Automatically Neutralizing Subjective Bias in Text

Abstract

Texts like news, encyclopedias, and some social media strive for objectivity. Yet bias in the form of inappropriate subjectivity - introducing attitudes via framing, presupposing truth, and casting doubt - remains ubiquitous. This kind of bias erodes our collective trust and fuels social conflict. To address this issue, we introduce a novel testbed for natural language generation: automatically bringing inappropriately subjective text into a neutral point of view ("neutralizing" biased text). We also offer the first parallel corpus of biased language. The corpus contains 180,000 sentence pairs and originates from Wikipedia edits that removed various framings, presuppositions, and attitudes from biased sentences. Last, we propose two strong encoder-decoder baselines for the task. A straightforward yet opaque CONCURRENT system uses a BERT encoder to identify subjective words as part of the generation process. An interpretable and controllable MODULAR algorithm separates these steps, using (1) a BERT-based classifier to identify problematic words and (2) a novel join embedding through which the classifier can edit the hidden states of the encoder. Large-scale human evaluation across four domains (encyclopedias, news headlines, books, and political speeches) suggests that these algorithms are a first step towards the automatic identification and reduction of bias.

Introduction

Writers and editors of texts like encyclopedias, news, and textbooks strive to avoid biased language. Yet bias remains ubiquitous. 62% of Americans believe their news is biased BIBREF0 and bias is the single largest source of distrust in the media BIBREF1.

This work presents data and algorithms for automatically reducing bias in text. We focus on a particular kind of bias: inappropriate subjectivity (“subjective bias”). Subjective bias occurs when language that should be neutral and fair is skewed by feeling, opinion, or taste (whether consciously or unconsciously). In practice, we identify subjective bias via the method of BIBREF2: using Wikipedia's neutral point of view (NPOV) policy. This policy is a set of principles which includes “avoiding stating opinions as facts” and “preferring nonjudgemental language”.

For example a news headline like “John McCain exposed as an unprincipled politician" (Figure FIGREF1) is biased because the verb expose is a factive verb that presupposes the truth of its complement; a non-biased sentence would use a verb like describe so as not to presuppose something that is the subjective opinion of the writer. “Pilfered” in “the gameplay is pilfered from DDR” (Table TABREF3) subjectively frames the shared gameplay as a kind of theft. “His” in “a lead programmer usually spends his career” again introduces a biased and subjective viewpoint (that all programmers are men) through presupposition.

We aim to debias text by suggesting edits that would make it more neutral. This contrasts with prior research which has debiased representations of text by removing dimensions of prejudice from word embeddings BIBREF3, BIBREF4 and the hidden states of predictive models BIBREF5, BIBREF6. To avoid overloading the definition of “debias,” we refer to our kind of text debiasing as neutralizing that text. Figure FIGREF1 gives an example.

We introduce the Wiki Neutrality Corpus (WNC). This is a new parallel corpus of 180,000 biased and neutralized sentence pairs along with contextual sentences and metadata. The corpus was harvested from Wikipedia edits that were designed to ensure texts had a neutral point of view. WNC is the first parallel corpus targeting biased and neutralized language. We also define the task of neutralizing subjectively biased text. This task shares many properties with tasks like detecting framing or epistemological bias BIBREF2, or veridicality assessment/factuality prediction BIBREF7, BIBREF8, BIBREF9, BIBREF10. Our new task extends these detection/classification problems into a generation task: generating more neutral text with otherwise similar meaning.

Finally, we propose a pair of novel sequence-to-sequence algorithms for this neutralization task. Both methods leverage denoising autoencoders and a token-weighted loss function. An interpretable and controllable modular algorithm breaks the problem into (1) detection and (2) editing, using (1) a BERT-based detector to explicitly identify problematic words, and (2) a novel join embedding through which the detector can modify an editors' hidden states. This paradigm advances an important human-in-the-loop approach to bias understanding and generative language modeling. Second, an easy to train and use but more opaque concurrent system uses a BERT encoder to identify subjectivity as part of the generation process.

Large-scale human evaluation suggests that while not without flaws, our algorithms can identify and reduce bias in encyclopedias, news, books, and political speeches, and do so better than state-of-the-art style transfer and machine translation systems. This work represents an important first step towards automatically managing bias in the real world. We release data and code to the public.

