A Factored Neural Network Model for Characterizing Online Discussions in Vector Space

Hao Cheng Hao Fang Mari Ostendorf
University of Washington
{chenghao,hfang,ostendorf}@uw.edu

Abstract

We develop a novel factored neural model that learns comment embeddings in an unsupervised way leveraging the structure of distributional context in online discussion forums. The model links different context with related language factors in the embedding space, providing a way to interpret the factored embeddings. Evaluated on a community endorsement prediction task using a large collection of topic-varying Reddit discussions, the factored embeddings consistently achieve improvement over other text representations. Qualitative analysis shows that the model captures community style and topic, as well as response trigger patterns.

1 Introduction

Massive user-generated content on social media has drawn interests in predicting community reactions in the form of virality (Guerini et al., 2011), popularity (Suh et al., 2010; Hong et al., 2011; Lakkaraju et al., 2013; Tan et al., 2014), community endorsement (Jaech et al., 2015; Fang et al., 2016), persuasive impact (Althoff et al., 2014; Tan et al., 2016; Wei et al., 2016), etc. Many of these studies have analyzed content-agnostic factors such as submission timing and author social status, as well as language factors that underlie the textual content, e.g., the topic and idiosyncrasies of the community. In particular, there is an increasing amount of work on online discussion forums such as Reddit that exploits the conversational and community-centric nature of the user-generated content (Lakkaraju et al., 2013; Althoff et al., 2014; Jaech et al., 2015; Tan et al., 2016; Wei et al., 2016; He et al., 2016a; Fang et al., 2016), which contrasts with Twitter where the author's social status seems to play a larger role in popularity. This paper focuses on Reddit, using the karma score[1] as a readily available measure of community endorsement.

Some of the prior work on Reddit investigates specific linguistic phenomena (e.g. politeness, topic relevance, community style matching) using feature engineering to understand their role in predicting community reactions (Althoff et al., 2014; Jaech et al., 2015). In contrast, this paper explores methods for unsupervised text embedding learning using a model structured so as to provide some interpretability of the results when used in comment endorsement prediction. The model aims to characterize the interdependence of comment on its global context and subsequent responses that is characteristic of multi-party discussions. Specifically, we propose a factored neural model with separate mechanisms for representing global context, comment content and response generation. By factoring the model, we hope unsupervised learning will pick up different components of interactive language in the resulting embeddings, which will improve prediction of community reactions.

Distributed representations of text, or text embeddings, have achieved great success in many language processing applications, using both supervised and unsupervised methods. Unsupervised learning, in particular, has been successful at different levels, including words (Mikolov et al., 2013b), sentences (Kiros et al., 2015), and documents (Deerwester et al., 1990; Le and Mikolov, 2014). Studies have also shown that the learned embedding captures both syntactic and semantic functions of words (Mikolov et al., 2013a; Pennington et al., 2014; Levy and Goldberg, 2014; Faruqui et al., 2015a). At the same time, em-

[1] karma = #up-votes - #down-votes. See https://goo.gl/TnXgCr.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2296–2306
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

beddings are often viewed as uninterpretable – it is difficult to align embedding dimensions to existing semantic or syntactic classes. This concern has triggered attempts in developing more interpretable embedding models (Faruqui et al., 2015b), which is also a goal of our work. We leverage the fact that the structure of the distributional context impacts what is learned in an unsupervised way and include multiple objectives for separating different types of context.

Here, we are interested in linking two types of context with corresponding language factors learned in the embedding space that may impact comment reception. First, conformity to the topic and the language use of the community tends to make the content better accepted (Lakkaraju et al., 2013; Tan et al., 2014; Tran and Ostendorf, 2016). Those global *modes* typically influence the author's generation of local *content*. Second, characteristics of a comment can influence the *responses* it triggers. Clearly, questions and statements will elicit different responses, and comments directed at a particular discussion participant may prompt that individual to respond. Of more interest here are aspects of comments that might elicit minimal response or responses with different sentiments, which are relevant for eventual endorsement.

