Mapping Readers' Comments Back to Newspaper Articles

Asif Salekin, Md Anindya Prodhan, Muhammad Nur Yanhaona
University of Virginia
Email: {as3df, mtp5cx, mny9md}@virginia.edu

***Abstract*—Publishing readers' comments has long been an integral part of maintaining integrity in journalism. The evolution of electronic newspapers has greatly enhanced the scope and limit for these comments. So much so that now it is common to have hundreds of comments on popular issues in popular newspapers. The sheer volume of comments, however, introduces new challenges. For general readers it has become nearly impossible to go through all these comments. At the same time, for the newspaper editor it has become difficult highlighting discussions most relevant to an article.**

In this project we worked on automatically filtering and mapping comments that are most relevant to a newspaper article or a portion of the article using information retrieval techniques. To be able to do that we analyse a newspaper article and construct query language models for terms that occurred in its passages. Then we generate queries from those passages and match them against readers' comments treating the latter as candidate relevant documents.

This report describes the logic and implementation of our solution, presents the first set of results we got from analysing articles from two online newspapers, and discusses the problems we faced and possible future research directions.

I. Introduction

Publishing readers' comments on newspaper articles has been a long-standing tradition in good journalism since the early days of newspaper. Sometimes such comments convey readers' feelings about the original article, sometimes they provide complementary information that help other readers to get a holistic understanding of the discussed topics, sometimes they expose errors, omissions, or subtle bias on the part of the reporter, and so on. Publication of readers' comments has, therefore, become an integral part of transparency in journalism.

During the era of printed newspapers, these comments used to appear as follow-up discussions in days following the date of original article publishing. Because of space constraints, follow-up comments tend to be limited to only a few interesting news and furthermore limited in their numbers. The advent of electronic newspapers, however, has fundamentally altered that pre-existing culture. Currently, comments can be written on every published articles, by anyone interested, and there is no limits in the number of comments either. Now it is quite common to see several hundreds of comments on popular news articles of papers like New York Times, BBC News, etc.

In regards to fact finding and ensuring transparency this is an encouraging trends. The sheer volume of such comments, however, introduces several important challenges. On one hand, it can be overwhelming or completely unreasonable for a common reader to go through all the comments. On the other hand, for the editor it is difficult to highlight comments that may be most relevant to the original article or purge comments that are simply vile. To the best of our knowledge, so far little has been tried to aid both parties in this regard by automatic highlighting, filtering, and classification of readers' comments.

Although our broad objective for this research was to address all of the above, as part of the class project we focus on filtering relevant comments only. The central idea behind our solution is to view an article title as a query and the article itself as a feedback document for constructing the language model for the query intended by the title. Then to judge relevance of comments to the article or passages in the article, we automatically generate queries from the constructed model and rank comments against them.

For our experiment we crawled about 1000 articles each having at least 20 comments from Alzajeera News and collected 780 more similarly comment-rich articles from Yahoo-News gathered by authors of [3]. Language models for articles are uni-gram models that we generated using our own algorithm that relies on a notion of term significance which depends on both frequency and proximity of terms related to terms in the article title. Then a corpus based smoothing is applied to determine final term significance rankings. In short, a model in our case is not just a collection of terms; rather a probability estimate is associated with each term regarding its chance to be included in a query representing the information need served by the article or a part of that article.

Ranking of comments against generated queries are done using Okapi BM25 [7]. We varied different parameters of the ranking function as well as of our significance calculation to identify the best configuration. The ground truth for comments' relevance measurement is gathered through human evaluation of a small random sample of news from our crawled database.

The results are mixed. We identified several challenges regarding improving our solution further. Some problems were implementation specific issues that arose from simplifications we have to made to save time. Nevertheless, there are reasons to be optimistic and we hope to enhance our solution in the future.

The rest of the report is organized as follows. Section II discusses some related work; Section III discusses how we model our domain as an information retrieval problem; Section IV elaborates on aspects of our solution; Section V briefs on

the actual implementation; Section VI presents the results of the experiments done on a small sample of articles; Section VII ponders over possible routes for future work; finally, Section VIII concludes the report.

