Comment-to-Article Linking in the Online News Domain

Ahmet Aker, Emina Kurtic, Mark Hepple, Rob Gaizauskas, Giuseppe Di Fabbrizio
University of Sheffield
ahmet.aker,e.kurtic,m.r.hepple,r.gaizauskas@sheffield.ac.uk,difabbrizio@gmail.com

Abstract

Online commenting to news articles provides a communication channel between media professionals and readers offering a crucial tool for opinion exchange and freedom of expression. Currently, comments are detached from the news article and thus removed from the context that they were written for. In this work, we propose a method to connect readers' comments to the news article segments they refer to. We use similarity features to link comments to relevant article segments and evaluate both word-based and term-based vector spaces. Our results are comparable to state-of-the-art topic modeling techniques when used for linking tasks. We demonstrate that article segments and comments representation are relevant to linking accuracy since we achieve better performances when similarity features are computed using similarity between terms rather than words.

1 Introduction

User comments on news articles and other online content provide a communication channel between journalists and their audience, which has previously replaced prevalent one-way reporting from journalists to their readers. Therefore, several user groups in media business now rely on online commenting to build and maintain their reputation and broaden their readers and customer base. To achieve this, however, it is essential to foster high quality discussions in online commenting forums because quality and tone of comments are shown to influence the readers' attitudes to online news content (Anderson et al., 2013; Diakopoulos and Naaman, 2011; Santana, 2014).

In the present set up of online forums, comments are difficult to organize, read and engage with, which affects the quality of discussion and the usefulness of comments for the interested parties. One problem with comments in their current form is their detachment from the original article. Placed at the end of the article without clear reference to the parts of the article that triggered them, comments are hard to put into the context from which they originated, and this makes them difficult to interpret and evaluate. Comment-article linking is also necessary in more complex systems for information extraction from comments such as comment summarization (Hu et al., 2008; Khabiri et al., 2011; Lu et al., 2009; Ma et al., 2012; Llewellyn et al., 2014). Such systems rely on identifying relevant comments and those that link to the articles are good candidates.

In this paper we report the results of our experiments in comment-article linking. Specifically, the task is to bring together readers' comments with online news article segments that comments refer to. We compare the performance of text similarity measures to that of more elaborate topic modeling methods such as the ones proposed by Sil et al. (2011) and Das et al. (2014) and demonstrate that comparable linking results can be achieved by simpler text similarity methods.

Given the weak lexical overlap between comments and source articles, we also investigate the effect of alternative representations of comments and news article texts on the results of comment-article linking with similarity metrics. We analyze the performance of the similarity method using terms, i.e., sequences of words which have all a meaning in a domain (de Bessé et al., 1997), and show that term based similarity linking outperforms similarity linking based on words.

The paper starts with defining the linking task and the pre-processing steps we perform on the article and comments (Section 2). Then we provide the description of our linking approach (Section 3). In Section 4 we report our experimental results. We summarize the paper in Section 5.

2 Task and Pre-processing

2.1 The task

For the linking task we assume a news article A is divided into n segments $S(A) = s_1, ..., s_n$. The article A is also associated with a set of comments $C(A) = c_1, ..., c_l$. The task is to link comments $c \in C(A)$ with article segments $s \in S(A)$. We express the strength of link between a comment c and an article segment s as their linking score ($Score$). A comment c and an article segment s are linked if and only if their $Score$ exceeds a threshold, which we experimentally optimized. $Score$ has the range $[0, 1]$, 0 indicating no linking and 1 defining a strong link.

2.2 Pre-processing

First, we split the news article into segments. To compare results with existing data sets and exist-

Proceedings of the SIGDIAL 2015 Conference, pages 245–249,
Prague, Czech Republic, 2-4 September 2015.

ing contributions, we comply with segmentation approaches used in previous work (Sil et al., 2011; Das et al., 2014). We treat each article sentence as a segment and group each comment into a single unit regardless of the number of sentences it contains. Then each sentence-comment pair is preprocessed before it is analyzed for linking. The example in Table 2.2 illustrates the outputs of the pre-processing pipeline.

