Segment-Predict-Explain for Automatic Writing Feedback

HAMED NILFOROSHAN, JAMES SANDS, KEVIN LIN, RAHUL KHANNA, and EUGENE WU, Columbia University

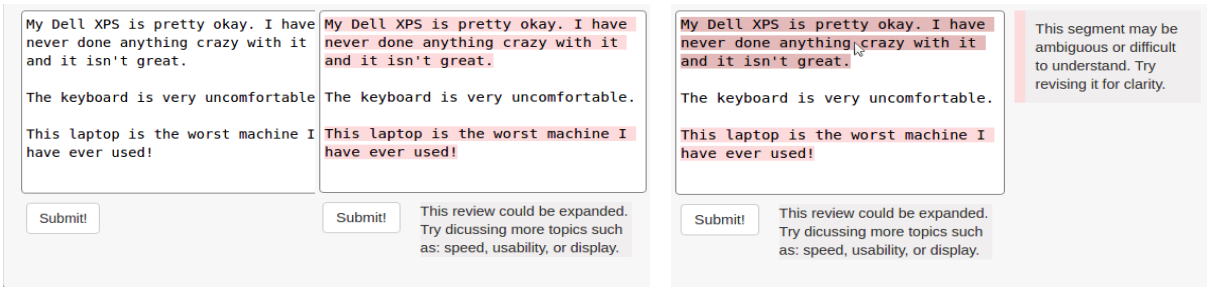
1. INTRODUCTION

The internet was envisioned as "a universal place of information" where humans can freely share knowledge [Berners-Lee 1997]. Increasingly, users rely on websites centered around specific communities or topics such as Q&A sites (e.g., Quora), forums (e.g., Reddit, Couchsurfing), reviews (e.g., Yelp, Amazon, AirBnB), and topic-oriented chat services (e.g., Mest [Unauthored 2016]). These services let users both share free-form text (e.g., review a product), and find what others have written (e.g., how do others like this product?); the value of such services relies on their ability to present high quality user-generated content to other users. Unfortunately, unlike more structured forms of input such as dates or emails for which validation is straightforward, it is challenging to measure and ensure the quality of free-form text content. In fact, the desired characteristics may differ depending on the community: while being balanced and unbiased is important in product reviews, a forum for political discourse may encourage strong stances. Although there are numerous techniques to improve the quality of the text that is served, they are still limited in terms of scalability and effectiveness; there is a need for a low-cost scalable system that can be easily customized to new domains.

Post-Submission approaches seek to identify high quality content after it has been written and submitted; by removing poor content [Spirin and Han 2012], finding high quality content algorithmically [Agichtein et al. 2008; Wang et al. 2013] or via user voting [Tang et al. 2013]. However, this assumes that the corpus of contributions is large enough that high quality content can be found for every entity, which is unlikely for less popular or niche websites and topics.

Alternatively, *Pre-Submission* approaches seek to improve quality prior to submission. Indirect approaches include content guides [Yelp 2016b; Amazon 2016; Wikipedia 2016] and incentivization [Yelp 2016a; Bosu et al. 2013], but only a subset of users read through such guidelines or are active enough to benefit from incentives. Direct approaches that provide rapid feedback during the writing process can improve content quality [Kulik and Kulik 1988], for instance, by asking students to write feedback for their peers' assignments [Kulkarni et al. 2015]. However, even when crowdsourced, such approaches can take on the order 20 minutes [Kulkarni et al. 2015]; nearly as long as the median web-browsing session (30 minutes) [Mozilla 2016]. In contrast, automated feedback systems are instantaneous, but traditionally limited to syntactic, rather than content quality, issues [Madnani and Cahill 2014; Microsoft 2016; Google 2016]. Recent automated systems have highlighted the power of using machine learning features to generate feedback. These techniques identify features of the user's text that indicate low-quality, and translate those features into general feedback text for the entire document [Krause 2015b; Krause et al. 2016; Biran and McKeown 2014]. They have shown that simple identificaton procedures—looking for outlier features whose value differs from typical values in high quality text, or using features weights in linear models—can generate helpful feedback. Although promising, there are a number of potential directions to improve automated feedback. Psychology studies emphasize the value of giving feedback for specific text segments rather than for an entire document [Nelson and Schunn 2007], and writing quality models show the predictive benefits of accounting for multi-feature interactions [Weimer et al. 2007; Ghose and Ipeirotis 2011]. To the best of our knowledge, we are the first to integrate these findings into an automated writing feedback system and evaluate their effects.

To this end, we have built Dialectic, which is a customizable automatic feedback generation system. It employs a *Segment, Predict, Explain* pattern that first splits the user's text into coherent segments, predicts the quality of each segment, and generates explanations at the document and segment-level for how low-quality text could be improved. The system automates common tasks such as text segmentation, model training, and deployment, and comes with a



Non-annotated review text. Feedback is shown below text. Hovering over highlights shows segment specific feedback.

