Provocation: Contestability in Large-Scale Interactive NLP Systems

Tanushree Mitra
The Information School
University of Washington
tmitra@uw.edu

1 Overview

Designing computational systems for language analysis means, increasingly, designing for interactions with natural language processing (NLP) algorithms. For example, sentiment analysis, topic modeling, toxicity classification, and other language modeling techniques have become common in interactive user-facing systems. These developments raise novel, complex challenges, both in terms of designing such systems and user's interactions with them (Baumer et al., 2020). Accordingly researchers have advocated for rethinking interactive ML systems that involve users at every stage of the system (Amershi et al., 2014). For example, Horvitz' principles for effective "mixed-initiative" systems include querying users about goals and preferences and scoping system precision to match users' needs (Horvitz, 1999). Other approaches include value-centered design (Knobel and Bowker, 2011), transparency in design (Ananny and Crawford, 2018) and more recently contestability (Hirsch et al., 2017). Despite these various proposed approaches for ML-based systems, situating the discussion in the context of interactive NLP systems, in particular, has remained elusive. Each of these design principles (mixed-initiative, value-centered, transparency, contestability) can claim it's own spot for a full-day workshop discussion. Hence, to scope the conversation for this workshop, we will primarily focus on **contestability in NLP systems** in this provocation piece.

Contestability: The notion of contestability originates from the need for contesting or challenging machine predictions (Hirsch et al., 2017; Kluttz et al., 2018). For example, contestability in recidivism prediction systems or health behavior prediction systems. Here we argue the need for exploring contestability in large-scale online systems and in particular NLP systems. With respect to language-based systems, consider the Perspective API developed by Google's Counter-Abuse Technology team to identify toxic language in text. The technology has now been integrated within the New York Times comment interface to facilitate large-scale moderation of potentially toxic and obscene comments on news stories. However, the same technology has also incorrectly discovered a positive correlation between identity terms containing information on race or sexual orientation (e.g., the phrase "I am a gay black woman" received a high toxicity score) (developers.google.com). But we do not yet have an established way to contest these decisions made by NLP algorithms integrated within large socio-technical systems (in this case a news platforms with widespread readership).

Perhaps the closest form of contestability research in large-scale online systems relates to efforts around auditing algorithmic systems for bias (Bozdag, 2013; Chen et al., 2018), discrimination (Chen et al., 2016; Hannak et al., 2014; Mikians et al., 2012), and fairness (Dwork et al., 2012). While audit studies is a way to detect undesirable behavior in large-scale "black-boxed" Web systems, allowing users to meaningfully *contest* such undesirable behavior will offer ways to help people not only make sense of an algorithm's behavior, but also restore human agency in systems that are intertwining humans and non-human agents (Ananny and Crawford, 2018). *How can we design for contestability in large-scale online NLP systems? What are the goals when designing for contestability in online systems? What are the challenges and limitations of the contestability ideal?*

2 Designing Contestability in Large Online NLP Systems

The first step in designing for contestability is to layout the goals we would like to achieve when

Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing, pages 96–100

bringing the ideal of contestability in large-scale online socio-technical systems driven by language technologies. In most predictive systems, the goal is to contest a machine prediction in an attempt to correct a wrong decision. For example, in healthcare systems, contestability strives to improve the accuracy of ML models by deploying the system among expert users and then soliciting feedback from them (Hirsch et al., 2017). However, the goal in large online social systems is more nuanced. For example, in an online ride sharing system a driver might want to contest their ride and route recommendation, a rider might want to contest the tip suggestion or the matched driver, resulting in a multi-stakeholder contestability problem. In this particular case, it is a complex assemblage of the algorithm and the two stake holders (the driver and the rider) who are both using the same instance of the ride sharing platform.

Here I outline two examples of NLP-based online systems along with the phenomenon that could be contested and the contestation goal.

Contestability in Moderation Systems. Content moderation is a predominant practice in almost all social media platforms, be it, Twitter, Facebook, Reddit, and is now increasingly common in news platforms (e.g. Perspective APIs usage in NYT comment moderation interface). Most content moderation systems are based on linguistic models (see (Gunasekara and Nejadgholi, 2018) for a review). However, none currently have designs in place where a user can appeal against content removal. If the *goal* is to contest moderation practices, how do we design to meet that goal? Some social media platforms, like Reddit, have a two-tier governance structure, the first tier enforced by the platform's content policy and a 2nd tier rule enforcement set by the human moderators within the community. Such differences in governance structures across online platforms make it almost impossible to come up with a "one size fits all" contestation goal when building a language-based moderation system.

