Estimating sentiment in Eli: Computational analysis of tone in student responses to student writing (PROPOSAL)

Michael Wojcik
Rhetoric & Writing
Michigan State University
East Lansing, Michigan
wojcikm4@msu.edu

Abstract

Eli is a web-based application for coordinating and assessing writing review, typically used by students in English composition classes at the secondary or undergraduate level. It provides students and instructors with information about the reviews that students write of one another's writing, based on reception (whether the original author finds the review helpful), but does not currently attempt to compute any metrics based on the content of the reviews themselves. I describe a prototype of a proposed addition to Eli which employs sentiment-determination techniques to analyze the rhetorical tone—the attitude of the writer toward the material—of reviews. I describe how this information may be useful for instructors, and how the introduction of NLP techniques into an application like Eli can extend its capabilities; and discuss what sentiment analysis might tell us about writing review.

1 Credits

This project was developed for Joyce Chai's CSE841 Natural Language Processing class at Michigan State University, Spring 2011. The original Eli research project was developed by Mike Mcleod and Bill Hart-Davidson at the WIDE (Writing in Digital Environments) Research Center at Michigan State University. WIDE subsequently entered a partnership with Red Cedar Solutions Group in Okemos, Michigan, to produce a commercial version of Eli, which is currently in beta-test stage.

2 Introduction

Eli, a web-based application for coordinating and assessing writing review, is an innovative approach to teaching (writing) composition, particularly at the secondary and undergraduate levels, and potentially a useful tool for professional writing management as well. It grew out of work at the Writing in Digital Environments (WIDE) Research Center at Michigan State University which was aimed at applying the tools of "social media" websites to contemporary composition pedagogy, and is now being developed into a commercial product for the education market (WIDE, 2011). Eli provides writers, typically students in writing classes, with an environment in which they can read and review one another's writing, and then provide feedback on the reviews they receive in turn, so they can become better reviewers as well as better writers. Research in language-composition pedagogy has shown that peer review coupled to a process for improving reviewing skills is one of the most effective ways to improve a writer's performance; Eli is designed to streamline this process.

In the Eli environment, students can post their own writing, then read other students' texts and write free-form reviews of them. (They may also respond to more structured review questions set by the instructor, such as Likert-scale ratings and checklists of requirements. Instructors can provide "prompts" to guide students in writing their reviews.) As students receive reviews on their own writing, they can evaluate those reviews, and develop plans for revising their own writing. Meanwhile, Eli shows them how the reviews they created were received by their fellow authors. So each Eli user does four kinds of writing: primary text, reviews, responses to reviews, and revision plans. Part of Eli's function is simply to provide space for and coordinate all of these activities. Beyond that, though, it tracks user actions such as creating and viewing objects (primary texts, reviews, etc), highlighting or copying text, commenting, and so on.

Eli calculates a *helpfulness metric* (now the subject of a patent application by the unviersity) which gauges the utility of a review based on the receiving author's actions and feedback (Hart-Davidson *et al.*, 2009). For example, when an author copies a suggestion from a review onto the revision plan for the next revision of the writing assignment, Eli notes that action and increases the reviewer's helpfulness score. Eli maintains a cumulative helpfulness score for each user (student), based on the average helpfulness of that user's reviews. Instructors use this information to track student performance as a peer reviewer, and students use it to improve their reviewing skills. However, Eli currently does not attempt to compute any metrics based on the actual *content* of reviews, only what the reviewed author does with them.

This is an application that appears ripe for the application of Natural Language Processing techniques, to give the Eli system mechanisms to estimate additional helpfulness characteristics and other information about the reviews that users write. In fact, WIDE has been discussing this possibility since at least 2008, when I begin some exploratory work for the Center on heuristics for gauging an author's *ethos*, or reputation among members of the audience (Wojcik, 2009). In this project I will implement a framework for adding NLP features to Eli, and a prototype module that will do sentiment analysis on student reviews, and present it in terms of rhetorical tone. I'll evaluate my selected sentiment-determination algorithms against my manual determination of tone for the reviews in the data set. After discussing what sentiment analysis might tell us about writing review in theory, and how the sentiment-analysis information might be used by instructors, I'll touch on some of the broader implications of what NLP might do for Eli.