The pre-processing includes tokenization[1] and lemmatization (step 2) in in Table 2.2, where an original article sentence is shown in step 1)). Next, we use either words with stop-word removal (step 3)) or terms (shown in 4) where each term is split by a semicolon) to represent the article sentence and also each comment. Terms are extracted using the freely available term extraction tool *Tilde's Wrapper System for CollTerm* (TWSC)[2] (Pinnis et al., 2012). We also record named entities (NEs) (shown in 5)) extracted from either article segments or comments.

1	**Original article sentence:** *An Afghan policewoman walked into a high-security compound in Kabul Monday and killed an American contractor with a single bullet to the chest, the first such shooting by a woman in a spate of insider attacks by Afghans against their foreign allies.*
2	**After tokenization and lemmatization:** *an afghan policewoman walk into a high - security compound in kabul monday and kill an american contractor with a single bullet to the chest , the first such shooting by a woman in a spate of insider attack by afghan against their foreign allies .*
3	**When words are used:** *afghan, policewoman, walk, high, security, compound, kabul, monday, kill, american, contractor, single, bullet, chest, shooting, woman, spate, insider, attack, afghan, foreign, allies*
4	**When terms are used:** *shooting by a woman;woman in a spate; spate of insider; compound in kabul; kabul monday; insider attack; afghan policewoman; american contractor; single bullet; security compound; foreign allies; policewoman; security; compound; contractor; bullet; chest; shooting; woman; spate; insider; attack; allies; afghan; kabul; monday*
5	**Extracted NEs:** *Kabul*

Table 1: Text pre-processing pipeline example.

3 Method

This work investigates a simple method for linking comments and news article sentences using a linear combination of similarity scores as computed through a number of different similarity metrics (features). However, some comments directly quote article segments verbatim, therefore explicitly linking comments to article segments. To account for this, we consider a comment and an article sentence linked if their quotation score ($quoteScore$) exceeds a threshold. Otherwise, a similarity score is computed and articles are linked if their similarity score is above a threshold. The following paragraphs describe how features and thresholds are computed.

Each metric is computed based on the comment $c \in C(A)$ and a segment $s \in S(A)$ as input. We pair every segment from $S(A)$ with every comment from $C(A)$. With this set up we are able to link one-to-many comments with one segment and also one-to-many segments with a particular comment, which implements an *n* to *m* comment-segment linking schema.

3.1 Quotation Based Linking

We link all comments including quotes to the article sentences they quote. To determine whether a segment is quoted in the comment, we compute $quoteScore = \text{len}(quote)/\text{len}(S)$ with *len* [3]. *len* returns the number of words of the given input and *quote* is a place holder for consecutive news article words found in the same order within the comment. If the $quoteScore$ exceeds an experimentally set threshold of 0.5 (50% of consecutive article segment words are found in the same order within the comment), then the segment is regarded as quoted in the comment, the comment-segment pair is linked, their linking $Score$ is set to $quoteScore$ and no further linking features are considered. However, qualitative observations on random data portions have shown that only sentences longer than 10 words render meaningful quote scores, so we add this as an additional constraint.

3.2 Similarity Linking

3.2.1 Similarity Feature Extraction

If a comment does not contain a quote as described above, we compute the following features to obtain the value of the similarity score without considering the quote feature:

- **Cosine:** The cosine similarity (Salton and Lesk, 1968) computes the cosine angle between two vectors. We fill the vectors with terms/word frequencies extracted from the article segment/comment.

[1]For shallow analysis we use the OpenNLP tools: https://opennlp.apache.org.

[2]TWSC uses POS-tag grammars to detect word collocations producing NP-like word sequences that we refer to as terms. Terms are extracted from the original version of the sentences, but words in the terms are replaced with their lemmas.

[3]For this feature the original version, i.e., without preprocessing, of article segment and comment are used.