Wiki Neutrality Corpus (WNC)

The Wiki Neutrality Corpus consists of aligned sentences pre and post-neutralization by English Wikipedia editors (Table TABREF3). We used regular expressions to crawl 423,823 Wikipedia revisions between 2004 and 2019 where editors provided NPOV-related justification BIBREF11, BIBREF2, BIBREF12. To maximize the precision of bias-related changes, we ignored revisions where

[noitemsep]

More than a single sentence was changed.

Minimal edits (character Levenshtein distance $<$ 4).

Maximal edits (more than half of the words changed).

Edits where more than half of the words were proper nouns.

Edits that fixed spelling or grammatical errors.

Edits that added references or hyperlinks.

Edits that changed non-literary elements like tables or punctuation.

We align sentences in the pre and post text by computing a sliding window (of size $k = 5$) of pairwise BLEU BIBREF13 between sentences and matching sentences with the biggest score BIBREF14, BIBREF15. Last, we discarded pairs whose length ratios were beyond the 95th percentile BIBREF16.

Corpus statistics are given in Table TABREF12. The final data are (1) a parallel corpus of 180k biased sentences and their neutral counterparts, and (2) 385k neutral sentences that were adjacent to a revised sentence at the time of editing but were not changed by the editor. Note that following BIBREF2, the neutralizing experiments in Section SECREF4 focus on the subset of WNC where the editor modified or deleted a single word in the source text (“Biased-word” in Table TABREF12).

Table TABREF12 also gives a categorization of these sample pairs using a slight extension of the typology of BIBREF2. They defined framing bias as using subjective words or phrases linked with a particular point of view (like using words like best or deepest or using pilfered from instead of based on, and epistemological bias as linguistic features that subtly (often via presupposition) focus on the believability of a proposition. We add to their two a third kind of subjectivity bias that also occurs in our data, which we call demographic bias, text with presuppositions about particular genders or races or other demographic categories (like presupposing that all programmers are male).

The dataset does not include labels for these categories, but we hand-labeled a random sample of 500 examples to estimate the distribution of the 3 types. Table TABREF13 shows that while framing bias is most common, all types of bias are represented in the data, including instances of demographic bias.

Wiki Neutrality Corpus (WNC) ::: Dataset Properties

We take a closer look at WNC to identify characteristics of subjective bias on Wikipedia.

Topic. We use the Wikimedia Foundation's categorization models BIBREF17 to bucket articles from WNC and the aforementioned random sample into a 44-category ontology, then compare the proportions of NPOV-driven edits across categories. Subjectively biased edits are most prevalent in history, politics, philosophy, sports, and language categories. They are least prevalent in the meteorology, science, landforms, broadcasting, and arts categories. This suggests that there is a relationship between a text's topic and the realization of bias. We use this observation to guide our model design in Section SECREF19.

Tenure. We group editors into “newcomers” (less than a month of experience) and “experienced” (more than a month). We find that newcomers are less likely to perform neutralizing edits (15% in WNC) compared to other edits (34% in a random sample of 685k edits). This difference is significant ($\tilde{\chi }^2$ p $=$ 0.001), suggesting the complexity of neutralizing text is typically reserved for more senior editors, which helps explain the performance of human evaluators in Section SECREF53.

Methods for Neutralizing Text

We propose the task of neutralizing text, in which the algorithm is given an input sentence and must produce an output sentence whose meaning is as similar as possible to the input but with the subjective bias removed.

We propose two algorithms for this task, each with its own benefits. A modular algorithm enables human control and interpretability. A concurrent algorithm is simple to train and operate.

We adopt the following notation:

$\mathbf {s} = [w^s_1, ..., w^s_n]$ is a source sequence of subjectively biased text.

$\mathbf {t} = [w^t_1, ..., w^t_m]$ is a target sequence and the neutralized version of $\mathbf {s}$.

Methods for Neutralizing Text ::: MODULAR

The first algorithm we are proposing has two stages: BERT-based detection and LSTM-based editing. We pretrain a model for each stage and then combine them into a joint system for end-to-end fine tuning on the overall neutralizing task. We proceed to describe each module.

Methods for Neutralizing Text ::: MODULAR ::: Detection Module

The detection module is a neural sequence tagger that estimates p_i, the probability that each input word w^s_i is subjectively biased (Figure FIGREF26).