3 Sentiment Analysis

The literature on sentiment analysis (including sentiment classification, extraction, retrieval, aspect identification, etc) is large. This is a popular area of NLP research, no doubt in part because of its commercial value: it is used by media-analysis professionals in government, public relations, and similar fields; by market analysts studying online product reviews; and in other well-funded areas. But it is also of considerable theoretical interest, since it incorporates syntactic, semantic, and pragmatic problems in language analysis; often deals with unstructured and noisy data;[1] and seeks to extract high-level, often implicit meaning from the input.

Liu (2009) is a useful overview of the problem of sentiment analysis and common formulations of this kind of work, including a brief look at the application to online product reviews, one of the most frequently discussed areas.

Many older approaches to sentiment analysis used a bag-of-words model that tried to identify sentiment-bearing terms. Gamon and Aue (2005) describe an automatic approach along these lines. As Nakagawa *et al.* (2010) point out, however, bag-of-words models inevitably have a high error rate because sentiment terms are often embedded in phrases that invert their meaning (e.g. "it's not that I dislike"). They present an approach using hidden variables representing the sentiment of dependency subtrees, which are combined to infer the sentiment of the sentence. This is more complex and computationally expensive than a bag-of-words approach but has significantly better accuracy.

Arora *et al.* (2010) go further, with a model that combines bag-of-words features and structural features, the latter derived from linguistic annotation graphs; they use a genetic-programming approach to reduce the large number of features extracted by their model to a salient subset. While this approach looks very powerful, it is also likely to be resource-intensive both in implementation and execution. Another novel approach, called HL-SOT for "Hierarchical Learning with Sentiment Ontology Tree," is described by Wei and Gulla (2010). They focus

[1] Online reviews, for example, are notorious for misspellings and other errors, slang, and inconsistencies. Brody and Elhadad (2010), among others, discuss some of these problems.

on solving two problems with sentiment analysis of product reviews in particular (though their method seems applicable to other problem domains): applying information contained in the hierarchy of attributes of the topic being reviewed, and coping well with sentences that express complex sentiments about multiple attributes. As with the system presented by Arora *et al.*, I believe Wei and Gulla's method is interesting but too complex to implement for this project.

Greene and Resnik (2009) move away from lexical approaches like bag-of-words to focus on syntactic "packaging," or how key ideas are arranged in the phrasal structure of the input. They present a strong argument for the presence of implicit sentiment cues for readers in syntactic structures, and their method has some success at identifying these.

Another recent proposal, from Brody and Elhadad, has two especially interesting features. One is that it incorporates information from topic aspects, which the authors claim often dominate sentiment interpretation: for example, "cheap" may be positive when referring to restaurant prices, but negative when applied to restaurant decor. The other is that their system is unsupervised, which not only means it's cheaper to implement, but, according to the authors, deals well with irregular data (unrecognized words, errors, etc).

3.1 Sentiment Analysis of Writing Reviews

Because student-writing reviews are a more focused genre than many of the domains where sentiment determination is currently popular (for example, product comments forums for online shopping sites), and since the writers feel some constraint to produce well-formed text, the input has relatively low noise, but sentiment markers also tend to be more subtle. Students are generally reluctant to express overtly negative opinions of one another's writing, particularly when author and reviewer are part of a small group such as a class.[2] They may try to couch their negative opinions "constructively" or in an ostensibly positive gloss, so e.g. "your second paragraph doesn't make sense" might become "I think your second paragraph might be better if you explained your ideas more". They also often preface their substantive remarks with formal expressions of approval ("overall I liked your essay"), as a sort of social nicety; this will distort a straightforward sentiment evaluation.

For those reasons, I predict that a bag-of-words approach will not be very successful. I intend to implement such a system, possibly along the lines described by Gamon and Aue, as a baseline, but expect more sophisticated methods to significantly outperform it. My candidates for improved methods are (probably simplified versions of) those of Nakagawa *et al.*, Greene and Resnik, and Brody and Elhadad; I'll probably investigate implementing them in that order, and due to time constraints may not be able to try all of them.

