Sheffield-Trento System for Comment-to-Article Linking and Argument Structure Annotation in the Online News Domain

Ahmet Aker, Fabio Celli, Emina Kurtic, Mark Hepple, Rob Gaizauskas
University of Sheffield
University of Trento
`ahmet.aker,e.kurtic,m.r.hepple,r.gaizauskas@sheffield.ac.uk,fabio.celli@live.it`

Abstract

In this paper we describe and evaluate an approach to linking readers' comments to online news articles. For each comment that is linked based on its comment, we also determine whether the commenter agrees, disagrees or stays neutral with respect to what is stated in the article. We use similarity features to link comments to relevant article segments and Support Vector Regression models for assigning argument structure. Our results are compared to competing systems that took part in MultiLing OnForumS 2015 shared task, where we achieved best linking scores for English and second best for Italian.

1 Introduction

Readers' commenting to online news articles has become a much used way of communication between online media outlets and their readers and as such is gaining importance for many stakeholders in online news media business. At present it is not easy to determine which parts of the news article a comment relates to. Being able to link a comment to the article segment that triggered it, however, is a crucial step in higher level comment processing tasks, like automatic comment summarization. Comments that link to the article can be clustered to identify topically related contributions to conversations and representative comments from clusters can be used to build summaries. Furthermore, the relative importance of comments for the summary can be determined based on their linking scores, as well as comment cluster size for example.

Linking comments to the article segments can be enriched with further information on how a comment related to the article segment. For example, for a comment summary it is interesting to know whether a particular comment agrees or disagrees with the article. In this way an overview of the comment-article argument structure and readers' sentiment towards the article can be built and included into the summary.

In this paper we report the details of our system for comment-article linking with argument structure detection that was submitted to the MultiLing Online Forum Summarization (OnForumS) 2015 shared task. Specifically, the task is to bring together readers' comments with online news article segments that comments refer to. Furthermore, we determine if a comment agrees, disagrees or states no opinion on what it stated in the news article segment. Our linking approach is based on text similarity measures for which we have demonstrated that they perform similarly to more elaborate topic modelling methods on the comment-to-artcile linking task (Aker et al., 2015). In this way we establish a simple and effective system that can be used for linking directly or as a baseline for further experimentation.

The paper starts with defining the tasks (Section 2) and the description of the linking and argument structure assignment methods (Section 3). In Section 4 we report our experimental results and summarize the paper in Section 5.

2 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.

For argument structure detection, we assign each segment-comment pair (s, c) to *agree*, *disagree* or *neutral* categories.

3 Method

3.1 Linking

For linking comments and news article sentences we use the method described in (Aker et al., 2015) that involves an initial linking of all comments that include quotations and a secondary step, in which similarity metric linking is performed for all comments without quotes. 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. 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.

The articles and comments are pre-processed before these two linking steps are performed. For preprocessing, we first split the news article into segments. We treat each article sentence as a segment and group each comment into a single unit regardless of the number of sentences it contains, although the shared task allowed comments to be split into single sentences, so only parts of comments could be linked to the article sentences. 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 pre-processing includes tokenization[1] and lemmatization, after which we either use words with stop-word removal or terms 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.

The first linking step involves linking all comments that include 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.

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.

- **Dice:**

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

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:**

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

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

- **NE overlap:**

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

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.

---

[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.

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:

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

where $weight_i$ is the weight associated with the ith feature. The weights are trained based on linear regression using the Weka package. 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.

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). 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 **??** 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.

3.2 Prediction of agreement/disagreement relations.

We trained the system for agreement/disagreement on 2260 comments extracted from CorEA (Celli et al., 2014), an Italian news blog corpus manually annotated with agreement/disagreement labels. The labels are numerical and can be: "agreement" (1), "disagreement" (-1) and "neutral" (0). We eliminated the non-applicable cases, annotated as "NA" in CorEA. In the dataset we used there are

---

http://theguardian.com

weight	feature
-0.3834	ratio
+0.4619	internal-punctuation ratio
-0.2585	apices ratio
-0.6137	char-word ratio
+0.5489	uppercase initial unique words ratio
-0.7739	median of the similarity score of Uppercase words
-0.2561	mean sine of paired word frequency

Table 1: Selected features.

