IRSE Projecet

Jan Cichomski (r1026448)

May 1, 2025

Outline

Architecture

TODO

1. Term Vocabulary

1.1 Term Vocabulary

- To lower case
- remove punctuation
- tokenize
- remove english stop words (added custom stop words)
- lammatize

1.1 Temr Vocabulary - Code

```
def preprocess_text(doc):
    doc = doc.translate(str.maketrans("", "",
        string.punctuation)).lower()
    words = word_tokenize(doc)
    words = [
        lemmatizer.lemmatize(word)
        for word in words
        if word not in stop_words and word.isalpha()
    ]
    return " ".join(words)
```

1.1 Term Vocabulary - Custom stop words

```
stop_words.update(
    [
        "add",
        "added",
        "adding",
        "addition",
        "also",
        "almost",
        "another",
        "easily",
        "easy",
    ]
)
```

1.2 Term Vocabulary

Two types of terms:

- 1-grams
  - min_df=20
  - max_df=0.5
- 2-grams
  - 10,000 terms
  - min_df=50
  - max_df=0.4

1.3 Term Vocabulary

- 2-grams with aggressive filtering

2 Document Embedding

2.1 Document Embedding

I use all fields for embedding:

- name
- description
- ingredients
- steps
- Tested different combinations
- Make sense as user may ask about any information

TODO: add some data

2.2 Document Embedding

The same approche as for embedding documents

```
def retrieve_documents(query_text, recipies,
                       recipe_ids, k, threshold):
query = preprocess_text(query_text)
...
```

2.3 Document Embedding - Edge Cases

- Problem: When query has no terms from vocabulary
  - TF-IDF produces zero vector for the query
  - Cosine similarity returns 0 for all documents
- Consequences:
  - Without similarity threshold: All documents returned (no filtering)
  - With any similarity threshold: No documents returned (empty result)