To address the problem of non-substantive polite expressions of approval in student reviews, I intend to look at the method described by Taboada *et al.* (2009), which tries to identify which paragraphs and sentences in a movie review contain substantive comments. Whether this is feasible in my project, or applicable to student-writing reviews, is still an open question at this point.

Another possibly useful study is that of Andreevskaia *et al.* (2007), looking at sentiment specifically in some blog genres. In some ways Eli has the "feel" of a blog, with main posts (the primary texts) and comments (the reviews), so approaches designed for blogs may be applicable; also, Eli users may tend to write in a "blog-like" fashion because the system reminds them of that environment.

4 NLP Framework for Eli

Eli is a modular web application written primarily in PHP, with data stored in a SQL relational database. It's quite straightforward to add new features to Eli, if they're relatively independent of existing workflows (as is the one I'm proposing). The output from my module will be web-based so that it can be integrated with other "instructor" views in Eli.

Because part of my goal is to open Eli up to future NLP-based projects, I intend to implement a general framework for NLP of the data contained in Eli. I will use the Apache implementation of

[2] Eli can be configured to anonymize authors and reviewers, and it can be used to have users review the work of authors they have no other connections with; but typically reviewers and authors are all members of a group, and are aware of that connection.

the Unstructured Information Management Architecture (UIMA) framework as the overall structure, and use existing Apache UIMA modules for basic tagging and parsing. UIMA is a standard framework for handling unstructured data, particularly the sort of noisy *ad hoc* text found on social-networking websites and the like, originally developed at IBM and now standardized by OASIS (Apache Foundation, 2011). It's being used in a wide range of applications, recently and famously in the IBM Watson Jeopardy-winning system (Pearson, 2011). The UIMA framework should make it straightforward to implement various sentiment-analysis engines and integrate them into Eli.

Since most of Eli is written in PHP, where possible my additions will also be in PHP. However, UIMA itself runs modules written in C++ and Java, so the majority of my code will be Java. In order to provide the web-hosting infrastructure and other access required by the system, the project will likely run on servers controlled by the WIDE Center; but I'll be able to turn in all the associated code via *handin*, and I'll implement command-line drivers for the sentiment modules so they can be tested outside Eli, using CSE department servers.

Data for the experiment will come from the existing Eli repository, which has been populated by students and instructors during the beta-test period. (Active testing is still being conducted, and will be at least through the end of the semester.) WIDE has agreed that this project falls under the scope of their existing Eli research, which has already received human-subjects research approval from the IRB; Bill Hart-Davidson is the primary investigator on that project.

5 Sentiment Analysis and Review Tone in Eli

For the prototype I will implement a number of sentiment-determination algorithms, and evaluate their performance against my own determination of the tone of the reviews. See the discussion of algorithms in section 3.

Rhetorical tone is a more abstract and complex concept than simply sentiment. Part of my larger project, which falls into the area I call *computational rhetoric* (Wojcik, 2011), is to investigate to what extent we can heuristically determine rhetorical tone based on extracted features such as sentiment. For this project, though, I will probably just use a simple mapping from sentiment to tone.

I believe this information has at least three possible uses. First, instructors may find that students respond differently toward reviews based on their tone, and use that information to make suggestions to students about how to couch their review comments. Second, review-tone information might be used to flag certain problematic reviews for instructor attention. Third, tone analysis of reviews could begin to address open research questions such as whether the helpfulness of a review is correlated to its tone or perceived sentiment.

6 Development Plan

The project is currently in its research stage, but I intend to begin assembling software for the development system next week (28 March). By the time of class presentations (25 April) I expect to have a working system and some results to report.

Project milestones include:

1. Obtaining access to the Eli data. I have secured approval from WIDE and have discussed the question with Mike Mcleod, the lead developer for WIDE projects. Eli is now primarily under the control of developers at Red Cedar Solutions Group, so Mike was taking the matter up with them. I expect it to be resolved before the end of the week.

