Comments-Oriented Blog Summarization By Sentence Extraction

Ritika Tanwani (201101153), Sk Dilwar Hossain (201305652),
Tejas Shah (201102073), Tohar Patel (201305605)
IIIT Hyderabad

Abstract
Comments are very important part of blog posts. Most of the research done on blog summarization didn't consider comments. Comments gives the general view of readers and can be used to identify most relevant sentences from original post. The proposed solution is to derive relevant comments and then retrieve post sentences mostly referred to by these comments. The relevant comments are identified by ReT(Response and Topic) graph and named entities similarity score. Followed by summation-based sentence selection showed promising results.

Keywords
Blog, Comments, Cosine Similarity, Named Entity

1. INTRODUCTION
Most of the blogs have comments associated with them. A recent study on blog conversation showed that readers treat comments associated with a post as an inherent part of the post [2]. These comments gives the general opinion about the blog post and this knowledge can then be used to extract most relevant sentences from the blog post.

2. PROBLEM STATEMENT
Given a document D consisting of a set of sentences D ={s1 , s2 , . . . , sn }, and a set of comments C = {c1 , c2 , . . . , cl } associated with D, the task of comments-oriented document summarization is to extract a subset of sentences from D, denoted by Sc (Sc ⊂ D), that best represents the tpic(s) presented in D and discussed among its comments C.

3. APPROACH

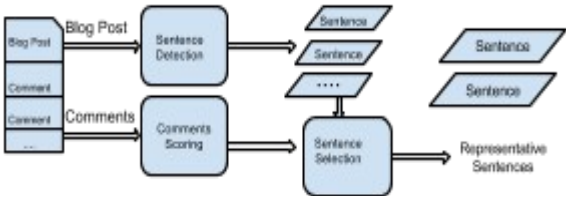

Figure 1: Comments Oriented Blog Summarization

Each word in the comment gets a weight from the different factors. The words that qualify a certain threshold are considered for the summary generation.

The factors are as follows :

A. Representative words extraction from comments:

1. Readers authority:

The reader authority is calculated as the number of distinct user that post a reply to the users comment.

Contribution to the weight of the term = tf(term,ci) * A(ci)

where

- tf(term,ci) is the term frequency of the word in the comment ci
- A(ci) is the authority author of the comment ci

2. Likes count: The number of likes to a comment
Contribution to the weight of the term = L(Ci)
where, L(Ci) = number of likes on the comment

3. Cluster: we have a algorithm that clusters the comments and a weight is assigned to a cluster.
Contribution to the weight of the term = tf(term,ci)*W(ci)
where,
- tf(term,ci) is the term frequency of the word in the comment ci
- W(ci) weight of the cluster to which the comment belongs

4. Reply count : The number of replies a comment has got.
Contribution to weight of the term=Rep(Ci)
where, Ci = number of replies to the comment Ci

5. Named Entity: All the named entities of blog post are identified by using Stanford CoreNLP[2]. For each comment Named Entity Score E(Ci) is calculated as :
E(Ci)=Number of named entities in that comment
Contribution to the weight of the term = E(Ci)

B. Top sentence extraction from Blog post:
1. The blog is divided into sentences.
2. Each sentence is then represented by a bag of words and weightage is given by
W(Si)=Sum(W(Ti))|Ti belongs to Si
where,
W(Si) = weightage of the sentence Si,
W(Ti) = weightage of the term Ti as calculated in A.
3. A number of top weighted sentences is shown as the summary.

4. DATASET

We selected TechCrunch blogs (http://techcrunch.com/) which have relatively large readership and are widely commented.

5. CONCLUSION

Our proposed solution gives a good score for bigger length blog posts and big enough dataset of comments associated with it.

6. REFERENCES

[1] Meishan Hu, Aixin Sun and Ee-Peng Lim "Comments-Oriented Blog Summarization by Sentence Extraction".
[2] Stanford CoreNLP v1.3.4 – 2012-11-12
[3] Link to the source code github.