Contestability in Informatin Retrieval Systems. Language models are fundamental to driving online information retrieval systems, such as search and recommendation systems (see (Zhai, 2008; Hiemstra, 2001; Liu and Croft, 2004)). For a regular user, online search and recommender systems have become an integral part of their daily lives. Despite their increasingly important role in selecting, presenting, ranking, and recommending what information is considered most relevant for us—a key aspect governing our ability to meaningfully participate in public life (Gillespie, 2014), there is no notion of whether this information is credible or whether the returned results are re-inforcing existing societal biases. Neither do users have any means to contest or challenge to understand *why* they are seeing *what* they are seeing on search platforms and their recommendation feeds. The lack of contestability in search platform coupled with an unwavering trust placed in search engines can together lead to misinformed citizenry. Our goal for including contestability in search systems, would be to contest potential inaccurate results, unreasonable rankings (e.g., a low credible alternative news source ranked higher than a reputed journalistic source), inaccurate recommendations (for e.g., YouTube recommending more pro-conspiracy videos after user watches one anti-vaccine video) or biased results.

2.1 *Why, When*, and *How* to Contest

What are the ways in which we can design for contestability in online systems? Does contestability help users improve their understanding and trust of large-scale online systems? One possibility is to draw inspiration from the design of intelligibility in context-aware systems (Lim et al., 2009). Of particular interest are end-user programming systems which allowed users to ask questions when their expectations were not met (Ko and Myers, 2004). Users asked *why did* questions when something unexpected happened and *why not* questions when something expected did not occur (Ko and Myers, 2004, 2009). Another line of work that is relevant is (Kulesza et al., 2011, 2009) *What You See is What You Test* for Machine Learning (WYSIWYT/ML) method for systematic testing of machine learning applications. WYSIWYT/ML offers three key functionalities: 1) *advises* the user about which predictions to test, 2) *contributes* more tests "like" the user's, 3) *measures* how much of the assistant's reasoning has been tested, and 4) continually *monitors* over time whether previous testing still "covers" new behaviors learned. Inspired from these approaches, here I lay a list of intuitive questions that can inform the design of contestability. By no means, this list is exhaustive. The hope is that discussions at the workshop will

refine and expand the list.

1. **Why**: Why did the online system do X, where X can be *recommend, suggest, rank, moderate, etc.*?

2. **Why Not**: Why did the system not do Y? For example, in designing contestability for content moderation in social platforms, a user can contest why did the system did not remove this other post?

3. **What if**: What would the system do if Z happens? For example, what would Amazon recommend if I bought the lower ranked product from its ranked product list?

4. **How to**: How can I get the system to do A, given the current context? For example, how can I get this ride sharing service to pair me with a female driver, given that I am a woman and I am traveling in a region notorious for crimes against woman?

3 Challenges for Contestability in Large Online NLP Systems

Infusing contestability in complex online systems can lead to unforeseen challenges. The ideal of contestability as a means to provide agency to users can be limited in multiple ways. Here I list a few.

Contestability can unintentionally occlude. Ananny and Crawford (Ananny and Crawford, 2018) in their critical interrogation of the ideal of transparency argue that transparency to promote understandibility of black-box ML systems can backfire in various ways, one of them being intentional occlusion—"visibility produces such great quantities of information that important pieces of information become inadvertently hidden in the detritus of the information made visible." Contestability can also result in similar occlusion. For example, an online system that offers too many options for contesting the system's decision may unnecessarily distract the user from the central information that the system intends to offer.

Contestability can reinforce pre-existing biases. Considering that most algorithms driving large-scale online systems are personalized—i.e., they serve results based on user's past behavior—such personalization tend to re-enforce pre-existing biases. Imagine a factuality model that sits within an online system and determines the credibility of content based on linguistic features (such models already exist in the literature; see (Soni et al., 2014) and (Mitra et al., 2017). Offering real-time corrections every time the user contests search results that goes against their pre-existing beliefs, often act as mandates and can backfire by inadvertently provoking users into attitude consistent misperceptions (Garrett and Weeks, 2013). Another unforeseen scenario can occur for certain classes of search queries, popularly known as "data voids"—search terms for which the available relevant data is limited or non-existent (Golebiewski and boyd, 2018). Allowing the user to contest and introduce problematic corrections into the system defeats the purpose of contestability in such scenarios.

Contestability has temporal limitations: When to introduce contestability? Decisions about when to allow contestability in an online social system can get too complex too quickly to reach any useful design decision. A large-scale social system has multiple stakeholders, each of whose contestability action at any point in time can effect the system and in turn the contestability decision of the next action. For example, a language model working in the moderation interface of a news website has to balance between several stakeholders: the *users* commenting on the news story, the *moderator* managing the commenting system, the *reporter* who wrote the story, and the *editor* who reviewed the story. Moreover online systems are continually changing over time as data are being generated, interacted, and added by users, and as the number of users interacting with the system change. Thus when thinking about designing for contestability in multi-stakeholder online systems, the notion of temporal dimension is key.

