cb-04-szymanskir-precision_and_recall_for_best_performing_model_analysis

January 22, 2019

1 Important

`make predictions` has to be run before any cell in this notebook.

2 Imports

```
In [1]: from booksuggest.evaluation.cb_evaluation import (
            calculate_single_score, read_similar_books
        )
        from matplotlib.ticker import MaxNLocator
        import matplotlib.pylab as plt

        import pandas as pd
        import seaborn as sns
```

3 Visualization settings

```
In [2]: sns.set(context='paper', font_scale=1.2, style='ticks', palette='muted',
                rc={"axes.labelsize":16, "ytick.labelsize": 14, "xtick.labelsize":14,
                    "font.family": "sans-serif"})
```

4 Analysis

The goal of this notebook is to visualize the precision and recall scores against the amount of recommended books and find the optimal amount of books. Only the best perfoming content-based model will be considered – the tag based model.

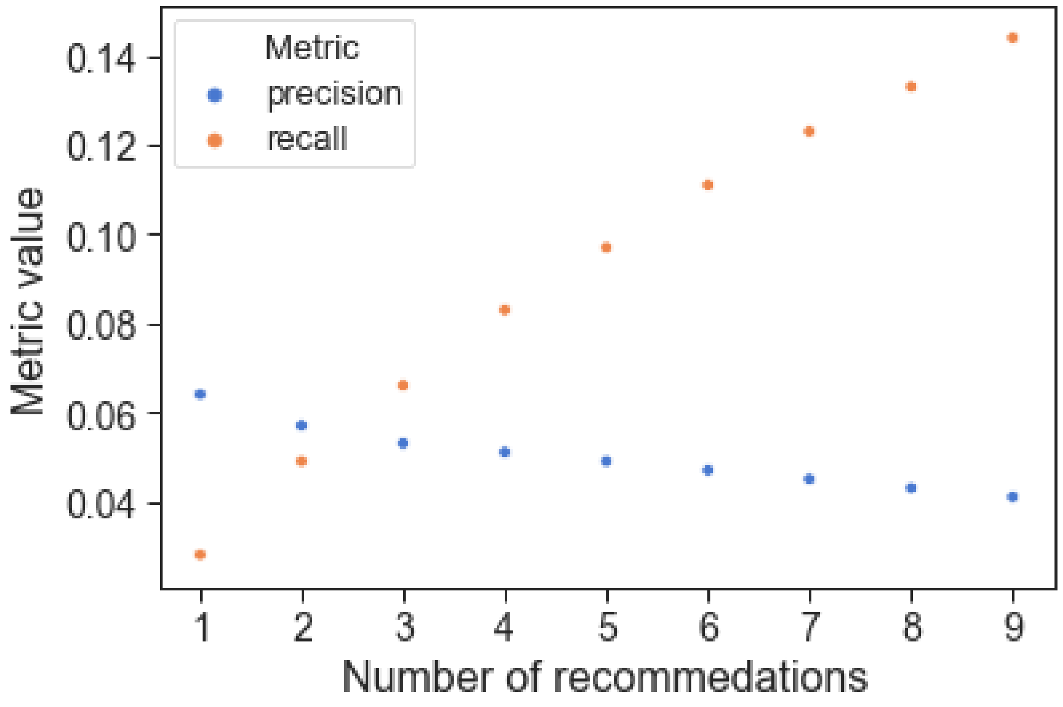
```
In [3]: tag_predictions_filepath = '../models/predictions/cb-results/tag-predictions.csv'
        test_cases = read_similar_books('../data/processed/similar_books.csv')

In [4]: rec_count_range = range(1, 10)
        scores = [calculate_single_score(tag_predictions_filepath, test_cases, rec_count)
                  for rec_count in rec_count_range]

In [5]: precision_scores = [score['precision'] for score in scores]
        recall_scores = [score['recall'] for score in scores]
        df_scores = pd.DataFrame({
            'precision': precision_scores,
            'recall': recall_scores,
            'book-number': list(rec_count_range)
        })
        df_scores = df_scores.set_index('book-number').stack().reset_index().rename(
            columns={'level_1': 'Metric', 0: 'Metric value'})
```

```
In [6]: ax = sns.scatterplot(x='book-number', y='Metric value', hue='Metric',data=df_scores)
        ax.set_xlabel('Number of recommedations')
        ax.xaxis.set_major_locator(MaxNLocator(integer=True))
        plt.setp(ax.get_legend().get_texts(), fontsize='13') # for legend text
        plt.setp(ax.get_legend().get_title(), fontsize='13') # for legend title
        # ax.get_figure().savefig('cb-tag-precision-recall-plot.pdf', bbox_inches='tight')
```

```
Out[6]: [None, None]
```



```
In [7]: ax.get_figure().savefig('cb-tag-precision-recall-plot.pdf')
```