Creating Conversations: An Automated Dialog System

Lisa Gandy, Kristian Hammond
Northwestern University
2133 Sheridan Rd, Evanston IL, 60208
lisagandy@u.northwestern.edu, khammond@cs.northwestern.edu

Abstract

Online news sites often include a comments section where readers are allowed to leave their thoughts. These comments often contain interesting and insightful conversations between readers about the news article. However the richness of these conversations is often lost among other meaningless comments, and moreover all comments are found at the bottom of the web page. In this article, we discuss how our system inserts reader conversations into the news article to create a multimedia presentation called Shout Out. Shout Out features two virtual news anchors: one anchor reads the news and when appropriate the anchor pauses to have a conversation about the news with another anchor. This current iteration of Shout Out combines natural language techniques and reader conversations to create an engaging system.

Introduction

Building machines and systems that create new content is an active research area in Computer Science (Sauper and Barzilay 2009) (Evans 2004). However, the majority of approaches to this problem have typically been hampered by strong domain-dependence and a subsequent lack of scalability. We have been working on a new approach that combines human editorial judgment with the inherent scalability of the web; we call this approach "machine-generated content."

News at Seven (Nichols and Hammond 2009) is an example of one such machine-generated content system which creates an automatically generated audio/visual news show complete with animated anchors and text-to-speech generated dialogue. News at Seven builds its content using a variety of narrative arcs that create a certain type or style of presentation. In this paper, we present and discuss in detail a narrative arc: *Shout Out*. In Shout Out, one anchor presents a news story, and then stops at appropriate moments to discuss the story with the other anchor.

Previous systems such as Newsblaster (Evans 2004) and Sauper and Barzilay's Wikipedia Article Generation system (2009) combine several web resources to create entirely new content. Shout Out also combines various web resources, however Shout Out not only creates new content from these resources, it also creates *news commentary*.

In order to provide context for the remainder of this paper, a final dialog produced by the Shout Out system is given below. In this presentation Anchor 1 gives the news and also discusses the news with a second anchor. The news article is entitled *'Moonlight' comes back from the dead on the CW (sort of)*. Remarks from the news article and from reader comments are labeled as either from the news article or from the reader comments.

Anchor 1 [Article]: In today's news, Moonlight comes back from the dead on the CW (sort of). Though Alex O'Loughlin's vampire drama "Moonlight" was canceled by CBS in 2008, The CW is resurrecting it for a vampire-themed Thursday beginning June 3…

Anchor 1 [Comment 1]: You know, I don't know if this makes me happy or furious. Why the hell didn't the CW pick up Moonlight after CBS foolishly canceled it? It could have been a giant hit for the CW then - now just filler for The Vampire Diaries. Alex has moved on and so has the rest of the cast. So is it better late than never? You tell me.

Anchor 2 [Reply, Comment 1]: Well, they probably didn't pick it up because it doesn't feature teenagers or kids in their 20s.

Anchor 1 [Article]: After its cancellation, efforts were made to sell "Moonlight" to another network, but ultimately they failed. With the "Twilight" phenomenon showing no signs of stopping and the success of "The Vampire Diaries" and "True Blood, " The CW hopes that the "Moonlight" reruns will find a new audience.

Anchor 2 [Comment 2]: There only gonna play a rerun of the series during the summer, no new season planned.

Once a Shout Out presentation is formed it is combined with images and flash to create a multimedia presentation.

The remainder of the paper will proceed as follows: in Section 2, we discuss the Shout Out system in detail, and feature a running example based on the example given

above. We conclude by discussing our future goals concerning Shout Out.

The Shout Out System

This section will focus on how the Shout Out system gathers reader comments, identifies reader conversations, and injects those conversations at appropriate points in a news article to create the final Shout Out product.

Gathering Articles and Reader Comments

As mentioned previously, Shout Out augments news articles with reader comments. In order to gather the news articles, the Shout Out system monitors the RSS feeds of Zap2It.com and Entertainment Weekly for new articles. When new articles appear on the feed Shout Out picks up the article for processing. A web scraper extracts the content of the news article and then divides the content into paragraphs. Reader comments are also mined from the article. However, individual articles often contain a small number of reader comments. Since reader comments are an important part of Shout Out, this presents a problem for the system. In order to mine more comments, Shout Out uses the Google search engine to find a similar article to the original from another news outlet. Since the system currently uses news articles from two websites, if the original website used is Zap2it then the system searches for articles from Entertainment Weekly, and vise versa.

