Information Retrieval Programming Assignment 4: Language Models

Shayan Amani
(993550898, sa1149)

Soheil Gharatappeh
(933639024, sg1147)

Source Code

Please refer to our GitHub repository for the source code.

Implement Unigram Language Models with Smoothing

1. Find the `run files` for all variants in here.
2. All the measures are compared in the following table and plot.

Evaluation	MAP	Precision@R	NCCG@20
U-L	?	?	?
U-JM	0.568	0.5734	0.7275
U-DS	0.564	0.5635	0.724

3. The following table shows mean and standard error for these two evaluations.

Evaluation	MAP	Precision@R	NCCG@20
Mean	0.566	0.56845	0.72575
Standard Error	0.002	0.00495	0.00175

- According to the table, U-DS performs better than U-JM.
- As we can see in the following figures, Unigram Language Models have a better performance in comparison to TF-IDF.
- No, the deference $(MAP, Prec@R, ndcg@20) = (0.002, 0.00495, 0.00175)$ is not significant.

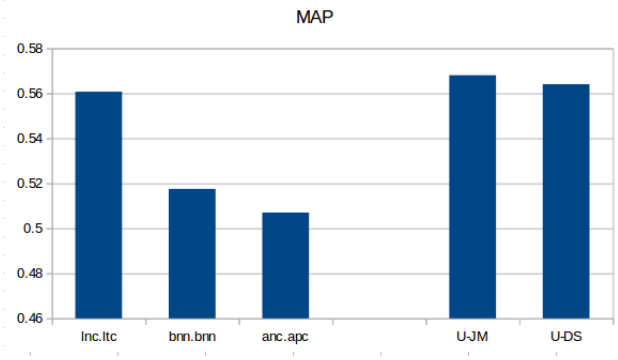

Figure 1: Map comparison.

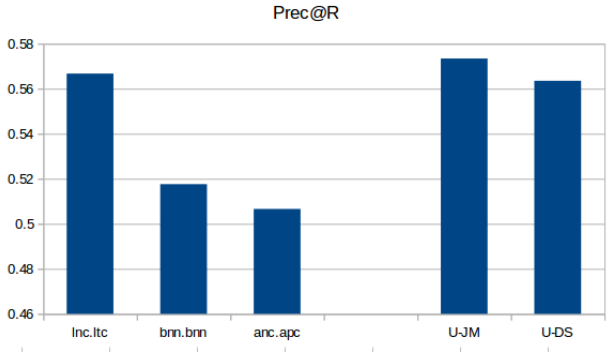

Figure 2: Prec@R comparison.

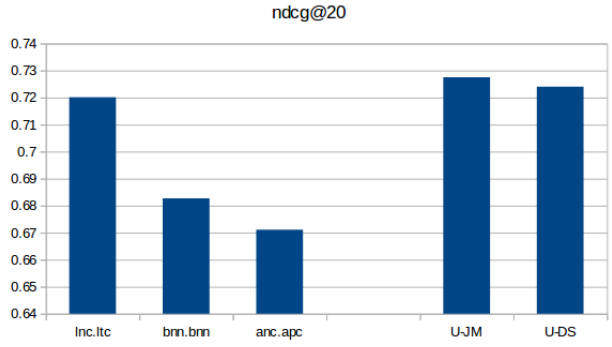

Figure 3: ndcg@20 comparison.

- No, they theoretically cannot agree on "the best" method. As mentioned, each of this measures, focuses on one thing. So, it cannot perform well in the other. The root cause of that is that we don't have a perfect model that can give us 100% precision and 100% recall. When we increase precision, the recall would decrease. Because we take a set of more strict criteria in our search engine. This is true the other way around; if we recall more docs, there is more chance to recall non-relevant docs too.

- The search result for Brush%20rabbit: Jelinek: PerformSearch: Top results found: 9 1 Q0 3062422698c70396fe4505a60c680d50022a3314 1 4.509259 Team3-Practical

  1 Q0 0760e843e1c62c7aeb1c21f994f05992876aa0a1 2 4.2766123 Team3-Practical

  1 Q0 ffa99d68412bf2f3cb97cfd421b06ebda8fa1e0c 3 4.033634 Team3-Practical

  1 Q0 1ba3f2a59e0f888ebd7141fc2295001e9b1de38a 4 3.9935496 Team3-Practical

  1 Q0 4483e0c4975618b0f54ab2bebc1a425f0b529f93 5 3.5972593 Team3-Practical

  1 Q0 3072b5c2b49c2ec4ae61e9d3f6f8b684f09e1237 6 3.3580415 Team3-Practical

  1 Q0 49b6e4fd53560f54bcbcf3337aae646bb154c4e5 7 3.3580415 Team3-Practical

  1 Q0 6353c749240c1ce37201b361c829e85c342e71a1 8 2.931142 Team3-Practical

  1 Q0 bd652e0cc7d7f2ceed3fb61a472460f6fd47a2f9 9 2.3964243 Team3-Practical

  Driklete: Top results found: 9 1 Q0 3062422698c70396fe4505a60c680d50022a3314 1 4.4249597 Team3-Practical

  1 Q0 49b6e4fd53560f54bcbcf3337aae646bb154c4e5 2 3.5834122 Team3-Practical

  1 Q0 1ba3f2a59e0f888ebd7141fc2295001e9b1de38a 3 3.3933446 Team3-Practical

  1 Q0 ffa99d68412bf2f3cb97cfd421b06ebda8fa1e0c 4 3.025436 Team3-Practical

  1 Q0 4483e0c4975618b0f54ab2bebc1a425f0b529f93 5 3.0035439 Team3-Practical

  1 Q0 0760e843e1c62c7aeb1c21f994f05992876aa0a1 6 2.402131 Team3-Practical

  1 Q0 3072b5c2b49c2ec4ae61e9d3f6f8b684f09e1237 7 2.3778613 Team3-Practical

  1 Q0 6353c749240c1ce37201b361c829e85c342e71a1 8 2.355969 Team3-Practical

  1 Q0 bd652e0cc7d7f2ceed3fb61a472460f6fd47a2f9 9 2.3102467 Team3-Practical