Participant and run	Precision score
BASE-overlap	0.928571428571429
USFD_UNITN-run2	0.892561983471074
JRC-run1	0.857868020304569
UWB-run1	0.851063829787234
JRC-run2	0.829192546583851
USFD_UNITN-run1	0.818461538461538
BASE-first	0.738878143133462
CIST-run2	0.709424083769634
CIST-run1	0.702239789196311

Table 2: MultiLing OnForumS 2015's results for the linking task - English

1000 disagreement, 783 agreement and 215 neutral labels. The reported inter-annotator reliability for the annotion of the 3 labels is k=0.6 (Fleiss et al., 1981).

The CorEA corpus is in Italian, but we trained a cross-language model, extracting a vector of 84 shallow statistical dimensions about text encoding, characters, ngrams, punctuation, numbers, parentheses, uppercases, lowercases, word freq, word length, string similarity, emoticons, parentheses, tf*idf, similarity of uppercase words and sine of the frequency of word pairs. We normalized all the features and trained the system using a 66% of the data and tested it on 33%. We performed feature selection searching for the subset of features with the highest individual predictive ability and the lowest degree of redundancy (Hall and Smith, 1998).

We trained a Support Vector Regressor (Shevade et al., 2000) and obtained a Mean Absolute Error (MAE) of 0.42 over a majority baseline (score mean) of 0.44. The selected features and the weights in the SVM are reported in Table 1. The system has not been tested on English.

4 Evaluation

The performance of our system (*USFD_UNITN*) was evaluated within the MultiLing 2015 Online Forum Summarization (OnForumS) task and reported relative to a baseline system and 3 further competing systems. The evaluation was performed with English and Italian data, and the precision results are reported for linking (Tables 2 and 4) and argument structure (Tables 3) detection.

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

Participant and run	Precision score
CIST-run2	0.990601503759398
CIST-run1	0.988527724665392
UWB-run1	0.974358974358974
BASE-first	0.915531335149864
JRC-run2	0.896153846153846
USFD_UNITN-run1	0.891891891891892
JRC-run1	0.884848484848485
BASE-overlap	0.881578947368421
USFD_UNITN-run2	0.85981308411215

Table 3: MultiLing OnForumS 2015's results for the argument structure detection task - English

Participant and run	Precision score
BASE-overlap	0.590909090909091
UWB-run1	0.25
USFD_UNITN-run1	0.2
JRC-run1	0.152380952380952
CIST-run1	0.0842696629213483
CIST-run2	0.33
BASE-first	0.0103092783505155

Table 4: MultiLing OnForumS 2015's results for the linking task - Italian

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 evaluated. However, for Italian our second run with the threshold 0.5 was not considered. Furthermore, for Italian, we submitted no argument structure detection modules.

For linking task in English our second run achieved 89% precision and outperformed all competing systems apart form the overlap baseline. Our first run (run with the 0.3 threshold) achieved 82% precision and came 5th. For Italian our first and only run got the 2nd position scoring 20% precision. Our best precision result for argument structure assignment in English is 0.89, which is the 6th place among all competing systems.

5 Conclusion

In this paper we report the details of the Sheffield-Trento system for comment-to-article linking and argument structure assignment in the online news domain for English and Italian. The system links readers' comments to news article sentences that triggered them and is based on a combination of quotation detection and a combined similarity computation between comment and article sentence. In addition argument structure (agreement, disagreement, neutral) is assigned to comment-article sentence pairs. The system has been evaluated within the MultiLing 2015 Online Forum Summarization (OnForumS) shared task. For the linking task in English our system outperforms all other competing systems. However, for Italian linking as well as for argument structure assignment in both languages, there is a substantial scope for improvement compared to other systems that participated in the shared task.

Acknowledgements

References

Ahmet Aker, Emina Kurtic, Mark Hepple, Rob Gaizauskas, and Giuseppe Di Fabbrizio. 2015. Tcomment-to-article linking in the online news domain. In *Proceedings of the SigDial*.

Fabio Celli, Giuseppe Riccardi, and Arindam Ghosh. 2014. Corea: Italian news corpus with emotions and agreement. In *Proceedings of CLIC-it 2014*, pages 98–102.

Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. 1981. The measurement of interrater agreement. *Statistical methods for rates and proportions*, 2:212–236.

Mark A Hall and Lloyd A Smith. 1998. *Practical feature subset selection for machine learning*. Springer.

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.

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.

Shirish Krishnaj Shevade, S Sathiya Keerthi, Chiranjib Bhattacharyya, and Karaturi Radha Krishna Murthy. 2000. Improvements to the smo algorithm for svm regression. *Neural Networks, IEEE Transactions on*, 11(5):1188–1193.