Fig. 1. Example of Dialectic feedback interface.

library of default components for each task. Thus, developers can easily customize Dialectic to their domain by simply creating machine learning features and explanation generation functions. We introduce a novel perturbation-based feature selection algorithm to take feature interactions into account when generating the feedback. Although some of these components (segmentation, quality prediction) are well known, their combination into a user-facing automatic feedback system (Figure 1) does not exist. Our user study shows that feedback generated by combining segmentation and our perturbation-based technique improves the quality of participant's laptop reviews more than adding either individually to a state-of-the-art system.

In summary, our main contributions are:

(1) The design, implementation, and release of Dialectic, an automatic text feedback system that uses a *Segment, Predict, Explain* pattern to provide document and segment-level feedback for user written text. It is easily customizable to new domains by adding new machine learning features and explanation generators. The Python package can be installed via the command `pip install dialectic_pipeline`.

(2) A novel perturbation-based technique that identifies combinations of features that, if changed, will most improve the predicted quality of input text. We formalize this problem and propose an efficient heuristic to search the exponential solution space that empirically produces helpful feedback.

(3) End-to-end evaluation by training Dialectic on Amazon product reviews using existing features from the literature. We use a crowd-sourced user study and find that the unique combination of segmentation *and* perturbation-based feedback generates feedback that improves average review quality by 14.4%, which is over 3x more than improvements using an alternative state-of-the-art method.

2. SYSTEM DESIGN AND USAGE

The design of the Dialectic system mirrors its three step *Segment, Predict, and Explain* workflow (Figure 2). We implement a standard set of segmentation algorithms, and by default use TopicTiling [Riedl and Biemann 2012], which segments the text when it detects shifts in the topic distribution [Blei et al. 2003]. For prediction, Dialectic automatically trains two Random Forest classifiers [Breiman 2001]—one to predict the quality of entire documents, and the second to predict the quality of an individual segment[1]. To help bootstrap new deployments, we have implemented a library of state-of-the-art features based on a survey of existing literature [Krause 2015a; Liu et al. 2007; Tan et al. 2016; Siersdorfer et al. 2010; Ghose and Ipeirotis 2011]. Users can select from this library or add custom features appropirate for their domain; for convenience, Dialectic can use automatic feature selection algorithms to pick the most useful features for the model. If the model detects a low-quality segment or document, Dialectic uses the explain step to select a set of features and transform them into feedback text. We use a novel perturbation-based approach to select combinations of features whose values, if perturbed very little, will cause the model to re-classify the input as high quality. The key characteristic that differentiates our approach is that it accounts for perturbations of feature combinations, rather than

[1]We empirically found that random forests had the highest accuracy in our experiments

individual features in isolation [Krause et al. 2016], and that it explicitly seeks perturbation that will improve the model's quality prediction. We are able to explore this exponential space of perturbations using a heuristic algorithm that exploits the structure of random forest models. Once we identify these features, we use developer-provided explanation functions to transform the features into user-facing feedback text. The current unoptimized system is able to generate feedback for 1-5 paragraphs of text in under 1 second.

To use Dialectic for a new application, developers simply perform four steps. First, they provide a training corpus of documents (i.e reviews, posts) labeled with some form of quality metric (i.e up-votes, likes). Second, they optionally add to Dialectic a set of features (say, based on existing literature) that is well suited for their domain. Third, they implement explanation generator functions, which take as input the set of features identified by the perturbation technique, and returns a string. For example, an explanation generator for product reviews might return the string "Not enough detail, please. . ." if features relating to review informativeness (i.e jargon usage, sentence count) are passed as input to the function. Finally, Dialectic provides a simple javascript library that integrates Dialectic's backend with the developer's web app, so that developers can easily apply Dialectic to any input text-field.

Fig. 2. Dialectic splits input text into coherent segments; predicts quality for each segment and the whole text; and generates and shows explanations to improve low-quality segments and the text overall.

3. FINDINGS AND CONCLUSION

We conducted a crowdsourced study (N=85) that asked users to write a laptop review, showed them auto-generated feedback under four different system conditions, gave them the choice to optionally improve their reviews using that feedback, and measured the extent to which the writer improved their review post-feedback. The two independent variables that we varied were *granularity*—whether the feedback was provided at the document level, or at the document *and* segment level—and *explanation selection*—whether Dialectic's perturbation-based analysis was used to select feedback to show users, or feedback was shown using a state-of-the art feedback system [Krause 2015b]. This resulted in a 2x2 between-subjects design. The system used 47 features and 4 explanation functions based on existing product review and writing quality literature, and was trained on a corpus of Amazon product reviews. We trained the segment-level model by labeling each segment with the helpfulness of the document it constituted, and validated this assumption by testing the classifier on segment labels generated in a separate crowdsourced study. We recruited and trained three independent non-authors to code the review quality using a rubric based on prior literature [Minqing Hu 2004].

