Towards Debate Automation: a Recurrent Model for Predicting Debate Winners

Peter Potash and **Anna Rumshisky**
Department of Computer Science
University of Massachusetts Lowell
{ppotash,arum}@cs.uml.edu

Abstract

In this paper we introduce a practical first step towards the creation of an automated debate agent: a state-of-the-art recurrent predictive model for predicting debate winners. By having an accurate predictive model, we are able to objectively rate the quality of a statement made at a specific turn in a debate. The model is based on a recurrent neural network architecture with attention, which allows the model to effectively account for the entire debate when making its prediction. Our model achieves state-of-the-art accuracy on a dataset of debate transcripts annotated with audience favorability of the debate teams. Finally, we discuss how future work can leverage our proposed model for the creation of an automated debate agent. We accomplish this by determining the model input that will maximize audience favorability toward a given side of a debate at an arbitrary turn.

1 Introduction

Conversational agents are a well-researched area of natural language generation (Pilato et al., 2007; Bigham et al., 2008; Augello et al., 2008; Agostaro et al., 2005; Bessho et al., 2012). Elsewhere in the field of natural language generation, there is work that seeks to generate persuasive text (Carenini and Moore, 2006; Reiter et al., 2003; Rosenfeld and Kraus, 2016), which is a logical first step towards creating an automated debate agent. One major deficiency of existing work in this area is its assessment of how convincing (or compelling) a piece of text is; the approaches use theory-driven models of persuasion, rather than being empirically motivated. Furthermore, none of these works provide a model that can optimize persuasiveness at an arbitrary point in a conversation.

One of the main reasons for a lack of empirically-driven persuasive generation systems is the absence of labeled data. In order to alleviate this problem (though not directly for the sake of producing an automated debate agent), Zhang et al. (2016) have introduced a dataset of debate transcripts from the "Intelligence Squared" (IQ2)[1] debates. In these debates, two teams are present, arguing either for or against a given topic. For each debate, an audience poll is conducted both prior to and after the debate. Whichever team has the largest gain in audience support between the pre/post debate polls is the winner. This is a natural way to account for the fact that some sides of a debate may be harder to argue than others, and that audience members may be initially biased given a debate topic.

Because of the sequential nature of debating, a Recurrent Neural Network (RNN) is an attractive choice for modeling the problem. Rather than just using the final hidded state for prediction, which likely has lost information from early in the debate, we propose to use an attention mechanism (Bahdanau et al., 2014) that creates a weighted sum over all hidden states, and is subsequently used for the final prediction. We motivate the use of an RNN, as opposed to a temporally flat classifier, for several reasons. First, using an RNN allows us to naturally incorporate predicting audience favorability at each turn while explicitly modeling the turn sequence. Logistic regression, on the other hand, would not allow us to model the sequence explicitly. Secondly, our model allows us to take raw features as input, without having to compute summary statistics necessary for the fea-

[1] http://www.intelligencesquaredus.org/

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2455–2465
Copenhagen, Denmark, September 7–11, 2017.

tures used in the model of Zhang et al. (2016). Finally, since our end goal is debate automation, an RNN is a natural choice for debate turn generation.

There are two major difficulties dealing with the IQ2 dataset: first, since the construction of the dataset is non-trivial, there are only 108 data points, resulting in Zhang et al.'s proposal for leave-one-out (LOO) evaluation. Second, considering the use of an RNN, the sequences are long, with an average length of 246 (and a standard deviation of 67). In order to overcome this, we incorporate signals based on implicit audience feedback during the debate into the model's loss function. Instead of just training the model based on error from the audience's final verdict, propogated through a substantial amount of timesteps, there are intermittent errors propogated backward through the network based on audience reactions, such as applauding or laughing. These internal signals also help regularize the network. In a way, they help generalize the hidden representation of the RNN, allowing it to better contain a distributed representation of the audience's favorability towards a given team.

In our proposed model, the audience's opinion is directly a function of the weighted hidden representations. Since the previous hidden representations are all fixed at a given timestep, and the current hidden representation is directly a function of these previous hidden representations as well as the current input, the audience's current poll depends directly on the timestep's input. Therefore, at a given timestep, our framework allows us to determine the input that would maximize the audience's favorability toward the orating team. This is due to the fact that the inputs are themselves representations of a given team's statement at a particular turn in the debate.