Many previous systems have used internet search engines to mine for similar articles. These systems include Tell Me More (Iocabelli, Birnbaum and Hammond 2010), Local Savvy (Liu and Birnbaum 2008) and Sauper and Barzilay's Wikipedia article generation system (2009). These systems use a combination of mined entities and constraints such as location or relevant domain. Likewise, the Shout Out system forms queries by combining entities and a news source constraint as well as a time constraint.

In this case, entities are extracted using WPED (Wikipedia Entity Detector) (Iacobelli et al. 2010). The date constraint is included to ensure that all retrieved articles pertain to the same event in time. The date ranges from the publish date of the original article to one day after, if it is not the current day. Finally, the preferred news source is also added to the query.

Per our running example, where the original article is from zap2it, was published on May 6, 2010 and is entitled *'Moonlight' comes back from the dead on the CW (sort of)*. The query *Moonlight source: entertainment _weekly 2455323 - 2455324* is formed and this results in finding the article *'Moonlight' reruns head to CW* which was published on the Entertainment Weekly website on May 6, 2010.

We performed a small user study including 35 participants to confirm that the news articles returned by the Google News Search are indeed similar in topic to the original article. All studies mentioned in this paper were performed using Mechanical Turk (Kittur, Chi, and Suh 2008).

Our survey consisted of 10 original articles and an average of 5 articles mined by the system in relation to each original; this resulted in a total of 49 distinct original-retrieved pairs. Survey participants were asked "Based on the title alone, how similar do you think the topic of article 1 is to article 2?" Answers were based on a five point Likert scale, ranging from "very similar" to "very dissimilar". Final results for the survey are shown below in Figure 2.

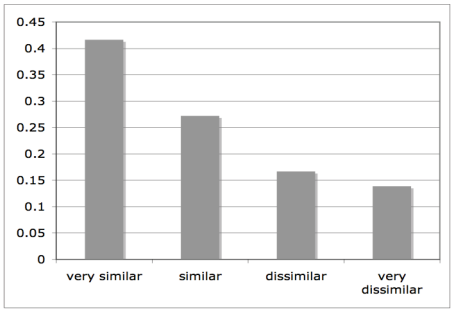

Figure 2: Original and retrieved title similarity.

The difference between proportions of titles found to be very similar and similar in topic (77%) and dissimilar and very dissimilar in topic (23%) is significant with a p-value less than 0.0004.

Once a set of suitable articles are found, comments are scraped from each article. All comments are filtered for length and must be shorter than 80 words, must not be all uppercase, as well as contain no profanity. Per our running example, the original zap2it article, *'Moonlight' comes back from the dead on the CW (sort of),* had 7 suitable reader comments, and corresponding articles from Entertainment Weekly had 17 suitable comments, after filtering.

Identifying Reader Conversations

Comments often feature meaningful conversations in relation to events. For example (Shamma and Churchill 2010) found that twitter comments reflect the structure of media events and often indicate the level of interest in an event itself.

The goal of Shout Out is to utilize reader comments to create conversational dialogs centered on a news article. There are usually several "conversations" between commenters that are found in comment streams. The Shout Out system seeks to extract these conversations and

include them in the final Shout Out dialog to make the dialog more conversational and less one sided.

In order to find conversations within comment streams, the system first makes use of the natural layout of a reader comment forum to pair comments and their replies. For example, Entertainment Weekly often uses separate div tags to differentiate replies. However, not all users mark their comment as a reply by explicitly linking their reply to the original comment. Therefore we also analyze the content of the reader comments to match replies to comments through certain clues.

The first clue that a comment is a reply is that it will reference the name of the original commenter. We explicitly check that a reference to another commenter is made within the first 1/3 of a comment. For example a typical reply comment would be "I agree, Sara. Z&M was a riot! The only Smith movie I haven't seen is Jersey Girl…", where *Sara* is the original commenter's name.