Combining segmentation *and* perturbation-based explanation outperformed all other conditions by a statistically significant margin. Perturbation-analysis was shown to cause an independent statistically significant improvement, while segmentation alone did not. The effect sizes of each variables in isolation were smaller than their combination; Dialectic, which combines segmentation and perturbation-analysis, improved the overall writing quality measure by nearly $3.9\times$ over the baseline, and $2.4\times$ over the next-best condition (no segmentation with perturbation analysis). These results show that while segment-level feedback that takes into account multi-feature interactions is very helpful, segment-level feedback is not useful when the explanations do not take multi-feature interactions into account, and perturbation-based feedback is less useful when provided only at the overall document level. These results demonstrate the value of using the *Segment, Predict, Explain* pattern for creating automated interfaces that provide targeted, complex feedback to improve user-generated online writing.

Writing is central to our online lives—whether is sharing our opinions, reviewing a product, or chatting with friends and colleagues. We plan to explore how the Segment, Predict, Explain pattern can be tailored to other forms of online communication. Consider a real-time policital conversation (e.g., mest.io). The predictive variables may be tuned towards improving civility of discourse rather than traditional quality measures. Similarly, feedback design can play a part in not only explaining the writing quality, but affecting the *way* users communicate.

REFERENCES

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding high-quality content in social media. In *WSDM*.

Amazon. 2016. Amazon: Community Guidelines. https://www.amazon.com/gp/help/customer/display.html?nodeId=201929730. (2016).

Tim Berners-Lee. 1997. Realising the Full Potential of the Web. https://www.w3.org/1998/02/Potential.html. (1997).

Or Biran and Kathleen McKeown. 2014. Justification narratives for individual classifications. In *AutoML*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. In *JMLR*.

Amiangshu Bosu, Christopher S Corley, Dustin Heaton, Debarshi Chatterji, Jeffrey C Carver, and Nicholas A Kraft. 2013. Building reputation in stackoverflow: an empirical investigation. In *MSR*.

Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

Anindya Ghose and Panagiotis G Ipeirotis. 2011. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. In *TKDE*. IEEE.

Google. 2016. Check spelling and grammar in Google Docs. support.google.com/docs/answer/57859. (2016).

Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. In *HCI*.

Markus Krause. 2015a. Bull-O-Meter: Predicting the Quality of Natural Language Responses. In *HCOMP*.

Markus Krause. 2015b. A Method to automatically choose Suggestions to Improve Perceived Quality of Peer Reviews based on Linguistic Features.

James A Kulik and Chen-Lin C Kulik. 1988. Timing of feedback and verbal learning. *Review of educational research* (1988).

Chinmay E. Kulkarni, Michael S. Bernstein, and Scott R. Klemmer. 2015. PeerStudio: Rapid Peer Feedback Emphasizes Revision and Improves Performance. In *ACM*.

Jingjing Liu, Yunbo Cao, Chin-Yew Lin, Yalou Huang, and Ming Zhou. 2007. Low-Quality Product Review Detection in Opinion Summarization.. In *EMNLP-CoNLL*.

Nitin Madnani and Aoife Cahill. 2014. An Explicit Feedback System for Preposition Errors based on Wikipedia Revisions. In *NAACL*.

Microsoft. 2016. Check spelling and grammar in Office 2010 and later. support.office.com. (2016).

Bing Liu Minqing Hu. 2004. Mining opinion features in customer reviews. In *AAAI*.

Mozilla. 2016. Browsing Sessions. https://blog.mozilla.org/metrics/2010/12/22/browsing-sessions/. (2016).

M Nelson and C Schunn. 2007. The nature of feedback: Investigating how different Types of feedback affect writing performance. In *Learning Research and Development Center*.

Martin Riedl and Chris Biemann. 2012. TopicTiling: a text segmentation algorithm based on LDA. In *ACL*.

Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. 2010. How useful are your comments?: analyzing and predicting youtube comments and comment ratings. In *WWW*.

Nikita Spirin and Jiawei Han. 2012. Survey on web spam detection: principles and algorithms. In *KDD*. ACM.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *WWW*.

Jiliang Tang, Xia Hu, and Huan Liu. 2013. Social recommendation: a review. In *SNAM*. Springer.

Unauthored. 2016. Mest. mest.io. (2016).

Gang Wang, Konark Gill, Manish Mohanlal, Haitao Zheng, and Ben Y Zhao. 2013. Wisdom in the social crowd: an analysis of quora. In *WWW*.

Markus Weimer, Iryna Gurevych, and Max Mühlhäuser. 2007. Automatically Assessing the Post Quality in Online Discussions on Software. In *ACL*.

Wikipedia. 2016. Editor guidelines. en.wikipedia.org/wiki/Wikipedia:Policies_and_guidelines. (2016).

Yelp. 2016a. Applying to be Elite. yelp.com/elite. (2016).

Yelp. 2016b. Content Guidelines. yelp.com/guidelines. (2016).