We evaluate our model on the dataset from Zhang et al., posting state-of-the-art accuracy. Our results show that our proposed regularization technique is imperative for the RNN-based model to perform competitively with the models previously proposed by Zhang et al.. The attention mechanism also contributes to the best performing system. Afterward, we show how our model can be used to track audience favorability throughout the debate, as well as the aforementioned input optimization, using it in a case study to instruct a debate team about optimal debate strategy at a given turn.

2 Related Work

Previous work that focuses on conversational language seeks to predict such qualities as disagreements (Allen et al., 2014; Wang and Cardie, 2016), divergence (Niculae and Danescu-Niculescu-Mizil, 2016), and participant stance (Sridhar et al., 2015; Somasundaran and Wiebe, 2010; Thomas et al., 2006; Rosenthal and McKeown, 2015). What is most relevant for our purposes are the methods these models use for dealing with conversational data. Allen et al. (2014) apply discourse parsing (Joty et al., 2013) and fragment quotation graph (Carenini et al., 2007) tools to detect disagreement in online discussion threads. Wang and Cardie (2016) believe that disagreement can be predicted by the presence of substantially long sequences of negative sentiment, motivating them to build a sequential sentiment prediction model using a particular kind of Conditional Random Field (Mao and Lebanon, 2007). Niculae and Danescu-Niculescu-Mizil (2016) use several novel features that capture the flow of ideas in the data, as well as team dynamics. Ultimately, however, all these models apply manually derived, preprocessed features and use a basic classifier, like Random Forest or Logistic Regression. In contrast, an RNN model is able to learn which interactions and overall sequences of rhetoric are important for predictive power.

There is much less work that approaches the problem of predicting persuasiveness of text. This is due primarily to the lack applicable datasets. However, Habernal and Gurevych (2016b) have recently presented a dataset where argument pairs are annotated for argument convincingness, as well as finer-grained annotations related to the effectiveness of arguments (Habernal and Gurevych, 2016a). The authors experimented with feature-based classifiers, as well as various RNN architectures to construct predictive models for the dataset.

The most relevant work for this paper is of course Zhang et al. (2016). The authors use a set of features derived from the notion of idea flow in the debate. More specifically, they follow the method of Monroe et al. (2008) to identify talking points used by the sides present in a debate. The authors then create features based on the coverage of talking points during the debate. Finally, a Logistic Regression model uses these features to predict which team wins the debate. We also note the work of Santos et al. (2016), which also makes

predictions on a dataset derived from the IQ2 debates. In contrast, their work analyses speech signals, as opposed to textual data.

3 Predictive Model

In this section we explain how we apply an RNN to the task of predicting debate winners. We start by addressing the fact that for IQ2 dataset, each timestep involves a text span, as opposed to single tokens, and explaining how we convert this text span into a vector representation for RNN input. Secondly, we explain our RNN model architecture, including our use of an attention mechanism to create a weighted sum over all hidden states, as well as a regularization technique based on implicit audience reaction.

3.1 Representing Debate Turns

Our work follows that of Zhang et al. (2016), and uses talking point-based features, specifically a 'bag of talking points'. Talking points for each debate are identified using a term frequency inverse document frequency (tfidf) metric applied to text tokens. Token counts, whether at a document or corpus level, occur only for the introduction text, as done by Zhang et al. This is based on the belief that the introductory arguments best showcase potential talking points. We take the 10 tokens with highest tfidf scores for each debate, and, across all debates, each token ranking maps to a fixed index in the turn representation. This representation is binary.

Zhang et al.'s results suggest that the interaction of talking points between debate teams can possess strong predictive power. Therefore, we also calculate talking points at a team level within debates. We accomplish this by simply taking term frequency counts for tokens spoken by a given team. Like with the overall debate talking points, we chose the 10 highest ranked talking points from each side and include them in the input representation. Moreover, we believe we can use a simpler talking point metric than that proposed by Monroe et al. (2008) (and used by Zhang et al.) because the recurrent nature of the model will naturally capture the interaction, coverage, and ignorance of the two team's (and overall) talking points.

Aside from talking point-based features, we include the following linguistic features: 1) bag-of-words for tokens that have been used in at least 50 debates; 2) GloVe embeddings of tokens (Pennington et al., 2014). We use max pooling over all the tokens' embeddings to create the embedding features. We also use the following non-linguistic features: 1) whether the turn occurs during the opening, discussion, or conclusion phase of the debate; 2) whether the turn is from the 'for' or 'against' team, as well as moderator or other speakers, such as show host etc; 3) the initial audience poll is provided at each timestep. This is similar in spirit to Cho et al. (2014)'s decoder model that accesses the final encoder hidden state at each timestep.