The primary contribution of this work is the development of a factored neural model to jointly learn these aspects of multi-party discussions from a large collection of Reddit comments in an unsupervised fashion. Extending the recent neural attention model (Bahdanau et al., 2015), the proposed model can interpret the learned latent global modes as community-related topic and style. A comment-response generation model component captures aspects of the comment that are response triggers. The multi-factored comment embedding is evaluated on the task of predicting the comment endorsement for three online communities different in topic trends and writing style. The representation of textual information using our approach consistently outperforms multiple document embedding baselines, and analyses of the global modes and response trigger subvectors show that the model learns common communication strategies in discussion forums.

2 Model Description

To characterize different aspects of language use in a comment, the proposed model factorizes a comment embedding into two sub-vectors, *i.e.* a *local mode* vector and a *content* vector. The local mode vector, computed as a mixture of global mode vectors, exploits the global context of a comment. In Reddit discussions that we use, the global mode represents the topic and language idiosyncracies (style) of a particular subreddit. More specific information communicated in the comment is captured in the content vector. The generation process of a comment is modeled through a recurrent neural network (RNN) language model (LM) conditioned on local mode and content vectors, while the global mode vectors are jointly learned during the training. Moreover, a residual learning architecture (He et al., 2016b) is used to extend the RNN LM for separating the information flow of the mode and the content vectors.

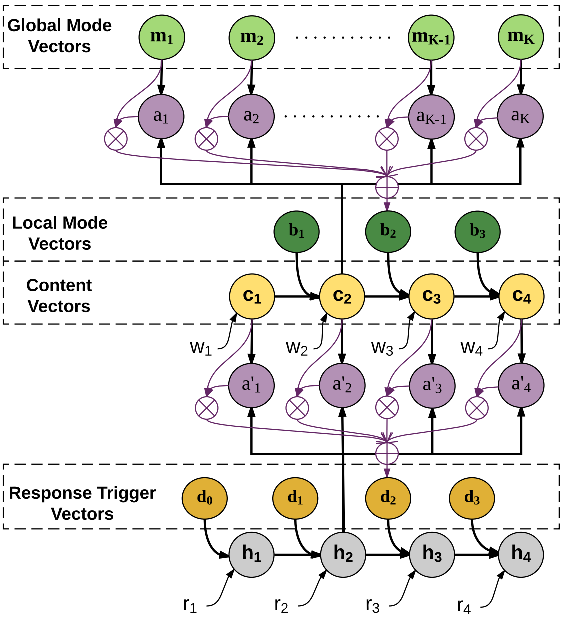


Figure 1: The structure of the full model omitting output layers, illustrating the computation of attention weights for $\mathbf{b}_2$ and $\mathbf{d}_3$ in a comment $w_{1:4}$ with its response $r_{1:4}$. Purple circles a_k and a'_j represent scalars computed in (1) and (6), respectively. $\otimes$ and $\oplus$ are scaling and element-wise addition operators, respectively. Black arrowed lines are connections carrying weight matrices.

In addition to the global context, the full model further exploits direct responses to the comment in order to learn better comment embeddings. This is achieved by modeling the generation of comment responses through another RNN LM conditioned on *response trigger* vectors. The response trigger vectors are computed as mixtures of content vectors, with the idea that they will characterize aspects of the comment that incent others to respond, whether that be information or framing.

The full model is illustrated in Fig. 1. While the end goal is a joint framework, the model is described in the following two sub-sections in terms of two components: i) mode vectors for capturing global context, and ii) response trigger vectors for exploiting comment responses.

2.1 Mode Vectors

Using an RNN LM shown in the upper part of Fig. 1, we model the generation process of a word sequence by predicting the next word conditioned on the global context as well as the local content. The global context is encoded in the local mode vector, computed as a mixture of global mode vectors with mixture weights inferred based on content vectors. The local mode vector indicates where the comment fits in terms of what people in this subreddit generally say. It changes dynamically with the content vector as the comment generation progresses, considering the possibility of topic shifts or different broad categories of discussion participants.