II. Related Work

There have been a few works done on matching comments with News Article segments. Sil et. al. [8] used supervised and unsupervised techniques to create structural classifier to match comments with News article segments. This paper used explicit semantic analysis and co-reference features to represent the text in article and text and showed that accuracy of discriminative approaches depend largely on effective feature selection. They used prior data to detect article segment and comment match in articles of a related topic.

To detect article segment and comment matching we need to extract topics from a News article using topic modeling. Several works have been done in topic modeling. MG-LDA [2], [9] is used to model local topics in a text. However, In a news article every segment may not be related to a comment. Also in this model, local topics scatter across the corpus which is not often the case in news articles. Hence, MG-LDA is not appropriate for our system. Also, Corr-LDA [9] is a topic model to understand correspondence. Since, Corr-LDA work with single vector model, a specific comment on a small segment of an article can show small correlation with the article using this model. To overcome this limitation Das et. al. [3] developed a correspondence topic model (SCTM) that uses multiple topic vectors. Hence, for each comment-article segment pair there would be number of topic vectors, which would enable comments to match with more correspondence segments of article. In [5] the authors focused on comments summarization. This paper selects few most representative comments from comment cluster for an article. They used Master-Slave Topic model (MSTM) and Extended Master-Slave Topic model (EXTM) where News articles is treated as master and comment text as slave. Using this topic models they cluster the comments based on their topics and rank the most representative comments from each comment clusters.

III. Problem Formulation

This section describes how we model the problem of finding relevant readers' comments as an information retrieval problem.

We view components of a news comprising the article and associated comments serve to satisfy a single information need that can be thought of as represented by a query that is similar to the title of the article. In our formulation, the comments are of interest only because the original article is by itself insufficient in conveying all information necessary to fully satisfy a reader's information need.

Given this formation, the comments of interest are those comments that address topics covered in the article but augment article's author arguments. These comments broadly fall into to categories.

1) Comments that support article author's arguments with additional information and insights, and
2) Comments that contradict the article with contrasting evidences and insights

Given the unregulated nature of comments publishing, the comments section is cluttered with several other kinds of comments such as (as we observed)

3) Comments that express nothing but the sentiment of a particular reader about discussed topics
4) Arguments exchanging among readers about each other's comments
5) Elaborate discussion on some offshoot topic that has minute relevance to the article itself

An abundance of comments of latter categories makes it difficult to for an unbiased reader to identify and consequently satisfy his partially fulfilled information need. Although distinguishing between the former and latter kinds of comments is often a subjective decision, henceforth difficult, we hold that comments of former categories are more likely to have the following characteristic that will distinguish them from the latter.

There is significantly larger overlapping between principal issues discussed in an article and in a relevant comment than between the article and a non-relevant comment.

Given this assumption, the problem of filtering relevant comments can be designed as a standard information retrieval problem that has two parts.

1) Construct query language models from passages of an article
2) Generate queries from these models and rank comments against individual queries

Here we consider passage specific models as for articles having many passages, there may be jump in the topics been discussed. Consequently, a comment may also have passage specific complementary discussions. In the general case, the article presents a composite language model that is a mixture of the models of individual passages; and in the degenerative base case, it represents a single passage itself.

Before we dive into a discussion about our implementation, we should emphasize that our language model should not be equate to a topic model similar to some related work discussed previously. Rather, it is intended to generate queries. To clarify the difference with an example if "Obama decided to go alone in the Immigration reform issue" is a title of a news article then from the topic modelling perspective "Immigration Reform" is the primary – and arguably the sole – topic but from the query generation perspective "Obama's Decision" alongside "Immigration Reform" should construe the crux of the discussion.

IV. Design

The center piece of our solution is the algorithm for developing query language models for individual passages of a newspaper article. Next we discuss the algorithm and. Afterwords we discuss how these models can be used to generate queries that would be used for ranking evaluation.