We acknowledge that it would be possible to model individual turns (sequences of tokens) with a separate RNN. We choose to use hand-engineered features for two reasons: First, the current representation, mainly the talking points and BOW features, are easily interpretable given the goal of providing rhetorical strategy for debaters. Using an RNN for this purpose would require training a decoder in order to interpret the optimal rhetoric at a given turn (see Section 7). Secondly, it follows that having a trainable representation would introduce additional parameters into the model, which is a concern, given the limited amount of data.

3.2 Recurrent Architecture

Our RNN model uses a long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) component. At each timestep, the model receives as input a turn representation defined in Section 3.1. After consuming all turn representations, a simple model without attention woud pass the final hidden state, h_f, through two fully-connected layers (with an intermediate representation h_a to which we apply sigmoid activation), whose weights have subscripts $post$ to identify that this transformation happens after the debate:

$$h_a = \sigma(W^1_{post} h_f + b^1_{post}) \quad (1)$$

$$a = W^2_{post} h_a + b^2_{post} \quad (2)$$

where σ is the sigmoid function. This transformation outputs a vector with three dimensions, which corresponds to the fact that the audience poll has three possibilities: for, against, and undecided.

Since the polling is given as a percentage breakdown, we apply $softmax$ to create a valid probability probability distribution for the audience,

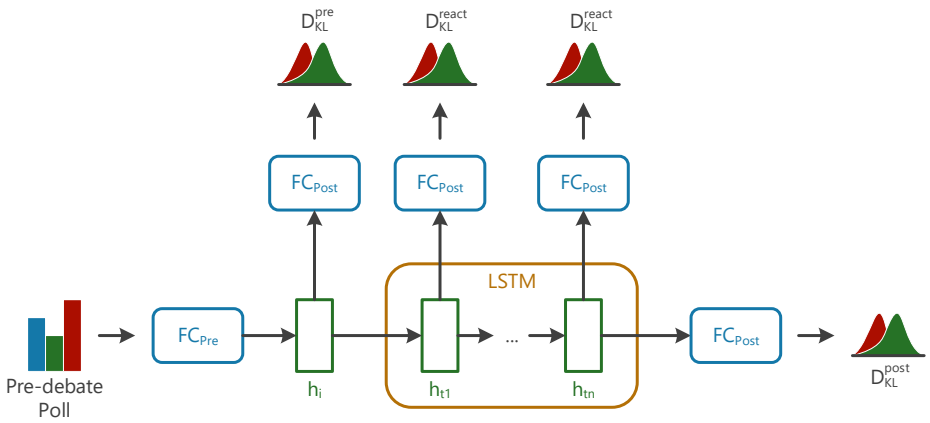

Figure 1: An illustration of our training objective from Equation 9 unrolled over time. FC_{pre} and FC_{post} refer to Equations 1/2 and 8, respectively.

$p(A)$:

$$p(A|\Theta) = \text{softmax}(a) \tag{3}$$

which is for a given set of model paramters, Θ.

We train the model to minimize the Kullback-Leibler (KL) divergence between the target and predicted audience poll percentages. Given a training corpus of debates D with target post-debate audience polls A_i^{target}, the optimization objective is:

$$\arg\min_{\Theta} \sum_{d \in D} D_{KL}(p(A_d^{target})||p(A_d|\Theta)) \tag{4}$$

which simply sums the KL-Divergence of the target and predicted audience poll percentages (probabilities) across all training examples. At test time, the model uses the percentages from $p(A|\Theta)$ to calculate which team increased their support from the audience the most, using the pre-debate audience poll, which is given. For notation purposes, we refer to this KL-divergence for post debate audience polling D_{KL}^{post}. The optimization objective from Equation 4 describes our base model. Shortly, we will describe how we regularize this base model using implicit audience feedback.

3.2.1 Attention Mechanism

The model we have described to this point uses the final hidden state to predict the final audience poll. A concern with this approach is that the final hidden state has a difficult time encoding the activity from the earlier parts of the debate. We propose to rectify this issue by creating a weighted sum over all hidden states, following the the attention mechanism from Bahdanau et al. (2014). Given hidden states from all RNN timesteps, $(h_1, ..., h_f)$, we determine the weight for h_i as follows. First, we compute a raw attention score:

$$r_i = v^T \tanh(W_a h_i + b_a) \tag{5}$$

where v, W_a, b_a are model parameters. h_i's weight is computed from applying softmax to r:

$$\alpha_i = \text{softmax}(r)_i \tag{6}$$

which we use to compute the weighted sum across all hidden states:

$$h_s = \sum_{i=1}^{f} \alpha_i h_i \tag{7}$$

Therefore, the attention version of our model uses h_s in Equation 1 to predict the final audience poll.