However when analyzing this clue there are some confounding factors that occur. First, we have to ensure that the original commenter's name is not a name that also occurs in the article text, where the commenter has the same first name as a celebrity. Second, we have to ensure that the commenter name is not a dictionary word. A good example of this error is that a commenter's nickname is Star, and then a comment mentions the word "star" and the comment is falsely classified as a reply.

We collected three days of articles from Zap2It and Entertainment Weekly and kept only the articles, which contained user comments. In regards to Zap2It articles, 27 articles were gathered, and there were an average of 6 comments per page and 0.33 conversations per article. In regards to Entertainment Weekly articles, 12 articles were gathered, and there were an average of 24 comments per page and 9.23 conversations per article. One important note about the Entertainment Weekly articles is that only the first page of comments is gathered, and only replies, which are one layer deep, are kept by the system.

The system correctly identified 100% of the conversations that occurred in articles from both news sources. In regards to Entertainment Weekly articles, this is fairly trivial as all replies are explicitly marked. However, the Zap2it site does not give the reader an opportunity to reply to other comments, and all conversations, which were present, were identified through the techniques explained in this section.

Per our running example, in which the system has gathered 24 comments, there are 6 conversations and 18 single comments. Some examples of conversations are given below.

Comment 1: I don't know if this makes me happy or furious. Why the hell didn't the CW pick up Moonlight after CBS foolishly canceled it? It could have been a giant hit for the CW then - now just filler for The Vampire Diaries. Alex has moved on and so has the rest of the cast. So is it better late than never? You tell me.

Reply 1: They probably didn't pick it up because it doesn't feature teenagers or kids in their 20s.

Comment 2: When "Hawaii Five-0 gets canceled, O'Loughlin will be free to make "Backup Plan 2."

Reply 2: LOL. You crack me up! BUP#2 has a nice ring to it.

Comment 3: Will this get in the way of Supernatural reruns?

Reply 3: No, they are moving Supernatural reruns to Friday nights after Smallville. I am guessing that is where the show will stay next season.

Aggregating Reader Conversations

Many of the conversations that occur in a reader comment forum are repetitive. In order to avoid this repetition we aggregate the comments into groups using agglomerative clustering. Agglomerative clustering is used rather than k-means, because we are not aware of the number of clusters beforehand. The method of similarity used to determine clusters is cosine similarity. The minimum cosine similarity used when clustering is 0.13, any comment whose cosine similarity to a candidate cluster is lower than 0.13 will form a new cluster. Experience reveals this number to be a good value for our purposes.

Per our running example, there are 2 clusters that are formed, these are given below.

Cluster 1

Comment 1: I agree had Moonlight come out a year or two later it might have done OK. A lot of people also skipped it because it sounded a lot like Forever Night, too.

Comment 2: IMO, "Moonlight" was a victim of bad timing. I think it would have succeeded had it come out a year or two later.

Cluster 2

Comment 1: There only gonna play a rerun of the series during the summer no new season planned.

Comment 2: wait i don't get it??? Are they gonna just play it in the summer or are they gonna make a new season?

In order to evaluate the effectiveness of clustering by the system we evaluated the intercluster and intracluster similarity of 11 pairs of clustered comments that were all mined taken from a set of related articles. Results are given below.

Per the two sample t-test t(df=10, 4.14), the difference Between intercluster and intracluster similarity is p less

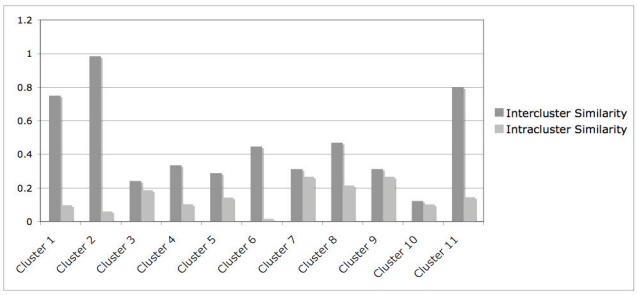

Figure 3. Intercluster vs. intracluster distance.

than 0.05. This distinct difference between the two similarities indicates that the clusters are compact and consist of repetitive comments that are differentiable from comments in other clusters.