A. Query Model Generation

Given that we treat an article title to be equivalent to a query representing readers' information needs, we have a mechanism to interpret the significance of the content of the article. We believe a reasonable interpretation for any article is that each passage provides a partial information about the issue underlying the title and each sentence within a passage works on building up the arguments of its passage. To summarize, their is no term in the article that is not relevant to readers' information need.

The relevance of these terms are, however, different and based on several parameters given below.

1) Term Frequency
2) Significance Level: minimum distance of a term from terms occurring in the title
3) Relevance Weight: the number of co-occurrences of a term with other significant terms in the article
4) Smoothing Factor: general likelihood of the term within the corpus

Among these the first and the last are self explanatory. So we focus on explaining the remaining two.

To calculate both significance level and relevance weight of terms we construct a graph $G = (V, E)$ for each passage P where the vertex set $V = \{w_i | w_i \in P\}$ and the edge set $E = \{(w_i, w_j) | w_i, w_j \in s_k, s_k \in P\}$. That is there is an edge between two words if they occur in the same sentence in anywhere in the passage.

Then the formula for the significance level of a word w_i is as follows,

$$\begin{aligned} &s_i = 1 : w_i \in title \\ &s_i = 1 + distance_{min}(w_i, w_j) | w_j \in title) : w_i \notin title \end{aligned} \tag{1}$$

Here distance is measured as the number of edges between the concerned words pair.

The formula for the relevance weight of a word w_i is as follows,

$$rWeight_i = \sum_{\substack{(w_i, w_j) \in E \\ s_j < s_i}} max\big(0, 1 - \log(s_j)\big) \tag{2}$$

If we reason about aforementioned formulas, we see that terms that occurred at close proximity of any terms in the title have more significance than terms that appeared further. Then terms that occurred more often and with more significant terms have more relevance weight than terms that occurred less often.

The rationale for this treatment lies in the assumption about the logical flow of arguments that we expect to present in a newspaper article. Sentences making statements about elements of the title should compose the main agenda. Then there will be sentences providing supporting evidences and so on. Consequently, terms occurring in different kind of sentences should have varying degrees of importance in the constructed model.

Note that in Equation 2 only terms with lesser significance, w_i, get the benefit of co-occurrence – not the term with greater significance, w_j. This apparent lacking is balanced by the frequency weight of individual term which has the following formula.

$$fWeight_i = tf \times \frac{s_i + k}{s_i(k+1)} \tag{3}$$

Here k is a scaling parameter used to exponentially decrease the importance of a term's occurrences with increase of its significance level. The likelihood of a term to contribute to a query representing the information need satisfied by the concerned passage of the article is calculated as a mixture of normalized relevance and frequency weight as captured by the following equation.

$$weight_i = \alpha \times rWeight_i + (1 - \alpha) \times fWeight_i \tag{4}$$

Equation 4 is combined with the collection probability of a terms $p(w_i, C)$ using the Jelinek-Mercer interpolated smoothing method [4] to give the final smoothed probability as follows.

$$p_q(w_i) = (1 - \lambda) \times weight_i + \lambda \times p(w_i, C) \tag{5}$$

B. Ranking Evaluation

If we remember the discussion of Section III, we see the criteria for judging relevance of a comment to the discussion presented in the article or a passage of the article is the degree of overlapping of comment's language model with that of the former. Here *overlapping* deserves careful attention.

Note that we expect from a relevant comment to complement the discussion of of the article with supporting or contradictory arguments. We do not expect the comment to say the same thing the article does. So there should be significant overlapping in principal terms of the comment and the article, but there should be also considerable distinction in their auxiliary terms. Therefore, ideally the goal of comments ranking should not be to maximize overlapping. Rather it should be to balance similarity in principal terms with dissimilarity in auxiliary terms.

If we define a cut-off threshold for partitioning the query language model of a passage P into a principal θ_{pri} and an auxiliary part θ_{aux} then the relevance judgement for a comment D with respect to the passage can be expressed as a harmonic mean as follows.

$$relevance(P, D) = \frac{1}{\frac{\beta}{sim(\theta_{pri}, \theta_d)} - \frac{1-\beta}{sim(\theta_{aux}, \theta_d)}} \tag{6}$$

This is a generic equation and any standard metric – for example KL-divergence – can be substituted as the similarity measure to calculate the relevance in the above.

