The Effect of Expanding Relevance Judgements with Duplicates

Gaurav Baruah
David R. Cheriton School of Computer Science
University of Waterloo, Canada
gbaruah@uwaterloo.ca

Adam Roegiest
David R. Cheriton School of Computer Science
University of Waterloo, Canada
aroegies@uwaterloo.ca

Mark D. Smucker
Department of Management Sciences
University of Waterloo, Canada
mark.smucker@uwaterloo.ca

ABSTRACT

We examine the effects of expanding a judged set of sentences with their duplicates from a corpus. Including new sentences that are exact duplicates of the previously judged sentences may allow for better estimation of performance metrics and enhance the reusability of a test collection. We perform experiments in context of the Temporal Summarization Track at TREC 2013. We find that adding duplicate sentences to the judged set does not significantly affect relative system performance. However, we do find statistically significant changes in the performance of nearly half the systems that participated in the Track. We recommend adding exact duplicate sentences to the set of relevance judgements in order to obtain a more accurate estimate of system performance.

Categories and Subject Descriptors: H.3.4 [Systems and Software Performance evaluation]: Efficiency and Effectiveness

Keywords: Duplicate Detection; Evaluation; Pooling

1. INTRODUCTION

The Temporal Summarization Track (TST) at TREC 2013 [2], required returning information relevant to topics, from a web-scale time-ordered document stream (the TREC KBA 2013 Stream Corpus [1]). The Sequential Update Summarization (SUS) task in the TST, called for participating systems (runs), to return updates (sentences) about events (topics), with the goal that new updates should contain information that is new to the user.

For the SUS task, the Stream corpus is effectively a collection of documents (along with their timestamps), spanning the time period between October 2011 to January 2013, with an average of 93,037 documents per hour. Each participating system was tasked to return updates from a duration spanning 240 hours for each topic. In all, the TST received 28 runs from the participants of the SUS task. The number of updates returned in the runs varied from 110 to 2,815,808.

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609534.

For evaluation of runs, the pool of updates judged by track assessors contains 9,113 sentences for 9 task topics.

We found that there are a very large number of duplicate sentences in the corpus. Indeed, the track organizers found duplicates amongst the sentences sampled from runs, while constructing the pool for assessment. Accordingly, they created an evaluation framework designed to accommodate for duplicates within the judged set of sentences. In effect, the identifier of a duplicate sentence (within the judged set of sentences) is mapped to the identifier of the designated original sentence (prototype). However, the duplicates of judged sentences from the corpus, were not mapped to the prototypes. The track's evaluation is also designed to omit sentences that are not in the judged set. Such omission is consistent with similar approaches [7, 3], and works very well for evaluating relative system performance (Section 3.1).

Including exact duplicates of judged sentences from the corpus may have ramifications for a system's evaluation as well as for the re-usability of the TST test collection. For example, if sentence s is an exact duplicate of a sentence p in the judged set, a system would neither be rewarded nor penalized for returning s, since s is not in the judged set of sentences. This may lead to an unfair evaluation for the system.

In this work we investigate the effect of expanding the set of judged sentences with their *exact* duplicates from the corpus, where, an exact duplicate is a sentence that exactly matches the text of the judged sentence for each character in the sentence string. We find that:

- There exist an extremely large number of exact duplicate sentences in the corpus (Section 2.2).
- Adding duplicates to the judged set of sentences, does not change relative ranking of systems with respect to Kendall's τ (Section 3.1).
- System performance is affected when the judged set of sentences is expanded with duplicates (Section 3.2), with 13 of 28 submitted runs showing statistically significant changes for a paired t-test with p-value ≤ 0.05.

2. OBSERVATIONS ON DUPLICATES

We briefly describe the original set of judged sentences of the SUS task and set a context for subsequent sections.

2.1 Original Judged Sentences

Each topic in the SUS task corresponds to an event (of type earthquake, storm, accident, bombing or shooting) that

occurred within the corpus duration. The query for the topic included query terms, and query start and end times. The query was considered active for the time interval between query start and end times (the query duration, typically 240 hours).

The assessors initially identified a set of nuggets (short snippets of text containing information relevant to the topic), from the edit history of the Wikipedia[1] article for an event. A pool of sentences for evaluation was created by sampling 60 updates per topic per submitted run with highest confidence scores (a confidence score was required for each update in a run). Some close duplicates were identified from this initial sample which allowed more updates to be included in the pool [2]. This created a pool of sentences totalling 9,113 for 9 topics. A pooled sentence was matched against nuggets and was considered relevant if it contained a nugget.