Module description. Each p_i is calculated according to

$\mathbf {b}_i \in \mathcal {R}^{b}$ represents w^s_i's semantic meaning. It is a contextualized word vector produced by BERT, a transformer encoder that has been pre-trained as a masked language model BIBREF18. To leverage the bias-topic relationship uncovered in Section SECREF14, we prepend a token indicating an article's topic category (<arts>, <sports>, etc) to $\mathbf {s}$. The word vectors for these tokens are learned from scratch.

$\mathbf {e}_i$ represents expert features of bias proposed by BIBREF2:

$\mathbf {W}^{in} \in \mathcal {R}^{f \times h}$ is a matrix of learned parameters, and $\mathbf {f}_i$ is a vector of discrete features.

$\mathbf {W}^{b} \in \mathcal {R}^{b}$, $\mathbf {W}^{e} \in \mathcal {R}^{h}$, and $b \in \mathcal {R}$ are learnable parameters.

Module pre-training. We train this module using diffs between the source and target text. A label p^*_i is 1 if w^s_i was deleted or modified as part of the neutralizing process. A label is 0 if it occurs in both the source and target text. The loss is calculated as the average negative log likelihood of the labels:

Methods for Neutralizing Text ::: MODULAR ::: Editing Module

The editing module takes a subjective source sentence $\mathbf {s}$ and is trained to edit it into a more neutral compliment $\mathbf {t}$.

Module description. This module is based on a sequence-to-sequence neural machine translation model BIBREF19. A bi-LSTM BIBREF20 encoder turns $\mathbf {s}$ into a sequence of hidden states $\mathbf {H} = (\mathbf {h}_1, ..., \mathbf {h}_n)$. Next, an LSTM decoder generates text one token at a time by repeatedly attending to $\mathbf {H}$ and producing probability distributions over the vocabulary. We also add two mechanisms from the summarization literature BIBREF21. The first is a copy mechanism, where the model's final output for timestep i becomes a weighted combination of the predicted vocabulary distribution and attentional distribution from that timestep. The second is a coverage mechanism which incorporates the sum of previous attention distributions into the final loss function to discourage the model from re-attending to a word and repeating itself.

Module pre-training. We pre-train the decoder as a language model of neutral text using the neutral portion of WNC (Section SECREF2). Doing so expresses a data-driven prior about how target sentences should read. We accomplish this with a denoising autoencoder objective BIBREF22 and maximizing the conditional log probability $\log p(\mathbf {x} \vert \widetilde{\mathbf {x}})$ of reconstructing a sequence $\mathbf {x}$ from a corrupted version of itself $\widetilde{\mathbf {x}} = C(\mathbf {x})$ using noise model C.

Our C is similar to BIBREF23. We slightly shuffle $\mathbf {x}$ such that x_i's index in $\widetilde{\mathbf {x}}$ is randomly selected from $[i - k, i + k]$. We then drop words with probability p. For our experiments, we set $k = 3$ and $p = 0.25$.

Methods for Neutralizing Text ::: MODULAR ::: Final System

Once the detection and editing modules have been pre-trained, we join them and fine-tune together as an end to end system for translating $\mathbf {s}$ into $\mathbf {t}$.

This is done with a novel join embedding mechanism that lets the detector control the editor (Figure FIGREF29). The join embedding is a vector $\mathbf {v} \in \mathcal {R}^h$ that we add to each encoder hidden state in the editing module. This operation is gated by the detector's output probabilities $\mathbf {p} = (p_1, ..., p_n)$. Note that the same $\mathbf {v}$ is applied across all timesteps.

We proceed to condition the decoder on the new hidden states $\mathbf {H}^{\prime } = (\mathbf {h^{\prime }}_1, ..., \mathbf {h}^{\prime }_n)$. Intuitively, $\mathbf {v}$ is enriching the hidden states of words that the detector identified as subjective. This tells the decoder what language should be changed and what is safe to be be copied during the neutralization process. Error signals are allowed to flow backwards into both the encoder and detector, creating an end-to-end system from the two modules.

To fine-tune the parameters of the joint system, we use a token-weighted loss function that scales the loss on neutralized words (i.e. words unique to $\mathbf {t}$) by a factor of $\alpha $:

Note that c is a term from the coverage mechanism (Section SECREF28). We use $\alpha = 1.3$ in our experiments. Intuitively, this loss function incorporates an inductive bias of the neutralizing process: the source and target have a high degree of lexical similarity but the goal is to learn the structure of their differences, not simply copying words into the output (something a pre-trained autoencoder should already have knowledge of). This loss function is related to previous work on grammar correction BIBREF24, and cost-sensitive learning BIBREF25.