Suppose there is a set of K latent global modes with distributed representations $\mathbf{m}_{1:K} \in \mathbb{R}^n$. For the t-th word w_t in a sequence, a local mode vector $\mathbf{b}_t \in \mathbb{R}^n$ is computed as

$$\mathbf{b}_t = \textstyle\sum_{k=1}^{K} a(\mathbf{c}_t, \mathbf{m}_k) \otimes \mathbf{m}_k,$$

where $\mathbf{c}_t \in \mathbb{R}^n$ is the content vector for the current partial sequence $w_{1:t}$, $\otimes$ multiplies a vector by a scalar, and the function $a(\mathbf{c}_t, \mathbf{m}_k)$ outputs a scalar *association probability* for the current content vector $\mathbf{c}_t$ and a mode vector $\mathbf{m}_k$. The association function $a(\mathbf{c}, \mathbf{m}_k)$ is defined as

$$a(\mathbf{c}, \mathbf{m}_k) = \frac{\exp(\mathbf{v}^T \tanh(\mathbf{U}\,[\mathbf{c}; \mathbf{m}_k]))}{\sum_{i=1}^{K} \exp(\mathbf{v}^T \tanh(\mathbf{U}\,[\mathbf{c}; \mathbf{m}_i]))}, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{n \times 2n}$ and $\mathbf{v} \in \mathbb{R}^n$ are parameters characterizing the similarity between $\mathbf{m}_k$ and $\mathbf{c}$.

The computation of the association probability is the well-known attention mechanism (Bahdanau et al., 2015). However, unlike the original attention RNN model where the attended vector is concatenated with the input vector to augment the input to the recurrent layer, we adopt a residual learning approach (He et al., 2016b) to learn content vectors. For the t-th word w_t in a sequence, the content vector $\mathbf{c}_t$ under the original attention RNN model is computed as

$$\mathbf{c}_t = f(\mathbf{W}\mathbf{x}_t + \mathbf{G}\mathbf{b}_{t-1}, \mathbf{c}_{t-1}), \tag{2}$$

where $\mathbf{x}_t \in \mathbb{R}^d$ is the word embedding for w_t, $\mathbf{b}_{t-1} \in \mathbb{R}^n$ and $\mathbf{c}_{t-1} \in \mathbb{R}^n$ are previous local mode and content vectors, respectively, $\mathbf{W} \in \mathbb{R}^{n \times d}$ and $\mathbf{G} \in \mathbb{R}^{n \times n}$ are weight matrices transforming the input to the recurrent layer, and $f(\cdot, \cdot)$ is the recurrent layer activation function. To address the vanishing gradient issue in RNNs, we use the gated recurrent unit (Cho et al., 2014) for the RNN layer, *i.e.*

$$f(\mathbf{p}, \mathbf{q}) = (\mathbf{1} - \mathbf{u}) \odot \tanh(\mathbf{p} + \mathbf{R}[\mathbf{r} \odot \mathbf{q}]) + \mathbf{u} \odot \mathbf{q},$$

where $\odot$ is the element-wise multiplication, $\mathbf{R}$ is the recurrent weight matrix, and $\mathbf{u}$ and $\mathbf{r}$ are the update and reset gates, respectively. In this paper, we compute the content vector $\mathbf{c}_t$ as follows:

$$\mathbf{c}_t = f(\mathbf{W}\mathbf{x}_t, \mathbf{G}\mathbf{b}_{t-1} + \mathbf{c}_{t-1}). \tag{3}$$