2.2 Expanded Set of Judged Sentences

Given the original set of judged sentences, for each topic, for each sentence in the judged set, we found the exact duplicates within the query duration of the topic, from the Stream corpus. For each duplicate found, we added it to the original judged set and mapped its identifier to the identifier of its prototype. Table 1 lists the number of judged sentences, the number of duplicates known to exist within the original judged set, the number of sentences for which the duplicates were found within the query duration. Also shown are the number of relevant sentences in the original judged set for each topic, and the number of relevant sentences for which duplicates were found in the query duration from the corpus.

We see that the original judged set expands nearly a 1000 times from 9,113 to 9,034,179, when duplicates are added. However, the number of relevant duplicates found is extremely less at 97,256 (about 10 times the size of the original set). Table 2 shows the top occurring duplicate sentences. In fact, the top 3 most occurring duplicate sentences account for 67% of the total duplicates found. We observed that most of the duplicates that occur with high frequency are duplicates of non relevant sentences. They appear to be boilerplate sentences found in web-site navigation menus or at the end of news articles. In contrast the most frequent relevant duplicate, "National Hurricane Center in Miami said Isaac became a Category 1 hurricane Tuesday with winds of 75 mph.", was found to occur just 5,403 times, in the query duration for topic 5 ("Hurricane Isaac").

We feel that the high number of duplicates found overall could be because of news/web syndication services. News wire documents form the second largest component of documents in the Stream corpus [1], with social documents (blogs and forums) forming the bulk of the corpus, and documents sourced from links submitted to `bitly.com` forming the remainder. One would expect a high number of news/web articles to be generated after the occurrence of a catastrophic event.

3. EFFECT OF ADDING DUPLICATES TO THE ORIGINAL SET OF JUDGEMENTS

The TST introduces new measures to evaluate temporal summarization. The measures are analogous to precision/gain and recall and they are also designed to account for *latency* and *verbosity* of updates (sentences). Participant systems are tasked to return a timestamp along with each update about an event. Latency discounts are applied to sentences that are returned later than the first known occurrence of the nugget of information that they contain. The nuggets were identified, and their time of first occurrence noted, by the track's assessors, using the edit histories of the Wikipedia articles for the topics (Section 2.1). Verbosity discounts are applied based on the length of the returned sentences. Longer sentences are penalized more than shorter sentences by the verbosity discount, which essentially forms an aspect of "user friendliness" for a system.

The track introduces two precision-like measures, *Expected Latency Gain* (E[LG]) and *Expected Gain* (E[G]), as well as two recall-like measures, *Latency Comprehensiveness* (LC) and *Comprehensiveness* (C). The E[LG] and E[G] measures use the relevance score (0/1 for binary relevance) as a measure of gain and potentially discount the gain for update-latency. The C and LC measures attempt to capture how well a system performed at returning all identified nuggets and how quickly it emitted updates containing these nuggets from the canonical time of first occurrence for the nuggets.

The track coordinators consider E[LG] and LC to be the official metrics of the track and we report our analyses for these metrics only. Detailed descriptions of all metrics can be found in the 2013 Temporal Summarization Track Overiew [2]. We note here that in the track's evaluation framework, a system returning sentences not present in the judged set of sentences is neither penalized nor rewarded. On the other hand, returning duplicate sentences present in the judged set will result in the verbosity discount being applied for each.

3.1 Effect on Systems' Ranking

Table 3 provides Kendall's τ correlation between the rankings given by the original judged and expanded set for each of the four task measures. We can see that the correlation tends to be high indicating that relative performance is typically maintained regardless of assessment pool. This is consistent with other works conducting similar research [7, 10, 11, 3, 9]. Accordingly, we can see that not including all *exact* duplicates in the judged sentences was a reasonable and effective method of relative system performance evaluation. However, there may still be benefit to expanding the judged set because there are changes to the absolute performance scores of the systems (Section 3.2) which may be indicative of a change in the user experience.

3.2 Effect on System Performance

Overall, 13 submitted runs showed a statistically significant change in E[LG] and 12 submitted runs showed a statistically significant change in LC, for a paired t-test with p-value ≤ 0.05. The average difference across topics (and standard deviation) between the original judged set and the expanded set are presented in Tables 4 and 5 for E[LG] and LC, respectively. Runs for which there was no difference in score are not listed in the tables. Runs are listed in sorted order based upon the run names.

