Leaderboard Creation Algorithms

January 29, 2021

Algorithm 1 Similarity_Score

Input: r_i, s_j
Output: Similarity score for an editor based on his/her contribution in S

```
1: function SimilarityScore(r_i, s_j)
2:     s_sent ← list of senetences in s_j
3:     r_sent ← list of senetences in r_i
4:     total_score ← 0
5:     for k ← 0 to len(s_sent) do
6:         match ← "best match" of s_sent[k] in r_i
7:         s_vec ← vectorize(s_sent[k])
8:         match_vec ← vectorize(match)
9:         score ← cosine(s_vec, match_vec)
10:        total_score ← total_score + score + GQ    ▷ GQ is the gaze quotient of s_sent[k]
11:    end for
12:    return total_score
13: end function
```

In this document, we explain in detail the procedure followed to rank editors of an article in a leaderboard. Algorithm 1 demonstrates the proposed approach to measure the similarity between a revision and an SOI. Algorithm 2 describes the procedure to update the overall similarity score for each editor in the editors' list for a given set of SOIs for an article.

Algorithm 1 shows the procedure to find the similarity score of an SOI with a particular revision of the article. First, it vectorizes the sentences using Seq2Vec [1]. Then, it finds the aggregated cosine distance between the sentence vectors of the SOI, and their best matches in the revision (if any). The "best match" is found by using

Algorithm 2 Leaderboard_Creation

Input: R, S, E
Output: Sorted list of editors based on their content in S

```
1: function LEADERBOARD(R, S, E)
2:     for s_j in S do
3:         score_old ← 0                          ▷ E_j is the list of editors for the SOI s_j
4:         for r_i in R do                        ▷ e_a ∈ E_j is the editor for revision r_i
5:             score ← SimilarityScore(r_i, s_j)
6:             if score_old == score then
7:                 editor_score ← 0
8:             else
9:                 editor_score ← score − score_old
10:                score_old ← score
11:            end if
12:            e_a ← e_a + (1/i) * editor_score
13:        end for
14:        E_j ← Sort(E_j)
15:    end for
16:    Update E_j in E
17:    return E
18: end function
```

"get_close_matches" function of "difflib" package with a cutoff of 0.65 so that it matches highly similar sentences. The similarity score between an SOI and a revision is calculated by considering cosine distance sentences of SOI and their best match found in the revision. While calculating the similarity score, we also include the gaze quotients (GQ) of the sentences in SOI to consider their readability.

In Algorithm 2, we explain the procedure to rank all the editors of an article. The editor of a revision i will get 0 contribution score, if the SimilarityScore function returns same values for i^{th} and $(i-1)^{th}$ revisions. It is because the editor of revision i has not contributed to the text, which is present in this SOI. The editor for revision i will get a positive contribution score, if the similarity score of i^{th} revision is greater than the $(i-1)^{th}$ revision. A positive score implies that the editor for revision i has constructively modified the text, which appeared in the SOI. The editor of any revision is accordingly given a negative contribution score as well.

Line 11 of algorithm 2, shows the formula to calculate the consolidated editor's score concerning a given SOI. For a given SOI, for an editor (e_i), we calculate the editor's score by taking the weighted sum of the similarity score of an editor across

all the revisions. We use the inverse of the revision index as a weighting factor to consider the contribution's longevity. There can be a situation that an editor's edits (w.r.t. the SOI) are reverted in a revision, but the fact that it appears in the current revision shows that it comes back in one of the further revisions. We rank editors according to editors' SOI scores for each SOI. For all the SOIs of a particular article, this process is repeated.

Once we have E ready, we prepare the final leaderboard of editors. General formula to calculate final editor score (FES) for an editor (x) is given in eq. (**??**). We consider an editor's rank for all the SOIs to decide the final rank of the editor on the leaderboard.

References

[1] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML*, pp. 1188–1196, 2014.