3.2.2 Initializing RNN Hidden State

As we have mentioned, audience polls occur both before and after the debate. Thus, we continue the theme of using the RNN hidden state to express audience polling by exploiting the initial audience poll to initialize the RNN hidden state, h_0. The model uses the initial audience poll, a_{pre}, and applies a fully-connected layer with parameters W_{pre} and b_{pre}:

$$h_0 = \tanh(W_{pre} a_{pre} + b_{pre}) \tag{8}$$

We choose $\tanh$ for the activation function because it is the same activation function used by the LSTM cell. The RNN now is initialized with a hidden state that reflects the audience's initial attitude towards a given debate topic.

3.2.3 Regularization via Implicit Audience Feedback

The IQ2 dataset offers two challenges for implementing an RNN-based approach. First, which is a difficulty for any type of supervised model, is the small dataset size. There are a total of 108 data points, which, even with LOO evaluation, leaves only 107 examples for training a model. For neural networks in particular, there is worry that overfitting easily occurs when the amount of model parameters is much greater than the dataset size (Lawrence et al., 1998; Ingrassia and Morlini, 2005). Aside from the dataset size, the sequences of debate turns are long, averaging 246. This means that, on average, our model will run for 246 timesteps, making it difficult to train the network (Bengio et al., 1994) (the structure of the LSTM memory cell was designed to solve this issue, which motivates our use of it in our model).

In order to overcome these difficulties, we propose to regularize our network based on implicit audience feedback that occurs during the debate, and is provided as metadata with the debate transcript. Specifically, provided along side each debate turn, there is a 'non-text' field that indicates if any sounds occurred during the turn such as applause or laughter from the audience. We view the presence of applause or laughter from the audience as a sign of endorsement during that particular turn. Therefore, at that particular timestep, the hidden state should be able to directly predict this occurrence. Considering applause as a sign of endorsement is not controversial, but laughter could be viewed as more ambiguous. However, consider the audience of the debates: the debates air on the Bloomberg network and National Public Radio, suggesting a higher level of maturity of the audience, which is less likely to laugh *at* the participants, rather than at their jokes. For example, here is a turn in the debate 'Men are Finished' wherein laughter occurs: "Wait. What was that phrase you used, surviving off the fumes of sexism? I think we are our finest example there." This is an intentional joke by the speaker, who is part of of the winning team in the debate.

This signal can be integrated in a supervised manner into the loss function by converting the audience reaction at a given timestep into a three-dimensional vector, representing the current, implied audience favorability. We create such a vector at a debate turn if either applause or laughter occurs at that timestep, and the speaker is one of the debate teams. On possibility is to create have a one-hot vector implying the audience favorability at the turn, with the mapping of side to index dictated by the target vector, A_i^{target}, and is set for the corpus. There is a major problem with using a one-hot vector: the probability distributions learned by the model will become too skewed, since the ultimate goal is to better generalize the prediction of debate polls, rarely are the polls so unbalanced toward one side. Moreover, the one-hot vector will only ever have mass in the indices for the 'for' and 'against' teams, and neglecting the 'undecided' index, which is an important sector in the polling. Therefore, we create a soft vector as follows: a random number, n, is chosen in the interval $(\frac{1}{3}, 1)$. The index corresponding to the speaking team at timestep i has value n. The remaining two indices have value $\frac{1-n}{2}$. This vector is notated $A_{i_t}^{target}$, specifying that the reaction occurred at timestep t for debate i. On average, such reactions occur 21 times during a debate, with a standard deviation of 10. Consequently, this approach adds 2,268 more supervised signals to the dataset.

As we did with the post-debate poll, we can compute a lost based on the kl-divergence between $A_{i_t}^{target}$ and the prediction probability at timestep t, which is a function of h_t using the same transformations described in Equations 1, 2, and 3, but replacing h_f with h_t. The attention model can been used as well. In this case, we compute h_{s_t} by slicing r (from equation 5) to only include indices up to t. We denote the KL-divergence between target and prediction distributions across all timesteps of a training example is D_{KL}^{react}, since these signals are based on audience reaction.