Comparing (2) and (3), it can be seen that we first aggregate the local mode vector $\mathbf{b}_{t-1}$ and the content vector $\mathbf{c}_{t-1}$ and treat the resulting vector $\mathbf{G}\mathbf{b}_{t-1} + \mathbf{c}_{t-1}$ as the memory of the recurrent layer. The resulting hidden state vectors from the recurrent layer are content vectors $\mathbf{c}_t$'s. The use of residual learning is motivated by the following considerations. The local mode vector $\mathbf{b}_{t-1}$ can be seen as a non-linear transformation of $\mathbf{c}_{t-1}$ into a global mode space parameterized by $\mathbf{m}_{1:K}$. If the global information carried in $\mathbf{b}_{t-1}$ is residual for generating the following word in the comment, the model only needs to exploit the information in local content $\mathbf{c}_{t-1}$ and learns to zero out the local mode vector $\mathbf{b}_{t-1}$, *i.e.* $\mathbf{G} = 0$. He et al. (2016b) show that the residual learning usually leads to a more well-conditioned model which promises better generalization ability.

Finally, the RNN LM estimates the probability of the $(t+1)$-th word w_{t+1} based on the current local mode vector $\mathbf{b}_t$ and content vector $\mathbf{c}_t$, *i.e.*

$$\Pr(w_{t+1}|w_{1:t}) = \mathrm{softmax}(\mathbf{Q}(\mathbf{G}\mathbf{b}_t + \mathbf{c}_t)), \tag{4}$$

where $\mathbf{Q} \in \mathbb{R}^{V \times n}$ is the weight matrix, and V is the vocabulary size. Note that the model jointly learns all parameters in the RNN together with the mode vectors $\mathbf{m}_{1:K}$. This differs our model from the context-dependent RNN LM (Mikolov and Zweig, 2012), which is conditioned on a context vector inferred from a pre-trained topic model.

2.2 Response Trigger Vectors

Another important aspect of comments in online discussions is how other participants react to the content. In order to exploit those characteristics, we use comment-reply pairs in online discussions and build this component upon the encoder-decoder framework with the attention mechanism (Bahdanau et al., 2015), which is illustrated in the lower part of Fig. 1. The decoder is essentially another RNN LM conditioned on response trigger vectors aiming at distilling relevant parts of the comment which other people are responding to.

Let r_j denote the j-th word in a reply to a comment $w_1, \cdots, w_T$. The decoder RNN LM computes a hidden vector $\mathbf{h}_j \in \mathbb{R}^n$ for r_j as follows,

$$\mathbf{h}_j = f(\mathbf{W}^\dagger \mathbf{x}_j + \mathbf{G}^\dagger \mathbf{d}_{j-1}, \mathbf{h}_{j-1}), \tag{5}$$

where $\mathbf{W}^\dagger \in \mathbb{R}^{n \times d}$ and $\mathbf{G}^\dagger \in \mathbb{R}^{n \times n}$ are weight matrices, $\mathbf{x}_j$ is r_j's word embeddings from a shared embedding dictionary as used by the encoder RNN LM in Subsection 2.1, and $\mathbf{d}_{j-1} \in \mathbb{R}^n$ and $\mathbf{h}_{j-1} \in \mathbb{R}^n$ are the response trigger vector and hidden vector at the previous time step, respectively. The initial hidden vector $\mathbf{h}_0$ is set to be the last content vector $\mathbf{c}_T$. With the comment's content vectors $\mathbf{c}_1, \cdots, \mathbf{c}_T$ obtained from the encoder RNN LM in Subsection 2.1, a response trigger vector $\mathbf{d}_j$ is computed as the mixture:

$$\mathbf{d}_j = \textstyle\sum_{t=1}^{T} a'(\mathbf{h}_j, \mathbf{c}_t) \cdot \mathbf{c}_t, \tag{6}$$

where $a'(\mathbf{h}_j, \mathbf{c}_t)$ is a similar function to $a(\mathbf{c}_t, \mathbf{m}_k)$ defined in (1) with different parameters. Similar to the encoder RNN LM, the decoder RNN LM estimates the probability of the $(j+1)$-th word r_{j+1} in the reply based on the hidden vector $\mathbf{h}_j$ and the response trigger vector $\mathbf{d}_j$, *i.e.*

$$\Pr(r_{j+1}|r_{1:j}) = \mathrm{softmax}(\mathbf{Q}^\dagger\,[\mathbf{h}_j; \mathbf{d}_j]),$$

where $\mathbf{Q}^\dagger \in \mathbb{R}^{V \times 2n}$ is the weight matrix.

