Content-based Hashtag Recommendation Methods for Twitter
Dipesh Poudel ’22 (dpoudel18@earlham.edu)
Department of Computer Science, Earlham College

1. Abstract
This project designs and evaluates different content-based methods (Hashtag Frequency, Naïve Bayes model, and KNN-based cosine similarity) for Twitter hashtag recommendation. The results shows that Naïve Bayes model outperforms other two models on almost all metrics with an average accuracy score of 0.83.

2. Introduction
• A hashtag recommendation system is an important tool to organize similar content together for topic categorization.
• A very little research has been done on evaluating the performance of different existing models using the same dataset and the same evaluation metrics.
• This project first designs three content-based methods (Tweet similarity using hashtag frequency, Naïve Bayes model, and KNN-based cosine similarity) for hashtag recommendation.
• It then, evaluates the performance of these content-based methods (Tweet similarity using hashtag frequency, Naïve Bayes model, and KNN-based cosine similarity) using 5 evaluation metrics namely Precision, Recall, F1 Score, Hit Rate and Hit Ratio.
• The final product is a web application which lets the user input their tweet, and recommend hashtags for their tweet using these three models.

3. Software Architecture

4. Evaluation Framework

5. Data Pre-processing
• This project make use of Regex and NLTK library to clean the raw twitter data as shown below

6. Recommendation Methods
Hashtag Frequency
• The hashtags in the retrieved similar tweets are selected to be candidate hashtags which are then ranked based on which hashtags are present more and top-k hashtags are recommended.
Naïve Bayes
• Hashtags are recommended using Bayes’ Theorem as illustrated in the given formula.

\[
P(\text{Hashtag} / \text{Tweets}) = \frac{P(\text{Hashtag}) \cdot P(\text{Tweets} / \text{Hashtags})}{P(\text{Tweets})}
\]
• The top-k hashtags with the higher probabilities are recommended to the user.
KNN-based Cosine Similarity
• First, the cosine similarity between the test tweet and the train tweet is calculated.

\[
\text{CosineSimilarity}(T_{\text{test}}, T_{\text{train}}) = \frac{T_{\text{test}} \cdot T_{\text{train}}}{\|T_{\text{test}}\| \cdot \|T_{\text{train}}\|}
\]
• Then K-nearest neighbor of the test tweet ranks hashtags.

7. Evaluation Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
<th>Hit Rate</th>
<th>Hit-Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>HF</td>
<td>0.69</td>
<td>0.87</td>
<td>0.76</td>
<td>0.91</td>
<td>0.88</td>
</tr>
<tr>
<td>NB</td>
<td>0.72</td>
<td>0.86</td>
<td>0.78</td>
<td>0.93</td>
<td>0.9</td>
</tr>
<tr>
<td>CS</td>
<td>0.65</td>
<td>0.81</td>
<td>0.72</td>
<td>0.84</td>
<td>0.8</td>
</tr>
</tbody>
</table>

8. Conclusion / Future Work
• It is observed that even simple content-based recommended methods like Naïve Bayes produces good average accuracy score of 0.83.
• This project was conducted with a fairly small datasets due to Twitter API restrictions.
• There is potential to improve the performance of the models if an robust data pipeline is created with PostgreSQL database to store and query real-time tweets.

9. Acknowledgement
• I would like to acknowledge Dr. Charlie Peck and Dr. David Barbella for the immense help throughout the project.