The same strategy can be applied to h_i using the pre-debate polls. Although this signal does not propagate through the RNN, it can still train the weights of the fully-connected layers used in our model. We refer to this KL-divergence as D_{KL}^{pre}, since it uses the pre-debate poll. Bringing together these separate error signals, we arrive at the training objective of our full model:

$$\arg\min_{\Theta} \sum_{d \in D} D_{KL}^{pre} + D_{KL}^{react} + D_{KL}^{post} \qquad (9)$$

where Θ is the model parameters used to produce the prediction probabilities. Figure 1 provides an illustration of our training objective, unrolled over time.

With this new optimization objective, each example now trains our model based on (on average) 23 supervised signals. As a result, each training example allows the model to become more generalizable, particularly because the hidden states are now better-tuned to encode audience favorability. This methodology allows the model to better leverage the small dataset size. Moreover, the intermittent error signals from audience reaction, D_{KL}^{react}, combined with the pre-debate error signal, D_{KL}^{pre}, help assuage the difficulties of training our model based on a final error signal propagated for many timesteps. We would like to reiterate that this regularization technique is only used to *train* the model, and not used for prediction, and therefore will not be an issue when making predictions for new debates, nor will it create an unrealistic circumstance for using the model for creating a debate agent.

4 Experimental Design

Our experiments are conducted on the IQ2 dataset (Zhang et al., 2016). We use LOO evaluation, resulting in a training set of 107 examples. The evaluation metric is simply prediction accuracy for debate winners. The winning team is based on audience polling. Polls are conducted before and after the debate, and audience members can vote as being either for or against a given debate topic, as well as being undecided. The team that has the highest increase in audience support from the pre to post debate poll is the winning team. The model trains for 100 epochs. Once training is complete, we test on the held-out data point. As Zhang et al. note, there are three debates in the dataset that have a tie between the debate teams. Following their procedure, we do not test on these data points. However, we still include these examples in the training sets, because our training objective is to predict polls, not debate winners. The final test accuracy is averaged across the remaining 105 LOO runs. Furthermore, we note that the dataset is effectively balanced, as there are 53 and 52 examples with the two possible labels.

We implement all our models in TensorFlow (Abadi et al., 2016). We use the LSTM cell equipped with peephole connections (Gers et al., 2002). This architecture allows the gates to see the current cell state, along with the hidden state. We believe that because of the long sequences present in the dataset, it is important to have all the gates take into account the cell state when producing a hidden layer. This adds a stronger notion of memory to the model. While we expect the hidden state to represent audience favorability, we believe the cell state can capture the further latent notion of debate strategy, observable through the interaction of talking points between the debate teams. The models have cell and hidden size of 128, and the intermediate layer from Equation 2 has size 16. Lastly, we use a batch size of 8.

Model	Accuracy
LR BOW	0.50
LR React	0.60
LR Flow	0.63
LR Flow*	0.65
LSTM	0.55
LSTM + Att	0.57
LSTM + Reg	0.64
LSTM + Att, Reg	**0.71**
LSTM + Att, Drpt	0.60

Table 1: The results of LOO evaluation on the IQ2 dataset. See the beginning of Section 5 for an explanation of the models.

5 Results

The results of our experiment are presented in Table 1. Att means the model has the attention mechanism from Section 3.2.1; Reg means the model uses the optimization objective from Equation 9 (all other models use the optimization objective from Equation 4); Drpt means the model uses dropout (a popular regularization technique for neural networks (Srivastava et al., 2014)) of 0.5. We compare our results against the best models from Zhang et al. (2016). Each model uses a Logistic Regression (LR) classifier, and distinguishes itself by the features it uses. The main features developed by the authors relate to the interaction (flow) of talking points between the debate teams. There are two types of models that use the flow features: LR Flow and LR Flow*. Whereas the former uses all developed flow features, the latter uses feature selection to keep the most powerful flow features. LR React uses features based on audience reaction metadata, and LR BOW uses bag-of-words features.

The results show that the LSTM attention model regularized by audience reaction achieves the

highest accuracy. Furthermore, the results highlight the importance of this regularization technique, since the simple LSTM model records the second lowest accuracy of any of the models presented. This leads us to believe that the regular LSTM model falls victim to the lack of training data, preventing the larger amount of model parameters (compared to a logistic regression model) from generalizing. The results also show that the attention model has higher performance than the regular LSTM model, and the difference in performance is heightened when the regularization technique is applied. We believe this is because the attention mechanism adds additional parameters to the model, so it seems reasonable that adding additional training signals helps the model to generalize better. Lastly, our proposed regularization technique is far superior for generalization than the popular dropout method. We believe the strong performance of the proposed regularization technique is because it causes the LSTM's hidden states to better generalize the notion of encoding audience favorability. Furthermore, our model's goal is to predict *distributions*, as opposed to labels. Whereas dropout can be effective at aiding in collapsing representations of the same class into neighboring points of a latent space, our model needs to be able to predict polls that it may have not encountered in training. Our regularization technique aids in this as well by providing more training data, more polls.