Note the decoder RNN only aims at providing additional supervision signals in training the encoder RNN through a response generation task. At test time, we do not use the responses therefore do not need to run the decoder RNN LM.

3 Model Learning

The full model is trained by maximizing the log-likelihood of the data, *i.e.*

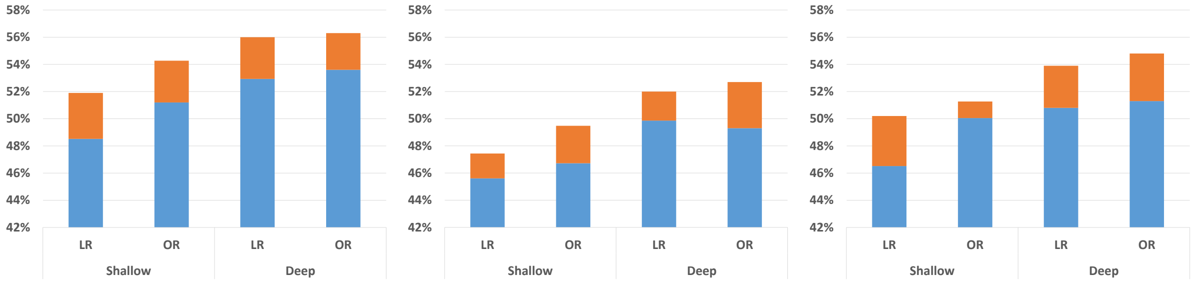
$$\textstyle\sum_i \log \Pr(w^{(i)}_{1:T^{(i)}}) + \alpha \log \Pr(r^{(i)}_{1:J^{(i)}} | w^{(i)}_{1:T^{(i)}}),$$

where the two terms correspond to the log-likelihood of the encoder RNN LM and the decoder RNN LM, respectively, and α is the hyper parameter which weights the importance of the second term. In our experiments, we let $\alpha = 0.1$. During the training, each comment-reply pair is used as a training sample. Considering that comments may receive a huge number of replies, we keep up to 5 replies for each comment. Due to memory limitations associated with the RNN, we use only the first 50 words of comments and the first 20 words of replies. If a comment has no reply, a special token is used. All weights are randomly initialized according to $\mathcal{N}(0, 0.01)$. The model is optimized using Adam (Kingma and Ba, 2015) with an initial learning rate 0.01. Once the validation log-likelihood decreases for the first time, we halve the learning rate at each epoch. The training process is terminated when the validation log-likelihood decreases for the second time. In our experiments, we learn word embeddings of dimension $d = 256$ from scratch. The number of modes K is set to 16. A single-layer RNN is used, with the dimension n of hidden layers set to 64.

4 Data and Task

In this paper, we work with Reddit discussion threads, taking advantage of their conversational and community-centric nature as well as the available karma scores. Each thread starts from a post and grows with comments to the post or other comments within the thread, presented as a tree structure. Posts and comments can be voted up or down by readers depending on whether they agree or disagree with the opinion, find it amusing vs. offensive, etc. A *karma* score is computed as the difference between up-votes and down-votes, which has been used as a proxy of community endorsement for a Reddit comment. Three popular subreddits with different topics and styles are studied[2] `AskWomen` (814K comments), `AskMen` (1,057K comments), and `Politics` (2,180K comments). For each subreddit, we randomly split comments by threads into training, validation, and test data, with a 3:1:1 ratio. The vocabulary of each subreddit is built on the training set. After removing singletons, the vocabulary sizes are 45K, 52K, and 60K for `AskWomen`, `AskMen`, and `Politics`, respectively.

[2] Comment IDs and labels used in this paper is at `https://github.com/hao-cheng/factored_neural`.