A reasonable simplification to Equation 6 is to consider only the similarity between the comment model and passage model in the principal terms. This is practical as it is highly unlikely that a comment will only reiterate the argument presented in an article. Hence we can ignore the dissimilarity term and rank comments based on the following formula.

$$rank(P, D) \sim sim(\theta_{pri}, \theta_d) \tag{7}$$

V. Implementation

We made a few modifications to the solution described in the previous section for practical considerations and – particularly – to save time. As we will discuss actual implementation in this section, these deviations will become apparent.

We use Apache OpenNLP parser [1] to extract sentences from the article then to determine the parts of speech of all words in the article. Then we keep only the nouns and verbs for later analysis. We make this decision as our observation indicates that conjunctions, prepositions, and adverbs usually have little to contribute to the central arguments of an article; and our collection of articles is not large enough to discard their influence applying smoothing alone. We are uncertain about the import of adjectives and currently leave them out. Our he system is configurable though and different parts of speech can be enabled and disabled as deemed useful.

For passage identification, we adopt the simple strategy that a paragraph with a minimum n sentences portrays a holistic argument/evidence and a sequence of m such paragraphs form a passage. If a paragraph is less than n sentences than we consider it automatically included in the current passage without increasing its paragraph count. Our plan was to try different combinations of n and m and determine what performs best. To our surprise, the vast majority of news articles seem to have paragraphs that are mostly one or two sentences long. Hence in most cases, we get only one passage, consequently one query model, per article. There are exceptions though.

The query language model is implemented exactly according to the formulas given in the previous section using our own data structure and algorithms. The for pairwise term association graph is implemented using two map data structures using words tagged and filtered by during parsing phase. Then frequency and relevance weights are measured concurrently by traversing the graph starting from the title words. Finally they are normalized. The time and space complexities for the algorithm are $\mathcal{O}(|E|)$ and $\mathcal{O}(|V| + |E|)$ respectively.

Computing the collection model is the costliest part because we have to analyse all articles. That is however computed only once and the model is been stored in a file. Before ranking the model is been loaded in memory and used as many times as required.

The biggest sacrifice that we have to make is about ranking evaluation. This is regrettably because time constraint due to our engagement in other research. We use Apache Lucene [6] for indexing comments of an article and ranking them against our generated query model. Since we could not manage time to study and implement a language model based ranking process over Lucene as that would be required by Equation 7, we directly generate a query from the model selecting the terms with highest weight and rank comments against this query using Okapi BM25 [7].

This adjustment has the shortcoming that the probabilities for principal terms available in the query model have been used during query generation, but the advantage of their presence has not been taken during ranking comments as the ranking function treated all query terms as equal. We suspect this has a significant adverse effect in our solution's performance. We hope to correct this problem in the future – depending on the feedback we may get regarding the future prospect of this work.

VI. Evaluation

VII. Future Work

VIII. Conclusion

References

[1] Apache opennlp. https://opennlp.apache.org/.

[2] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 127–134, New York, NY, USA, 2003. ACM.

[3] M. K. Das, T. Bansal, and C. Bhattacharyya. Going beyond corr-lda for detecting specific comments on news & blogs. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 483–492, New York, NY, USA, 2014. ACM.

[4] F. Jelinek and R. L. Mercer. Interpolated Estimation of Markov Source Parameters from Sparse Data. 1980.

[5] Z. Ma, A. Sun, Q. Yuan, and G. Cong. Topic-driven reader comments summarization. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 265–274, New York, NY, USA, 2012. ACM.

[6] M. McCandless, E. Hatcher, and O. Gospodnetic. *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA, 2010.

[7] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. pages 109–126, 1996.

[8] D. K. Sil, S. H. Sengamedu, and C. Bhattacharyya. Supervised matching of comments with news article segments. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2125–2128, New York, NY, USA, 2011. ACM.

[9] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 111–120, New York, NY, USA, 2008. ACM.