- **Dice:**

$$dice = \frac{2 * \text{len}(I(S,C))}{\text{len}(S) + \text{len}(C)} \quad (1)$$

where $I(S,C)$ is the intersection set between the terms/words in the segment and in the comment. *len* returns the number of entries in the given set.

- **Jaccard:**

$$jaccard = \frac{\text{len}(I(S,C))}{\text{len}(U(S,C))} \quad (2)$$

where $U(S,C)$ is the union set between the terms/words in the segment and comment.

- **NE overlap:**

$$NE_{overlap} = \frac{\text{len}(I(S,C))}{\text{len}(U(S,C))} \quad (3)$$

where $I(S,C)$ is the intersection set between the named entities (NEs) in the segment and in the comment and $U(S,C)$ is the NEs union set.

- **DISCO 1 + DISCO 2:** *DISCO* (DIStributionally similar words using CO-occurrences) assumes words with similar meaning occur in similar context (Kolb, 2009). Using large text collections such as the BNC corpora or Wikipedia, distributional similarity between words is computed by using a simple context window of size ± 3 words for counting co-occurrences. DISCO computes two different similarities between words: *DISCO1* and *DISCO2*. In DISCO1 when two words are directly compared for exact similarity DISCO simply retrieves their word vectors from the large text collections and computes the similarity according to Lin's information theoretic measure (Lin, 1998). DISCO2 compares words based on their sets of distributional similar words.

3.2.2 Computing Similarity Linking Score

Using a linear function, we combine the scores of each of these features (*cosine* to *DISCO*) to produce a final similarity score for a comment-segment pair:

$$Score = \sum_{i=1}^{n} feature_i * weight_i \quad (4)$$

where $weight_i$ is the weight associated with the i^{th} feature. The weights are trained based on linear regression using the Weka package and the training data described in the following section.

3.2.3 Training Data

Obtaining training data requires manual effort and human involvement and is thus very expensive, while resulting in relatively small training data sets. We therefore automatically assemble training data by using comments with article quotes as a training data set. As outlined above, in addition to original comment text, many comments include a brief quotation from the article, therefore directly indicating which article segments have triggered the comments. The set of comments with quotes linked to the article segments they quote are used as our training data.

To gather the training data, we downloaded 3,362 news articles along with their comments from The Guardian news paper web site[4] over a period of two months (June-July 2014). The Guardian provides for each topic (e.g., business, politics, art, etc.) a specific RSS feed URL. We manually collected RSS feeds for the topics: politics, health, education, business, society, media, science, thenorthener, law, world-news, scotland-news, money and environment. Using an in-house tool, we visited the news published through the RSS feeds every 30 minutes, downloaded the article content and also recorded the news URL. Every recorded news URL was re-visited after a week (the time we found sufficient for an article to attract commenters) to obtain its comments. Articles contained between 1 and 6,223 comments, averaging 425.95 (median 231.5) comments per article.

Each article was split into sentences and for each of these sentences (containing at least 10 words) it was determined whether it is quoted in any of the comments as described above. In case the $quoteScore$ was above 0.5 for a sentence-comment pair, the pair was included in the training set. Using this process we have extracted 43,300 sentence-comment pairs to use for training. For each pair, the similarity features listed in Section 3.2.1 were extracted. The $quoteScore$ was used as the expected outcome. We also included 43,300 negative samples into the training data in order to present linear regression with the behavior of the features for wrong sentence-comment links. The negative samples were created by pairing every sentence containing at least 10 words of article X with every comment of article Y. In this way we pair comments with sentences of another article that have not originally triggered the comments. Similar to the positive samples, the quote score was taken as the expected outcome. However, unlike the positive samples, the $quoteScore$ threshold of 0.5 was not applied for the negative samples.

4 Evaluation

4.1 Test Data

In this study, we use the *AT* corpus (Das et al., 2014) to test the above linking method. The AT data set consists of articles with comments downloaded from the technology news website *Ars Technica* (AT). In this data set there are 501 articles. Each article contains between 8 and 132 sentences with an average of 38. Each article has between 2 and 59 linked comments with an average of 6.3. As reported in Das et al. (2014), two annotators mapped comments to article sentences; however, the agreement between annotators cannot be assessed from the available data set due to the lack of double annotations.