6 Tracking Audience Favorability

One of the advantages of mapping a recurrent model's hidden states to audience favorability is that we can produce a favorability poll at any turn (timestep) during the debate. In contrast, a temporally flat model, such as the logistic regression models from Zhang et al., produce a prediction of audience favorability based on features extracted from the entire debate. Using our mapping of hidden states to audience favorability, we can determine, at each turn, the current audience favorability, and track it throughout the entire debate. Figure 2 shows this applied to the "men are finished" debate, wherein the lines on the graph, cut vertically, represent predicted audience polls at a given debate turn. This debate saw the greatest increase in audience support from the pre to post debate poll: the 'for' increased their favorability by 46% (46 points). The three lines correspond to the three possible positions an audience member can take regarding the debate topic. This visualization can be particularly useful for rhetorical analysis of debate performance, because the resulting graph allows us to see inflection points in audience favorability. These inflection points suggest that a debate team used very effective (or ineffective) rhetoric at that particular turn.

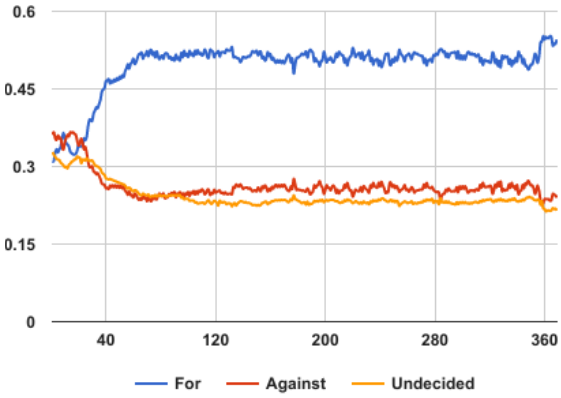

Figure 2: A visualization of audience favorability for the debate "men are finished". At each turn in the debate, our model predicts the audience favorability. The y-axis shows the percentage of the audience that supports a given side, and the x-axis show the turn number for a given poll. Even though these are purely predictions from the model, it is able to show the rise in audience favorability for the 'for' team, as well as the decline in favorability for the 'against' team. From the graph, we can see that the 'for' team had a large spike in audience support roughly between turns 20 and 40, which corresponds to the beginning of the debate's discussion section.

7 Optimizing Input for Audience Favorability

Aside from achieving a new state-of-the-art result on the IQ2 debate corpus, the main appeal of the model we have introduced is that it creates a mapping between the hidden states and audience favorability of the debate teams. This mapping is given in Equations 1 and 2, where a weighted sum over all over all hidden states (the actual notation in these equation apply the fully-connected transformation to a final hidden state, h_f, unlike the attention model which uses h_s from Equation 7) is transformed into a real-valued 3-dimensional vector a. The values of the vector indicate 'raw' fa-

vorability, which is realized as a probability distribution (or alternatively, a poll of the audience) after applying the softmax activation. Furthermore, given fixed model parameters Θ, the current hidden state is a function of the previous hidden states, previous cell state (if, like our model, an LSTM cell is used), as well as the current input. At a given timestep, the previous hidden and cell states are known. Therefore, a is directly a function of the current input x at a given timestep. This notion of optimizing input for a target 'class' is akin to the work of Simonyan et al. (2013), who use a trained convolutional neural network to find the optimal input image for a desired object class.

7.1 Input Optimization Objective

Similar to our approach in Section 3.2.3 to encode implicit audience feedback, we can construct a three-dimensional one-hot vector with the index switched on that corresponds to the debate team whose favorability we seek to optimize. We will call this vector A^{fav}. Given input x_i at timestep i, we seek to to optimize the probability of A^{fav} given x_i:

$$\arg\max_{x_i} p(A^{fav}|x_i, h_1, ..., h_{i-1}, c_{i-1}; \Theta) \quad (10)$$

Where $i \in (1, ..., T)$ and T is the maximum number of timesteps (turns) for a debate. In practice, we achieve this optimization by minimizing the cross-entropy between the the target one-hot vector and the output of applying the softmax function to a, as in Equation 3.