(a) `AskWomen` (b) `AskMen` (c) `Politics`

Figure 2: Averaged F1 scores of different classifiers. Blue bars show the performance using no comment embeddings. Orange bars show the absolute improvement by using factored comment embeddings.

Task: Considering the heavy-tailed Zipfian distribution of karma scores, regression with a mean squared error objective may not be informative because low-karma comments dominate the overall objective. Following (Fang et al., 2016), we quantize comment karma scores into 8 discrete levels and design a task consisting of 7 *binary* classification subtasks which individually predict whether a comment's karma is *at least* level-l for each level $l = 1, \cdots, 7$. This task is sensitive to the order of quantized karma scores, e.g., for the level-6 subtask, predicting a comment as level-5 or level-7 would lead to different evaluation results such as recall, which is not the case for a standard multi-class classification task. Additionally, compared to a standard multi-class classification task, these subtasks alleviate the unbalanced data issue, although higher levels are still more skewed.

Evaluation metric: For each level-l binary classification subtask, we compute the F1 score by treating comments at levels lower than l as negative samples and others as positive samples. Note that we only compute F1 scores for $l \in \{1, \ldots, 7\}$ since no comment is at a level lower than 0. The averaged F1 scores is used as an indicator of the overall prediction performance.

Range	Description
0/1	Whether the comment author is the user who initiated the thread.
$\mathbb{Z}_{\geq 0}$	Number of comments made by the author.
	Number of replies to the comment.
	Number of earlier comments.
	Number of later comments.
	Number of sibling comments.
	Number of comments in the subtree rooted from the comment.
	Height of the subtree rooted from the comment.
	Depth of the comment in the tree rooted from the original post.
$\mathbb{R}_{\geq 0}$	Relative comment time (in hours) with respect to the original post.
	Relative comment time (in hours) with respect to the parent comment.
	Normalized[†] number of replies to the comment.
	Normalized[†] number of comments in the subtree rooted from the comment.

Table 1: Content-agnostic features. † means two kinds of normalization are used: 1) zero-mean normalization; 2) divided by the squared-root-rank of the feature value in the thread.

	`AskWomen`	`AskMen`	`Politics`
Baseline	53.6%	49.3%	51.3%
BoW	53.1%	50.9%	51.8%
LDA	55.3%	51.1%	52.5%
Doc2Vec	55.2%	51.7%	53.0%
Factored\M	54.2%	51.8%	52.9%
Factored\R	55.1%	51.9%	53.4%
Factored	**56.3%**	**52.7%**	**54.8%**

Table 2: Averaged F1 scores of DeepOR classifiers using different text features. Baseline results do not use any text features.

5 Experiments

In this section, we evaluate the effectiveness of the factored comment embeddings on the quantized karma prediction task. We use the concatenation of the local mode vector and the content vector at the last time step as the factored comment embedding. First, we study the overall prediction performance of four different classifiers under two settings, i.e., using factored comment embeddings or not. Then we compare the factored comment embeddings inferred from the full model and its two variants with other kinds of text features using the best type of classifiers. Finally, we carry out error analysis on prediction results of the best classifiers using the factored comment embeddings.

5.1 Classifiers

The following four types of classifiers are studied:

- **ShallowLR**: A standard multi-class logistic regression model;
- **ShallowOR**: An ordinal regression model (Rennie and Srebro, 2005), which can exploit the or-

der information of the quantized karma labels;

- **DeepLR**: A feed-forward neural network using the logistic regression objective;
- **DeepOR**: A feed-forward neural network using the ordinal regression objective.

These classifiers have different objectives and model complexities, allowing us to study the robustness of the learned comment embeddings. The factored comment embeddings are inferred from the proposed models trained on the same training data but independently with these classifiers.

