Motivation

Novel text mining problem: mine news and text and social media jointly to discover the most controversial sentences in news.

- Highlighting controversial points in news articles for readers.
- Revealing controversies in news and their trends over time.
- Quantifying the controversy of a news source

Leverage relevant comments in Twitter to assess public opinions about an issue mentioned in a news articles

Implementation

We implemented
1. the algorithms that use our multimodal ranking theory
2. a real-time server and interface
   - content is gathered from the necessary sources
   - ranked using controversy scoring and
   - presented to the user in a pleasant yet efficient manner

In an effort to develop a keyword-based controversy application programming interface (API) for future usefulness, we have created an impressive user-facing product. Our API will continue to get more accurate by leveraging click-through rates and timestamps, indicative of human interest.

Overall Application Flow

1. The user performs a keyword search
   - Build a news article corpus
   - Build a twitter corpus
2. Map the most relevant social content to each sentence in a news article
   - Rank matching tweets according to their relevance to a sentence
   - Okapi BM25 retrieval function [2]
3. Analyze sentiment and linguistics features
4. Feed our controversy scoring function
   - For each news article produce:
     - A ranked set of sentences
     - An overall score
5. Make the system real-time and low-latency
6. Create a user-interface which leverages our API
7. Track usage to increase accuracy of controversy detection

Mining Approach

Controversy Scoring function

Entropy is a widely used measure of uncertainty of a random variable. We thus propose scoring controversy based on entropy (e.g., of the distribution over the polarities of sentiment) with a higher entropy indicating more controversy.

Entropy $H(X)$ of a random variable $X$ with $n$ outcomes $x_1, \ldots, x_n$ is defined as

$$H(X) = - \sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

where $p(x_i)$ is the probability mass function of outcome $x_i$.

For example, it is easy to interpret sentiment as a discrete random variable $X_{sent}$ with $n$ possible outcomes and probability function:

$$p(X_{sent} = x_i) = \frac{f(x_i) \in C'_i}{\sum_{i=1}^{n} f(x_i) \in C'_i}$$

where $f(x_i) \in C'_i$ is the number of comments that have sentiment equal to $x_i$ and $\sum_{i=1}^{n} f(x_i) \in C'_i$ is the total number of comments.

Results in previous evaluation reveal a high performance (82.59% in ranking sentences) on a data set created from controversial debate topics found online [1]

Future Directions

Over the course of Summer 2015, we hope to continue to improve our user-facing product and API. Our source-code has already been open-sourced [3] and our API routes will be made public shortly.

References


Acknowledgments

We would like to thank **Lisa Huang** for her continued contributions to this project as well as **Professor ChengXiang Zhai** and the **PURE Committee** for their support of our work.