of each used method.

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] D. E. Cahyani and I. Patasik, "Performance comparison of tf-idf and word2vec models for emotion text classification," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2780–2788, 2021.

[4] S. Kakar, D. Dhaka, and M. Mehrotra, "Tweet recommendation using clustered bert and word2vec models," in *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, 2023, pp. i–vi.

[5] A. Al Tawil, L. Almazaydeh, D. Qawasmeh, B. Qawasmeh, M. Alshinwan, and K. Elleithy, "Comparative analysis of machine learning algorithms for email phishing detection using tf-idf, word2vec, and bert," *Comput. Mater. Contin*, vol. 81, p. 3395, 2024.

[6] 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