2. Manual analysis of review data. I need to manually gauge sentiment in a corpus of reviews selected from the Eli data, for testing and supervised-learning purposes.

3. Configure development environment and server with necessary software (including Apache UIMA, etc).

4. Design and implement UIMA-based framework for NLP in Eli.

5. Implement baseline bag-of-words sentiment-analysis model in Java, including command-line testware.

6. Implement sentiment-analysis module for Eli using baseline model.

7. Implement UI for running sentiment analysis and displaying results in Eli.

8. Run end-to-end test with baseline sentiment-analysis module. Record results for comparison.

9. Implement additional sentiment-analysis models and modules. Test and compare to baseline.

10. Write final report.

Acknowledgements

I'd like to thank Dr Chai for the suggestion which led to the project in its current formulation, for her advice and encouragement, and for an interesting and enlightening course. I'd also like to thank Bill Hart-Davidson and Mike Mcleod (WIDE Center, Michigan State University) for access to the Eli system and data, and for numerous stimulating conversations about computational rhetoric in general and student writing-review evaluation specifically; this project would not exist without them. As always, my gratitude to Malea Powell for her encouragement and support.

References

[Andreevskaia and Bergler2006] Alina Andreevskaia and Sabine Bergler. 2006. Sentiment tagging of adjectives at the meaning level. In *Advances in Artificial Intelligence*, pages 336–346.

[Andreevskaia et al.2007] Alina Andreevskaia, Sabine Bergler, and Monica Urseanu. 2007. All blogs are not made equal: Exploring genre differences in sentiment tagging of blogs. In *International Conference on Weblogs and Social Media*.

[Apache Foundation2011] Apache Foundation. 2011. Apache uima. http://uima.apache.org/.

[Arora et al.2010] S. Arora, E. Mayfield, C. Penstein-RosÃ©, and E. Nyberg. 2010. Sentiment classification using automatically extracted subgraph features. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 131–139.

[Brody and Elhadad2010] S. Brody and N. Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812.

[Gamon and Aue2005] Michael Gamon and Anthony Aue. 2005. Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In *Proceedings of the ACL 2005 Workshop on Feature Engineering for Machine Learning in NLP, ACL*, pages 57–64.

[Greene and Resnik2009] S. Greene and P. Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 503–511.

[Hart-Davidson et al.2010] William Hart-Davidson, Michael McLeod, Christopher Klerkx, and Michael Wojcik. 2010. A method for measuring helpfulness in online peer review. In *Proceedings of the 28th ACM International Conference on Design of Communication - SIGDOC '10*, page 115, SÃ£o Carlos, SÃ£o Paulo, Brazil.

[Liu2009] Bing Liu. 2009. Sentiment analysis. *Invited talk at the 5th Annual Text Analytics Summit*.

[Nakagawa et al.2010] T. Nakagawa, K. Inui, and S. Kurohashi. 2010. Dependency tree-based sentiment classification using crfs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786–794.

[Pearson2011] Tony Pearson. 2011. How to build your own "watson jr." in your basement. https://www.ibm.com/developerworks/mydeveloperworks/blogs/InsideSystemStorage/entry/ibm_watson_how_to_build_your_own_watson_jr_in_your_basement7?lang=en, February.

[Taboada et al.2009] M. Taboada, J. Brooke, and M. Stede. 2009. Genre-based paragraph classification for sentiment analysis. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 62–70.

[Wei and Gulla2010] W. Wei and J. A Gulla. 2010. Sentiment learning on product reviews via sentiment ontology tree. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 404–413.

[WIDE2011a] WIDE. 2011a. Eli development blog. http://eli.wide.msu.edu.

[WIDE2011b] WIDE. 2011b. Writing in digital environments (wide) research center. http://wide.msu.edu.

[Wojcik2009] Michael Wojcik. 2009. Estimating ethos. http://www.ideoplast.org/rw/portfolio-2007-2008/ee/index.html.

[Wojcik2011] Michael Wojcik. 2011. Inventing computational rhetoric. Presentation (forthcoming) at the 62nd annual convention of the Conference on College Composition and Communication., April.