I hope that the workshop will provide opportunities to brainstorm on these complex questions and stir new questions in the domain.

References

Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *Ai Magazine*, 35(4):105–120.

Mike Ananny and Kate Crawford. 2018. Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *new media & society*, 20(3):973–989.

Eric PS Baumer, Drew Siedel, Lena McDonnell, Jiayun Zhong, Patricia Sittikul, and Micki McGee.

2020. Topicalizer: reframing core concepts in machine learning visualization by co-designing for interpretivist scholarship. *Human–Computer Interaction*, 35(5-6):452–480.

Engin Bozdag. 2013. Bias in algorithmic filtering and personalization. *Ethics and information technology*, 15(3):209–227.

Le Chen, Ruijun Ma, Anikó Hannák, and Christo Wilson. 2018. Investigating the impact of gender on rank in resume search engines. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 651. ACM.

Le Chen, Alan Mislove, and Christo Wilson. 2016. An empirical analysis of algorithmic pricing on amazon marketplace. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1339–1349. International World Wide Web Conferences Steering Committee.

developers.google.com. Ml practicum: Fairness in perspective api. https://developers.google.com/machine-learning/practica/fairness-indicators. [Online; accessed 8-Jan-2021].

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM.

R Kelly Garrett and Brian E Weeks. 2013. The promise and peril of real-time corrections to political misperceptions. In *Proc. CSCW*, pages 1047–1058. ACM.

Tarleton Gillespie. 2014. The relevance of algorithms. *Media technologies: Essays on communication, materiality, and society*, 167.

Michael Golebiewski and danah boyd. 2018. Data voids: Where missing data can easily be exploited.

Isuru Gunasekara and Isar Nejadgholi. 2018. A review of standard text classification practices for multi-label toxicity identification of online content. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*, pages 21–25.

Aniko Hannak, Gary Soeller, David Lazer, Alan Mislove, and Christo Wilson. 2014. Measuring price discrimination and steering on e-commerce web sites. In *Proceedings of the 2014 conference on internet measurement conference*, pages 305–318. ACM.

Djoerd Hiemstra. 2001. *Using language models for information retrieval*. Citeseer.

Tad Hirsch, Kritzia Merced, Shrikanth Narayanan, Zac E Imel, and David C Atkins. 2017. Designing contestability: Interaction design, machine learning, and mental health. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, pages 95–99. ACM.

Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166.

Daniel Kluttz, Nitin Kohli, and Deirdre K Mulligan. 2018. Contestability and professionals: From explanations to engagement with algorithmic systems. *Available at SSRN 3311894*.

Cory Knobel and Geoffrey C Bowker. 2011. Values in design. *Communications of the ACM*, 54(7):26–28.

Andrew J Ko and Brad A Myers. 2004. Designing the whyline: a debugging interface for asking questions about program behavior. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 151–158. ACM.

Andrew J Ko and Brad A Myers. 2009. Finding causes of program output with the java whyline. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1569–1578. ACM.

Todd Kulesza, Margaret Burnett, Simone Stumpf, Weng-Keen Wong, Shubhomoy Das, Alex Groce, Amber Shinsel, Forrest Bice, and Kevin McIntosh. 2011. Where are my intelligent assistant's mistakes? a systematic testing approach. In *International Symposium on End User Development*, pages 171–186. Springer.

Todd Kulesza, Weng-Keen Wong, Simone Stumpf, Stephen Perona, Rachel White, Margaret M Burnett, Ian Oberst, and Andrew J Ko. 2009. Fixing the program my computer learned: Barriers for end users, challenges for the machine. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 187–196. ACM.

Brian Y Lim, Anind K Dey, and Daniel Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2119–2128. ACM.

Xiaoyong Liu and W Bruce Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193.

Jakub Mikians, László Gyarmati, Vijay Erramilli, and Nikolaos Laoutaris. 2012. Detecting price and search discrimination on the internet. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, pages 79–84. acm.

Tanushree Mitra, Graham Wright, and Eric Gilbert. 2017. A parsimonious language model of social media credibility across disparate events. In *Proc. CSCW*.

Sandeep Soni, Tanushree Mitra, Eric Gilbert, and Jacob Eisenstein. 2014. Modeling factuality judgments in social media text. In *ACL (2)*, pages 415–420.

ChengXiang Zhai. 2008. Statistical language models for information retrieval. *Synthesis lectures on human language technologies*, 1(1):1–141.