7.2 Applying Optimized Input for Persuasive Strategy

In the debate 'men are finished', the 'for' team won the debate, increasing their favorability by an astonishing 46% (conversely, the 'against' team saw a 25% *decrease* in favorability). According to our model's sequential predictions (and visible in Figure 2), a major turning point occurred at turn 27. Quantitatively, we can examine the turn-by-turn change in audience favorability: from this we see that one of the largest increases in audience favorability occurred at turn 27. It is not a surprise to find out that the team that spoke during turn 27 was the 'for' team. When asked by the moderator if there can be equality between the sexes without deeming men as being finished, the 'for' team said the following (the text is annotated for the presence of talking points, marked by a subscript that specifies whose talking point it is: A (against), F (for), or G, a general talking point based on overall token frequency (see Section 3.1)):

> It is possible, but it just doesn't work that way. I mean, if we can all agree that there was male dominance for a long time and that male dominance is over, then I think we agree that $\underline{\text{men}}_{G,A}$ are finished. So the resolution is about male dominance which we've taken for granted for so many tens of thousands of years. And so, even if you have parity, you have the end of male dominance. I mean, if you have $\underline{\text{women}}_{F}$ rising and catching up to $\underline{\text{men}}_{G,A}$, then you no longer have male dominance. And so that's what I meant when I, early on, tried to define the resolution as $\underline{\text{men}}_{G,A}$ are finished, the era of male dominance, it's finished, which we've taken for granted for all this time.

Note that the term 'women' is only a talking point for the 'for' team. In their response the 'against' team says:

> They are not finished. That's absurd. You agreed to it in your opening that you didn't want to say $\underline{\text{men}}_{G,A}$ are finished. You thought there might be inklings of a suggestion that it may be happening. But what you're defending now is that $\underline{\text{men}}_{G,A}$ are finished. I'm saying it's absurd. I'm saying that some $\underline{\text{men}}_{G,A}$ are in trouble. But rather than declare their extinction, we should be doing what we can to help them.

To determine our model's strategy immediately after the 27th turn, we apply the previous hidden and cell states to the optimization objective in Equation 10, taking the place of $h_1, ..., hi-1$ and c_{i-1}, respectively. We fit the training objective to the current states, as well as the weights of the previously trained predictive model, and examine the resulting optimized input vector. We train the optimized input model for 15,000 epochs, which goes very fast because there is a 'single' training data point, and the model is not recurrent. As we can see in the actual 'against' team's response, the only talking point brought up is 'men', which can

hardly be viewed as an enlightening notion in the context of the debate. Alternatively, the highest rated talking point from the optimized input is in fact the exact talking point brought up by the 'for' team: 'women'. This suggestion by our model is in line with the hypothesis of Zhang et al. (2016), that winning teams are effective in adopting their opponent's talking points. In terms of bag-of-word features: the optimized input ranks the following tokens as the ten highest (in descending order of score, and note the tokens have been stemmed): 'sound', 'present', 'recent', 'line', 'decid', 'veri', 'spent', 'save', 'moder', and 'found'. Most of these tokens remain somewhat vague with respect to their relevance to the debate. The token 'recent' seems relevant, given that the debate topic has an inherent temporal nature. 'Save' is relevant in that some of the debate discussion approaches the question of whether men need saving. In the top 20 tokens we also find 'done', 'compar', 'grow', and 'without', which are all relevant: 'done' is synonymous with 'finished', 'compar' given that the debate is often comparing men to women, 'grow' could refer to the growth of women in society, and 'without' is a token specifically in the question the moderator asked prior to turn 27 (equality between the sexes without deeming men as being finished).

8 Conclusion

We have presented an RNN model for predicting debate winners, with the specific goal of predicting the final (or intermediate) audience poll. The model takes at each timestep a representation of a given debate turn. The model uses an attention mechanism that creates a weighted sum over all hidden states. In order to achieve state-of-the-art results on a corpus of debate transcripts (Zhang et al., 2016), we regularize the RNN model by propagating errors based on implicit audience reaction. Our results show that this regularization technique is critical for obtaining a state-of-the-art result. We have also shown the practical application of our model in two scenarios. First, the model can be used to make a prediction of audience polling at every debate turn. This allows for an analysis of the key turning points during the debate, based on inflections in audience favorability. Second, the model can be used to determine the optimal input at a given debate turn. Knowledge of this input can inform debaters as to the best current persuasive strategy. Future work can leverage optimal inputs to create a language model that can become an automated debate agent. However, since the input is partially based on the knowledge of talking points, there is a potential for an information retrieval-based task to provide the talking points for the debate agent (if one desires a fully-automated system than can work without the presence of introductory remarks, from which talking points are currently extracted). Finally, future work can also examine the trained model itself in further detail, seeking to understand the debate strategy.