Adding Conversations to the News

Once all comments are aggregated into clusters, they are injected into the original article. First the original article is divided into paragraphs. Next, the cosine similarity of each paragraph to each comment within every cluster is calculated. Once the comment with the highest cosine similarity to the paragraph is found it is paired with the paragraph. The cluster that the comment is part of is discarded. This process continues until either all paragraphs have been examined or there are no more reader comments available. Pairing comments with clusters is done in a two-stage pass. In the first pass, comments with replies (i.e. conversations) are preferred, in the second pass; comments without replies are paired with any paragraph that was not already matched. Using cosine similarity to match comments to paragraphs insures that any comments that were gleaned from articles that the system inaccurately chose as similar to the original article are not included in the final Shout Out dynamic.

In order to assess the quality of pairs of paragraphs and comments we surveyed 15 participants. The survey consisted of 24 pairs of paragraphs and comments. Survey participants were asked "Evaluate the relevance of the reader comment to the paragraph". Answers were based on a four point Likert scale ranking from "very relevant" to "very irrelevant". The results are given below in Figure 4.

The difference between proportions of comments found to be very relevant and relevant (85%) and irrelevant and very irrelevant (15%) is significant with a p-value less than 0.0001.

In order to assess whether adding comments into a news article actually enhanced the news article, we performed a small survey. We gave 30 participants two Shout Out transcripts. One transcript was the example given in the introduction, and the other transcript was the same example without reader comments. We asked participants to rate whether Transcript 1 or Transcript 2 was more entertaining than Transcript 2 or vice versa. 63% of participants found Transcript 1 more entertaining, this percentage of participants is significant at p less than 0.04.

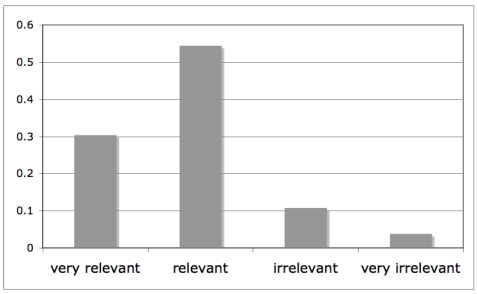

Figure 4: Relevance of comments to paragraphs.

Future Work and Conclusion

There are several improvements we intend to make in the future to the Shout Out dynamic. One improvement is to extend the Shout Out dynamic towards new genres. We believe that Shout Out would be suited towards political news and sports news, as both genres features commentators who are highly opinionated. Though there are improvements to be made, we are pleased with the system and believe that it demonstrates that reader conversations can easily be used to enhance the overall quality of a news article.

References

Evans et al. 2004. Columbia newsblaster: multilingual news summarization on the Web. In HLT-NAACL. 1-4.

Iacobelli, F., Birnbaum, L. and Hammond, K. 2010. Tell Me More, not just "more of the same". IUI 2010, Hong Kong.

Iacobelli, F., Nichols,N., Birnbaum, L. and Hammond, K. 2010. Finding new information via robust entity detection in proactive assistant agents (PAA2010) AAAI 2010 Fall Symposium. Arlington, VA.

Kittur, A., Chi, E.H. and Suh, B. Crowdsourcing user studies with Mechanical Turk. 2008. In the Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems. pp. 453-456.

Liu, J., Birnbaum, L. 2008. LocalSavvy: aggregating local points of view about news issues. WWW 2008 Workshop on Location on the Web. 33-40.

Nichols, N. Hammond, K. "Machine-Generated Multimedia Content." Proceedings of the Second International Conference on Advances in Computer-Human Interactions, 2009.

Sauper, C., and Barzilay, R. 2009. Automatically generating Wikipedia articles: a structure-aware approach, ACL and AFNLP: Vol. 1, Suntec, Singapore. 208-216.

Shamma, D., Kennedy, L. and Churchill, E. 2010. Tweetgeist: can the Twitter timeline reveal the structure of broadcast events? CSCW Horizons.