[4] http://theguardian.com

Method	Precision	Recall	F1
Metrics$_{term}$	**0.512**	0.292	0.372
Metrics$_{word}$	0.316	0.300	0.310
Metrics$_{termWord}$	0.414	0.310	0.356
SCTM	0.360	0.440	0.390
Corr-LDA	0.010	0.030	0.010

Table 2: Comparison of term/word based similarity metrics on article-comment linking to SCTM and Corr-LDA.

4.2 State-of-the art

The combined quotation and similarity-based linking investigated here is compared to the state-of-the-art SCTM method described in Das et al. (2014). SCTM (Specific Correspondence Topic Model that admits multiple topic vectors per article-comment pair) is an LDA-based topic modeling method that takes into account the multiplicity of topics in comments and articles. Their baseline is *Corr-LDA*, which Das et al. (2014) deem unsuitable since it is restricted to using only a single topic vector per article-comment pair. Evaluation on the same AT test data set allows for a direct comparison of our results to those of SCTM and Corr-LDA. Another recently proposed linking approach is reported in (Sil et al., 2011). However, it does not match the performance of its simple $tf * idf$ based baseline, so we do not consider this method in our evaluations.

4.3 Results

Table 4.3 shows the performance of the automated linking task using quotation and similarity metrics ($Metrics$) on the AT data.[5] The table shows the results for both term and word based representation of article segments (first two rows). Both results were obtained with the experimentally determined $Score >= 0.5$. The results in the table show that representation of article segments and comment texts as terms is superior to the bag-of-words representation for the comment-article linking task as it achieves substantially higher score in precision with a similar recall value. We also combined terms with words by merging the term list with the bag of words and used them to compute the metrics. The results are shown in the 3^{rd} row. Compared to the word only variant, $Metrics_{word}$, we see a substantial improvement in the precision and a slight one in the recall score. However, compared to the term only variant, $Metrics_{term}$, the precision score is still low indicating that terms only are indeed the better choice for representing article segments and comments for the linking task.

The results in Table 4.3 show that the state-of-the-art baseline SCTM outperforms the $Metrics$ regarding the overall F1 score due to higher recall. However, this difference in F1 score is small. The precision of $Metrics_{term}$ based similarity is substantially higher than that of the SCTM method at the expense of recall. Higher precision may be preferable to higher recall for the linking task as including wrong links in order to have higher coverage is noisier and therefore more disturbing for both human and automatic processing of comment-article links than leaving relevant comments unlinked. These results suggest that term based similarity linking is performing almost as well as the SCTM method overall, and if increasing precision over recall is favored for the comment-article linking task, it even could be a preferred method for this task.

[5]Note that the testing data does not contain any comment that quotes an article sentence as specified in our quote feature. This means all the results are achieved through the other features – cosine to Disco features.

5 Conclusion

In this paper we report initial experiments on linking reader comments to the relevant segments in the articles – a task which has multiple applications in organization and retrievability of information from online commenting forums.

Linking between articles and comments implies capturing similarity between a comment and related article segments. In Das et al. (2014) the similarity is defined as similarity in topic. The claim is that multiple topics occurring in a comment and article need to be modeled in order to establish successful links. In this work our aim was to investigate how well known similarity metrics combined with a quotation heuristic perform on the linking task, and how their performance compares to refined topic similarity modeling proposed in previous work. The results showed that the overall performance of combined quote and similarity metrics is comparable to that of topic modeling method despite substantial domain difference between training and testing data sets. The bias of the quote and similarity method is towards precision and in topic modeling towards the recall. We also found that linking using similarity based on terms, i.e., specialized word sequences that have meaning in a domain, achieves better results than linking based on words. This is not surprising given a low lexical overlap between comments and article segments. The fact that terms achieved good results indicates that it is worth exploring further representations that abstract away from lexical items. This will be one of our immediate future studies. Furthermore, we plan to also address the recall problem by investigating clustering methods to group "similar" comments and link these groups instead of the single comments. Finally, we will investigate how the linking task can be used for summarizing news comments.