There exist several runs with no E[LG] change which is primarily due to the fact that they do not contain any newly identified duplicates and so would not be penalized for them. The majority of the the other runs, do experience a general decrease in the performance with respect to the expanded

[1]http://www.wikipedia.org/

Topic	Original Judged Set of Sentences					Expanded Judged Set	
	#sentences	#known duplicates	#with duplicates found in corpus	#relevant sentences	#relevant with dup.s in corpus	#sentences	#relevant sentences
1	779	100	309	431	146	833794	1445
2	912	180	474	381	202	2241589	6301
3	762	112	494	211	154	552145	25199
4	1463	276	946	410	260	264474	22587
5	1069	0	689	82	63	821897	17043
6	1517	187	905	493	270	730296	18661
8	1128	205	609	172	102	1057643	1741
9	873	172	423	168	97	2430455	2384
10	610	89	338	287	143	101886	1895
Total	9113	1321	5187	2635	1437	9034179	97256

Table 1: The number of sentences in the Original judged set vs. the Expanded set

Frequency	Topics	Duplicate Sentence
3376809	2,9,1	All rights reserved./All rights reserved
2013684	2,9,8	Yahoo!
673876	3	New User ?
529085	5	3.
294662	8	This material may not be published, broadcast, rewritten or redistributed.
. . .		
166557	6,9	U.S.
111503	8,9	Register Sign In Help New Firefox ®16 Optimised for Yahoo! Notifications Help Mail My Y!
. . .		
81985	6	Join Here .

Table 2: Examples of Duplicate Sentences with high number of occurrences across all topics

Judged Sentences expanded with	Kendall's τ for Ranking Metric E[LG]	E[G]	LC	C
exact duplicates	0.899	0.894	0.942	0.937
lowercase duplicates	0.899	0.894	0.942	0.937

Table 3: Rank correlation between Original Judged sentences vs Expanded set, for TST measures

set; though, the affect on such runs is not consistent across topics. Of particular note is run 8 in Table 4, which has a general *increase* in performance indicating that run 8 did return relevant sentences which were not in the original judged set, but were duplicates of relevant sentences from the original set. This increase was found not to be statistically significant with a p-value > 0.05 for a paired t-test. However, with more topics, we may find more statistically significant positive improvements [8].

Furthermore, we can see that the duplicate detection in the expanded set does not hurt but improves LC performance on average. This makes sense since systems may have returned relevant sentences duplicate to those in the original set of judgements. By expanding the original judged set with exact duplicates, we argue that a more accurate assessment of absolute performance is being achieved since runs are now being rewarded or penalized for new sentences which were not present in the original set.

3.3 Variations on Duplicate Detection

We also tried duplicate detection with simple transformations (to ensure minimal information loss) like *lowercase*-ing, *whitespace*-collapsing (reducing sequences of whitespace to a single space) and *whitelower* (lowercase + whitespace). The

Run	μ (σ)	Run	μ (σ)
1	0.0112† (0.0134)	2	0.0097† (0.0100)
4	0.0107† (0.0138)	8	-0.0037 (0.0089)
9	0.0391 (0.0559)	10	0.0363† (0.0388)
11	0.0006 (0.0030)	12	0.0001 (0.0028)
13	0.0014 (0.0019)	14	0.0013† (0.0016)
15	0.0007 (0.0020)	16	0.0014 (0.0019)
18	0.0151† (0.0163)	19	0.0160† (0.0181)
20	0.0162† (0.0152)	21	0.0171† (0.0192)
22	0.0008 (0.0016)	23	0.0008 (0.0016)
24	0.0054† (0.0067)	25	0.0054† (0.0061)
26	0.0052† (0.0048)	27	0.0036† (0.0030)
28	0.0007 (0.0010)		

$\dagger$ denotes a p-value ≤ 0.05 for a paired t-test

Table 4: Average Difference and Standard Deviation of E[LG] between the Original Set and the Expanded Set of judged sentences

lowercase transformation is a common normalization technique employed in search engine indexing, and may introduce errors (e.g. US, the country, vs us, the pronoun). It did increase the total number of duplicates found to 10,872,223 but found only 44 new duplicates for relevant updates. Both *whitespace* transformations did not produce different sets of sentences than their basis transformations.

The Kendall's τ between the original judged set and the expanded set, using exact duplicates and lowercase duplicates, are identical (Table 3), due to the fact they both produced identical rankings, and in fact scores, for all systems. The *lowercase* transformation found additional duplicates overall but very few relevant duplicates and hence there are insignificant changes in the scores when averaged across topics and runs. With more topics and more participant systems, we might expect to see the effect of such transformations to be more pronounced.