Acknowledgments

This work was supported in part by the U.S. Army Research Office under Grant No. W911NF-16-1-0174. We would like to thank Alexey Romanov for his help with the figures.

References

Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Francesco Agostaro, Agnese Augello, Giovanni Pilato, Giorgio Vassallo, and Salvatore Gaglio. 2005. A conversational agent based on a conceptual interpretation of a data driven semantic space. In *Congress of the Italian Association for Artificial Intelligence*, pages 381–392. Springer.

Kelsey Allen, Giuseppe Carenini, and Raymond T Ng. 2014. Detecting disagreement in conversations using pseudo-monologic rhetorical structure. In *EMNLP*, pages 1169–1180.

Agnese Augello, Gaetano Saccone, Salvatore Gaglio, and Giovanni Pilato. 2008. Humorist bot: Bringing computational humour in a chat-bot system. In *Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on*, pages 703–708. IEEE.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Fumihiro Bessho, Tatsuya Harada, and Yasuo Kuniyoshi. 2012. Dialog system using real-time crowdsourcing and twitter large-scale corpus. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 227–231. Association for Computational Linguistics.

Jeffrey P Bigham, Maxwell B Aller, Jeremy T Brudvik, Jessica O Leung, Lindsay A Yazzolino, and Richard E Ladner. 2008. Inspiring blind high school students to pursue computer science with instant messaging chatbots. In *ACM SIGCSE Bulletin*, volume 40, pages 449–453. ACM.

Giuseppe Carenini and Johanna D Moore. 2006. Generating and evaluating evaluative arguments. *Artificial Intelligence*, 170(11):925–952.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. 2007. Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, pages 91–100. ACM.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. 2002. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143.

Ivan Habernal and Iryna Gurevych. 2016a. What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In *EMNLP*, pages 1214–1223.

Ivan Habernal and Iryna Gurevych. 2016b. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Salvatore Ingrassia and Isabella Morlini. 2005. Neural network modeling for small datasets. *Technometrics*, 47(3):297–311.

Shafiq R Joty, Giuseppe Carenini, Raymond T Ng, and Yashar Mehdad. 2013. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In *ACL (1)*, pages 486–496.

Steve Lawrence, C Lee Giles, and Ah Chung Tsoi. 1998. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report.

Yi Mao and Guy Lebanon. 2007. Isotonic conditional random fields and local sentiment flow. *Advances in neural information processing systems*, 19:961.

Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. 2008. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

Vlad Niculae and Cristian Danescu-Niculescu-Mizil. 2016. Conversational markers of constructive discussions. *arXiv preprint arXiv:1604.07407*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Giovanni Pilato, Agnese Augello, Giorgio Vassallo, and Salvatore Gaglio. 2007. Sub-symbolic semantic layer in cyc for intuitive chat-bots. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 121–128. IEEE.

Ehud Reiter, Roma Robertson, and Liesl M Osman. 2003. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1-2):41–58.

Ariel Rosenfeld and Sarit Kraus. 2016. Strategical argumentative agent for human persuasion. In *ECAI 2016: 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands-Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, volume 285, page 320. IOS Press.

Sara Rosenthal and Kathleen McKeown. 2015. I couldnt agree more: The role of conversational structure in agreement and disagreement detection in online discussions. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 168.

Pedro Bispo Santos, Lisa Beinborn, and Iryna Gurevych. 2016. A domain-agnostic approach for opinion prediction on speech. *PEOPLES 2016*, page 163.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124. Association for Computational Linguistics.

Dhanya Sridhar, James R Foulds, Bert Huang, Lise Getoor, and Marilyn A Walker. 2015. Joint models of disagreement and stance in online debate. In *ACL (1)*, pages 116–125.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 327–335. Association for Computational Linguistics.

Lu Wang and Claire Cardie. 2016. A piece of my mind: A sentiment analysis approach for online dispute detection. *arXiv preprint arXiv:1606.05704*.

Justine Zhang, Ravi Kumar, Sujith Ravi, and Cristian Danescu-Niculescu-Mizil. 2016. Conversational flow in oxford-style debates. *arXiv preprint arXiv:1604.03114*.