As baselines, we train the classifiers using only content-agnostic features, as shown in Table 1, which have strong correlations with community endorsement (Jaech et al., 2015; Fang et al., 2016). In our pilot work, we experimented with several groups of features from (Jaech et al., 2015) to find the content-agnostic features used in our paper. Since Jaech et al. (2015) work on a different task (ranking comments in a short time window), many of the useful content-agnostic features from (Jaech et al., 2015), including k-index, do not give additional improvement over the selected configuration for the karma prediction task.

All classifiers are trained on the training data for each subreddit independently, with hyperparameter tuned on the validation data. The penultimate weights are regularized using L_2 and the regularization parameters are selected in $\{0.0, 0.001, 0.01, 0.1, 1.0\}$. The number of hidden layers for deep classifiers are chosen from $\{1, 2, 3\}$, and the number of hidden neurons is selected from $\{32, 48, 64\}$.

We report the prediction performance on the test data, as shown in Fig. 2. We observe that using comment embeddings consistently improves the performance of these classifiers. While ShallowOR significantly outperforms ShallowLR, indicating the usefulness of exploiting the order information in quantized karma labels, the difference is much smaller for deep classifiers. Also, deep classifiers consistently outperforms their shallow counterparts.

5.2 Text Features

We compare the factored comment embeddings with the following text features:

- **BoW**: A sparse bag-of-word representation;
- **LDA**: A vector of topic probabilities inferred from the topic modeling (Blei et al., 2003);
- **Doc2Vec**: Embeddings inferred from the paragraph vector model (Le and Mikolov, 2014).

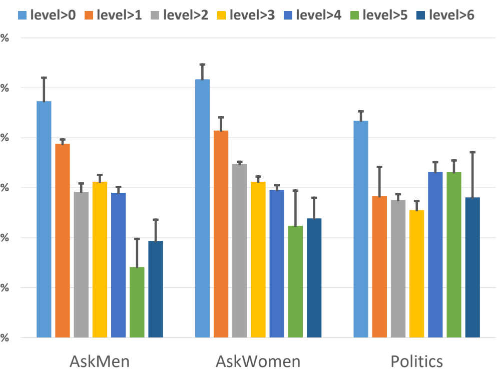
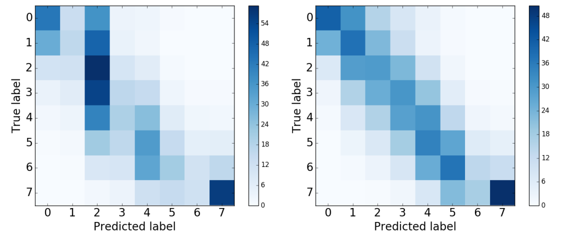


Figure 3: F1 scores of the DeepOR classifier for individual subtasks. Error bars indicate the improvement of using the factored comment embeddings over the classifier using no text features.

For these models, which do not use RNNs, all words in a comment are used. We use the gensim implementations (Řehůřek and Sojka, 2010) for both LDA and Doc2Vec. For LDA, the number of topic is selected in $\{16, 32, 64\}$, and 32 works the best on the validation set for all subreddits. For Doc2Vec, we select the embedding dimension from $\{32, 64, 128\}$, and 64 works the best on the validation set for all subreddits. We train the Doc2Vec for 20 epochs, and the learning rate is initialized as 0.025 and decreased by 0.001 at each epoch.

In addition to the factored comment embeddings obtained from our full model, we study two variants of the full model: 1) a model trained without the mode vector component (**Factored**$\backslash$M), which is a normal sequence-to-sequence attention model (Bahdanau et al., 2015), and 2) a model trained without the response trigger vector component (**Factored**$\backslash$R). All textual representations are used together with the baseline content-agnostic features described previously.