Methods for Neutralizing Text ::: CONCURRENT

Our second algorithm takes the problematic source $\textbf {s}$ and directly generates a neutralized $\mathbf {\hat{t}}$. While this renders the system easier to train and operate, it limits interpretability and controllability.

Model description. The concurrent system is an encoder-decoder neural network. The encoder is BERT. The decoder is the same as that of Section SECREF28: an attentional LSTM with copy and coverage mechanisms. The decoder's inputs are set to:

Hidden states $\mathbf {H} = \mathbf {W}^H\ \mathbf {B}$, where $\mathbf {B} = (\mathbf {b}_1, ..., \mathbf {b}_{n}) \in \mathcal {R}^{b \times n}$ is the BERT-embedded source and $\mathbf {W}^H \in \mathcal {R}^{h \times b}$ is a matrix of learned parameters.

Initial states $\mathbf {c}_0 = \mathbf {W}^{c0}\ \sum \mathbf {b}_i / n$ and $\mathbf {h_0} = \mathbf {W}^{h0}\ \sum \mathbf {b}_i / n$. $\mathbf {W}^{c0} \in \mathcal {R}^{h \times b}$ and $\mathbf {W}^{h0} \in \mathcal {R}^{h \times b}$ are learned matrices.

Model training. The concurrent model is pre-trained with the same autoencoding procedure described in Section SECREF28. It is then fine-tuned as a subjective-to-neutral translation system with the same loss function described in Section SECREF30.

Experiments ::: Experimental Protocol

Implementation. We implemented nonlinear models with Pytorch BIBREF29 and optimized using Adam BIBREF30 as configured in BIBREF18 with a learning rate of 5e-5. We used a batch size of 16. All vectors were of length $h = 512$ unless otherwise specified. We use gradient clipping with a maximum gradient norm of 3 and a dropout probability of 0.2 on the inputs of each LSTM cell BIBREF31. We initialize the BERT component of the tagging module with the publicly-released bert-base-uncased parameters. All other parameters were uniformly initialized in the range $[-0.1,\ 0.1]$.

Procedure. Following BIBREF2, we train and evaluate our system on the subset of WNC where the editor changed or deleted a single word in the source text. This yielded 53,803 training pairs (about a quarter of the WNC), from which we sampled 700 development and 1,000 test pairs. For fair comparison, we gave our baselines additional access to the 385,639 neutral examples when possible. We pretrained the tagging module for 4 epochs. We pretrained the editing module on the neutral portion of our WNC for 4 epochs. The joint system was trained on the same data as the tagger for 25,000 steps (about 7 epochs). We perform interference using beam search and a beam width of 4. All computations were performed on a single NVIDIA TITAN X GPU; training the full system took approximately 10 hours. We report statistical significance with bootstrap resampling and a 95% confidence level BIBREF32, BIBREF33.

Evaluation. We evaluate our models according to five metrics. BLEU BIBREF13 and accuracy (the proportion of decodings that exactly matched the editors changes) are quantitative. We also hired fluent English-speaking crowdworkers on Amazon Mechanical Turk. Workers were shown the BIBREF2 and Wikipedia definition of a “biased statement” and six example sentences, then subjected to a five-question qualification test where they had to identify subjectivity bias. Approximately half of the 30,000 workers who took the qualification test passed. Those who passed were asked to compare pairs of original and edited sentences (not knowing which was the original) along three criteria: fluency, meaning preservation, and bias. Fluency and bias were evaluated on a Semantic Differential scale from -2 to 2. Here, a semantic differential scale can better evaluate attitude oriented questions with two polarized options (e.g., “is text A or B more fluent?”). Meaning was evaluated on a Likert scale from 0 to 4, ranging from “totally different” to “identical”. Inter-rater agreement was fair to substantial (Krippendorff's alpha of 0.65 for fluency, 0.33 for meaning, and 0.51 for bias). We report statistical significance with a t-test and 95% confidence interval.