Acknowledgements

The research leading to these results has received funding from the European Union - Seventh Framework Program (FP7/2007-2013) under grant agreement n610916 SENSEI

Appendix – MultiLing Linking Task

We also participated in the linking task organized by MultiLing 2015. Similar to the task described in Section 2.1 the linking task within MultiLing was to link a comment to an article segment (sentence). However, unlike the task described above the comment was not treated as one unit, but split into sentences. This allowed to link parts of the comment (sentences) to article sentences and leave some out. Although the MultiLing linking task set-up defined this freedom within the comments we continued treating the entire comment as one unit. More precisely, when our linking approach found a link between a sentence in the comment and an article sentence it also linked all the remaining sentences within the comment to the article sentence. The evaluation was performed with English and Italian data.

Each participant was allowed to submit two runs. Our runs differed in how we set a threshold for linking similarity. The first run was set to a lower threshold (i.e. the $Score$ in equation 4 was set to 0.3). Anything below this threshold was not linked. In the second run the threshold was set to 0.5. For English both our runs were considered. However, for Italian there has been some problems in the submission, so that our second run with the threshold 0.5 was not considered.

Our results for English are that using our second run we obtained better results compared to all other 8 system submissions. With this set-up we achieved 89% precision. Our first run (run with the 0.3 threshold) achieved 82% precision. With this score it became the 5th system. For Italian our first run got the 6th position scoring 89% precision. Since our first run also did not perform well on the English data, it is likely that the performance on the Italian data would have been better could the second run be submitted.

References

Ashley A. Anderson, Dominique Brossard, Dietram A. Scheufele, Michael A. Xenos, and Peter Ladwig. 2013. The nasty effect: Online incivility and risk perceptions of emerging technologies. *Journal of Computer-Mediated Communication*.

Mrinal Kanti Das, Trapit Bansal, and Chiranjib Bhattacharyya. 2014. 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*, pages 483–492. ACM.

Bruno de Bessé, Blaise Nkwenti-Azeh, and Juan C. Sager. 1997. Glossary of terms used in terminology. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 4:117–156(39).

Nicholas Diakopoulos and Mor Naaman. 2011. Towards quality discourse in online news comments. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, pages 133–142, New York, NY, USA. ACM.

Meishan Hu, Aixin Sun, and Ee-Peng Lim. 2008. Comments-oriented document summarization: understanding documents with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298. ACM.

Elham Khabiri, James Caverlee, and Chiao-Fang Hsu. 2011. Summarizing user-contributed comments. In *ICWSM*.

Peter Kolb. 2009. Experiments on the difference between semantic similarity and relatedness. In *Proceedings of the 17th Nordic Conference on Computational Linguistics-NODALIDA09*.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics.

Clare Llewellyn, Claire Grover, and Jon Oberlander. 2014. Summarizing newspaper comments. In *Eighth International AAAI Conference on Weblogs and Social Media*.

Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on World wide web*, pages 131–140. ACM.

Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. 2012. Topic-driven reader comments summarization. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 265–274. ACM.

Mārcis Pinnis, Nikola Ljubešić, Dan Ştefănescu, Inguna Skadiņa, Marko Tadić, and Tatiana Gornostay. 2012. Term extraction, tagging, and mapping tools for under-resourced languages. In *Proceedings of the 10th Conference on Terminology and Knowledge Engineering (TKE 2012), June*, pages 20–21.

G. Salton and M. Lesk, E. 1968. Computer evaluation of indexing and text processing. In *Journal of the ACM*, volume 15, pages 8–36, New York, NY, USA. ACM Press.

Arthur D. Santana. 2014. Virtuous or vitriolic. *Journalism Practice*, 8(1):18–33.

Dyut Kumar Sil, Srinivasan H Sengamedu, and Chiranjib Bhattacharyya. 2011. Supervised matching of comments with news article segments. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2125–2128. ACM.