4. DISCUSSION

We believe that much of the negative effect of the original judged set expansion (on the gain-based measures) would be subsumed, in the majority of cases, if the verbosity penalization were applied to all sentences retrieved by a system. Currently, such penalization only occurs if a retrieved sentence is also present in the judged set. While beneficial for evaluating system performance for finding relevant sentences, ignoring unjudged sentences does not accurately reflect the user experience. One would expect a large difference in performance scores between submitting 1,000 sentences and 1

Run	μ (σ)	Run	μ (σ)
1	-0.0594† (0.0446)	2	-0.0594† (0.0446)
4	-0.0809† (0.0571)	8	-0.0138 (0.0183)
9	-0.0325† (0.0402)	10	-0.0408 (0.0611)
11	-0.1247† (0.0550)	12	-0.1324† (0.0557)
18	-0.0915† (0.0604)	19	-0.1045† (0.0586)
20	-0.0669† (0.0551)	21	-0.0734† (0.0526)
24	-0.0192† (0.0247)	25	-0.0297† (0.0328)
26	-0.0038 (0.0061)		

$\dagger$ denotes a p-value ≤ 0.05 for a paired t-test

Table 5: Average Difference and Standard Deviation of LC between the Original Set and the Expanded Set of judged sentences

million sentences; a difference which has the potential to overwhelm the user. The official set of judgements averages around 1,000 sentences per topic which results in an upper limit on verbosity penalization.

However, the current evaluation methology is consistent with that of [7], who found that removing unjudged documents from evaluation of ranked lists works well and does not affect the relative system performance. Indeed, the new TST metrics are stable for measuring relative system performance, even after processing an expanded set of judgements that is 1000 times the size of the original.

Nugget-based evaluation [4, 6] - where identified relevant material is representative of relevance - is aimed towards automatic identification of nuggets in the whole collection. However, tracking duplicates of the retrievable unit (e.g. documents, sentences) may be useful depending on the evaluation metrics for the specific task at hand (such as Temporal Summarization).

5. FUTURE WORKS

As an immediate future work, we plan to investigate the effect of including unjudged updates for verbosity discounts. Accounting for unjudged sentences in runs may not be a straightforward task due to the potential quantity of them. A simple but potentially inefficient mechanism would be to store information (e.g. length, timestamp, duplicates) about every sentence in the corpus which would facilitate applying the necessary discount. Alternatively, it may be possible to produce an estimate of unjudged sentence lengths. This may require the use of some form of sampling and the use of inclusion probabilities (e.g. [5]). Determining a reasonable method for applying verbosity penalization to unjudged sentences is an area of research that we intend to pursue.

The TST at TREC 2013 had only 9 topics. As per [8, 12], even though there are statistically significant changes in systems' performance, we cannot currently presume that the effects would be reproducible for a different/larger set of topics. We definitely need to test for the effect of duplicates on more number of topics.

6. CONCLUSIONS

We experimented with expanding the set of judged sentences with exact duplicates from the corpus and investigated its effects on the evaluation of temporal summarization. We found that adding exact duplicate sentences to the set of relevance judgements, does not affect relative ordering of temporal summarization systems. It does however, induce a change in the performance scores of the systems. 13 out of 28 systems that participated in the Temporal Summarization Track at TREC 2013 experienced a statistically significant change in performance scores with respect to the track's metrics. With more topics from the TREC 2014 version of the track, we expect to get a more accurate estimate of changes in performance when evaluating with a large number of duplicates. Expansion of relevance judgements with exact duplicates is simple and not only does it help produce more accurate performance scores but also potentially aids in reusability of the test collection for the development of new temporal summarization systems.

7. ACKNOWLEDGMENTS

We thank Rakesh Guttikonda for helpful discussions and ideas regarding this work. This work was made possible by the facilities of SHARCNET (www.sharcnet.ca) and Compute/Calcul Canada, and was supported in part by NSERC, and in part by the Google Founders Grant, and in part by the University of Waterloo.

8. REFERENCES

[1] KBA Stream Corpus 2013. `http://trec-kba.org/kba-stream-corpus-2013.shtml`.

[2] J. Aslam, F. Diaz, M. Ekstrand-Abueg, V. Pavlu, and T. Sakai. TREC 2013 Temporal Summarization. In *TREC*, 2013.

[3] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR*, pages 25–32, 2004.

[4] G. Marton and A. Radul. Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. In *HLT-NAACL*, pages 375–382, 2006.

[5] V. Pavlu and J. Aslam. A practical sampling strategy for efficient retrieval evaluation, Technical Report, College of Computer and Information Science, Northeastern University. 2007.

[6] V. Pavlu, S. Rajput, P. B. Golbus, and J. A. Aslam. IR system evaluation using nugget-based test collections. In *WSDM*, pages 393–402, 2012.

[7] T. Sakai and N. Kando. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, 11(5):447–470, 2008.

[8] M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *SIGIR*, pages 162–169, 2005.

[9] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *SIGIR*, pages 66–73, 2001.

[10] A. Trotman and D. Jenkinson. IR evaluation using multiple assessors per topic. *ADCS*, 2007.

[11] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *SIGIR*, pages 315–323, 1998.

[12] E. M. Voorhees and C. Buckley. The effect of topic set size on retrieval experiment error. In *SIGIR*, pages 316–323, 2002.