Experiments ::: Wikipedia (WNC)

Results on WNC are presented in Table TABREF35. In addition to methods from the literature we include (1) a BERT-based system which simply predicts and deletes subjective words, and (2) a system which predicts replacements (including deletion) for subjective words directly from their BERT embeddings. All methods appear to successfully reduce bias according to the human evaluators. However, many methods appear to lack fluency. Adding a token-weighted loss function and pretraining the decoder help the model's coherence according to BLEU and accuracy. Adding the detector (modular) or a BERT encoder (concurrent) provide additional benefits. The proposed models retain the strong effects of systems from the literature while also producing target-level fluency on average. Our results suggest there is no clear winner between our two proposed systems. modular is better at reducing bias and has higher accuracy, while concurrent produces more fluent responses, preserves meaning better, and has higher BLEU.

Table TABREF39 indicates that BLEU is more correlated with fluency but accuracy is more correlated with subjective bias reduction. The weak association between BLEU and human evaluation scores is corroborated by other research BIBREF35, BIBREF36. We conclude that neither automatic metric is a true substitute for human judgment.

Experiments ::: Real-world Media

To demonstrate the efficacy of the proposed methods on subjective bias in the wild, we perform inference on three out-of-domain datasets (Table TABREF45). We prepared each dataset according to the same procedure as WNC (Section SECREF2). After inference, we enlisted 1800 raters to assess the quality of 200 randomly sampled datapoints. Note that for partisan datasets we sample an equal number of examples from “conservative” and “liberal” sources. These data are:

The Ideological Books Corpus (IBC) consisting of partisan books and magazine articles BIBREF37, BIBREF38.

Headlines of partisan news articles identified as biased according to mediabiasfactcheck.com.

Sentences from the campaign speeches of a prominent politician (United States President Donald Trump). We filtered out dialog-specific artifacts (interjections, phatics, etc) by removing all sentences with less than 4 tokens before sampling a test set.

Overall, while modular does a better job at reducing bias, concurrent appears to better preserve the meaning and fluency of the original text. We conclude that the proposed methods, while imperfect, are capable of providing useful suggestions for how subjective bias in real-world news or political text can be reduced.

Error Analysis

To better understand the limits of our models and the proposed task of bias neutralization, we randomly sample 50 errors produced by our models on the Wikipedia test set and bin them into the following categories:

No change. The model failed to remove or change the source sentence.

Bad change. The model modified the source but introduced an edit which failed to match the ground-truth target (i.e. the Wikipedia editor's change).

Disfluency. Errors in language modeling and text generation.

Noise. The datapoint is noisy and the target text is not a neutralized version of the source.

The distribution of errors is given in Table TABREF50. Most errors are due to the subtlety and complexity of language understanding required for bias neutralization, rather than the generation of fluent text. These challenges are particularly pronounced for neutralizing edits that involve the replacement of factive and assertive verbs. As column 2 shows, a large proportion of the errors, though disagreeing with the edit written by the Wikipedia editors, nonetheless successfully neutralize bias in the source.

Examples of each error type are given in Table TABREF52 (two pages away). As the examples show, our models have have a tendency to simply remove words instead of finding a good replacement.

Algorithmic Analysis

We proceed to analyze our algorithm's ability to detect and categorize bias as well as the efficacy of the proposed join embedding.

Algorithmic Analysis ::: Detecting Subjectivity

Identifying subjectivity in a sentence (explicitly or implicitly) is prerequisite to neutralizing it. We accordingly evaluate our model's (and 3,000 crowdworker's) ability to detect subjectivity using the procedure of BIBREF2 and the same 50k training examples as Section SECREF4 (Table TABREF51). For each sentence, we select the word with the highest predicted probability and test whether that word was in fact changed by the editor. The proportion of correctly selected words is the system's “accuracy”. Results are given in Table TABREF51.

Note that concurrent lacks an interpretive window into its detection behavior, so we estimate an upper bound on the model's detection abilities by (1) feeding the encoder's hidden states into a fully connected + softmax layer that predicts the probability of a token being subjectively biased, and (2) training this layer as a sequence tagger according to the procedure of Section SECREF19.

The low human performance can be attributed to the difficulty of identifying bias. Issues of bias are typically reserved for senior Wikipedia editors (Section SECREF14) and untrained workers performed worse (37.39%) on the same task in BIBREF2 (and can struggle on other tasks requiring linguistic knowledge BIBREF39). concurrent's encoder, which is architecturally identical to BERT, had similar performance to a stand-alone BERT system. The linguistic and category-related features in the modular detector gave it slight leverage over the plain BERT-based models.

Algorithmic Analysis ::: Join Embedding

We continue by analyzing the abilities of the proposed join embedding mechanism.

Algorithmic Analysis ::: Join Embedding ::: Join Embedding Ablation

The join embedding combines two separately pretrained models through a gated embedding instead of the more traditional practice of stripping off any final classification layers and concatenating the exposed hidden states BIBREF40. We accordingly ablated the join embedding mechanism by training a new model where the pre-trained detector is frozen and its pre-output hidden states $\mathbf {b}_i$ are concatenated to the encoder's hidden states before decoding. Doing so reduced performance to 90.78 BLEU and 37.57 Accuracy (from the 93.52/46.8 with the join embedding). This suggests learned embeddings can be a high-performance and end-to-end conduit between sub-modules of machine learning systems.

Algorithmic Analysis ::: Join Embedding ::: Join Embedding Control

We proceed to demonstrate how the join embedding creates controllability in the neutralization process. Recall that modular relies on a probability distribution $\mathbf {p}$ to determine which words require editing (Equation DISPLAY_FORM31). Typically, this distribution comes from the detection module (Section SECREF19), but we can also feed in user-specified distributions that force the model to target particular words. This can let human advisors correct errors or push the model's behavior towards some desired outcome. We find that the model is indeed capable of being controlled, letting users target specific words for rewording in case they disagree with the model's output or seek recommendations on specific language. However, doing so can also introduce errors into downstream language generation (Table TABREF52).

Related Work

Subjectivity Bias. The study of subjectivity in NLP was pioneered by the late Janyce Wiebe and colleagues BIBREF41, BIBREF42. Several studies develop methods for highlighting subjective or persuasive frames in a text BIBREF43, BIBREF44, or detecting biased sentences BIBREF45, BIBREF46, BIBREF12, BIBREF47 of which the most similar to ours is BIBREF2, whose early, smaller version of WNC and logistic regression-based bias detector inspired our study.

Debiasing. Many scholars have worked on removing demographic prejudice from meaning representations BIBREF48, BIBREF49, BIBREF5, BIBREF50, BIBREF51. Such studies begin with identifying a direction or subspace that capture the bias and then removing such bias component to make these representations fair across attributes like gender and age BIBREF3, BIBREF48. For instance, BIBREF50 introduced a regularization term for the language model to penalize the projection of the word embeddings onto that gender subspace, while BIBREF51 used adversarial training to remove directions of bias from hidden states.

Neural Language Generation. Several studies propose stepwise procedures for text generation, including sampling from a corpus BIBREF52 and identifying language ripe for modification BIBREF53. Most similar to us is BIBREF26 who localize a text's style to a fraction of its words. Our modular detection module performs a similar localization in a soft manner, and our steps are joined by a smooth conduit (the join embedding) instead of discrete logic. There is also work related to our concurrent model. The closest is BIBREF54, where a decoder was attached to BERT for question answering, or BIBREF23, where machine translation systems are initialized to LSTM and Transformer-based language models of the source text.

Conclusion and Future Work

The growing presence of bias has marred the credibility of our news, educational systems, and social media platforms. Automatically reducing bias is thus an important new challenge for the Natural Language Processing and Artificial Intelligence community. By learning models to automatically detect and correct subjective bias in text, this work is a first step in this important direction. Nonetheless our scope was limited to single-word edits, which only constitute a quarter of the edits in our data, and are probably among the simplest instances of bias. We therefore encourage future work to tackle broader instances of multi-word, multi-lingual, and cross-sentence bias. Another important direction is integrating aspects of fact-checking BIBREF55, since a more sophisticated system would be able to know when a presupposition is in fact true and hence not subjective. Finally, our new join embedding mechanism can be applied to other modular neural network architectures.

Acknowledgements

We thank the Japan-United States Educational Commission (Fulbright Japan) for their generous support. We thank Chris Potts, Hirokazu Kiyomaru, Abigail See, Kevin Clark, the Stanford NLP Group, and our anonymous reviewers for their thoughtful comments and suggestions. We gratefully acknowledge support of the DARPA Communicating with Computers (CwC) program under ARO prime contract no. W911NF15-1-0462 and the NSF via grant IIS-1514268. Diyi Yang is supported by a grant from Google.