Since the DeepLR and the DeepOR perform best across all subreddits and they have similar trends, we report results of the DeepOR in Tabel 2. Among all text features, the BoW has the worst averaged F1 scores and even hurts the performance for `AskWomen`, probably due to the data sparsity problem. Both the LDA and the Doc2Vec outperform the BoW. The Doc2Vec performs slightly better on `AskMen` and `Politics`, which might be attributed to the relative larger training data size. The factored comment embeddings derived from the full model consistently achieve better averaged F1 scores. It can be observed that the



(a) w/o comment embeddings (b) w/ comment embeddings

Figure 4: The confusion matrices for the DeepOR classifier on Politics. The color of cell on the i-th row and the j-th column indicates the percentage of comments with quantized karma level i that are classified as j, and each row is normalized.

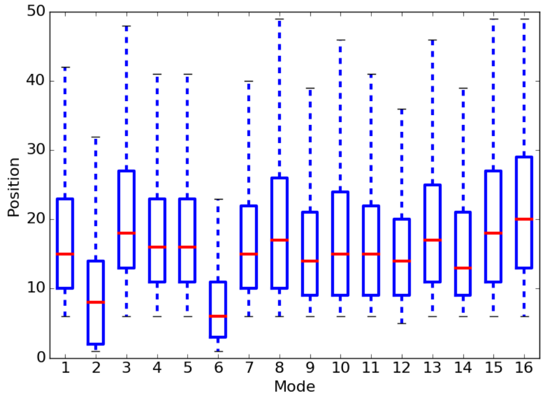


Figure 5: The box plot of strongest association positions for each global mode in Politics.

two variants of the full model mostly lead to similar performance as the Doc2Vec, though the Factored\R embeddings usually have higher averaged F1 scores than the Factored\M embeddings. These results suggest advantages of jointly modeling two components, which may drive the model to discover more latent factors and patterns in the data that could be useful for downstream tasks.

5.3 Error Analysis

In this subsection, we focus on analyzing how factored comment embeddings improve the prediction results of the DeepOR classifiers. The F1 scores for individual subtasks are shown in Fig. 3. Note that the higher the level is, the more skewed the task is, *i.e.* a lower positive ratio. As expected, comments with the lowest endorsement level are easier to classify. Adding comment embeddings primarily boosts the performance of the classifier on the high-endorsement tasks (level $> 5, 6$) and the low-endorsement tasks (level $> 0, 1$).

Confusion matrices for the DeepOR classifier with and without factored comment embeddings are shown in Fig. 4 for Politics. Using the additional comment embeddings leads to a higher concentration of cell weights near the diagonals, corresponding to errors that mainly confuse neighboring levels. Without any text features, the classifier seems to only distinguish four levels. We observe similar trends on AskWomen and AskMen.

6 Qualitative Analysis

In this section, we conduct analysis to better understand what the factored model is learning, again using the Politics subreddit. First, we analyze latent global modes learned from the full model. For each global mode, we extract comments with top association scores. Note that the model assumes a locally coherent mixture of global modes and updates the mixture for each observed word. Thus, each comment receives a sequence of association probabilities over the global modes. The association score β_k between a comment $w_{1:T}$ and Mode-k is then computed as $\beta_k = \max_{t \in \{1,\cdots,T\}} a(\mathbf{c}_t, \mathbf{m}_k)$ for $k \in \{1, \cdots, K\}$, where $a(\mathbf{c}_t, \mathbf{m}_k)$ is defined in (1). In Table 3, we show examples from the most coherent modes out of the 16 learned modes. Some modes seem to be capturing style (modes 2, 6, and 10), while others are related to topics (modes 7 and 16). Mode-2 captures the style of starting with rhetorical question to express negative sentiment and disagreement. Many comments in Mode-6 begin with words of drawing attention such as “bull” and “psst”. Mode-10 tends to be associated with comments telling a story about a closely related person. Many comments in Mode-7 discuss low salaries, whereas Mode-16 comments discuss politicians or ideology of the Republican.

The characteristics of examples in modes 2 and 6 suggested that modes might have a location dependency, so we looked at word positions with the strongest association of each mode, *i.e.* $\text{argmax}_{t \in \{1,\cdots,T\}} a(\mathbf{c}_t, \mathbf{m}_k)$. For each Mode-$k$, we only keep comments with association score higher than $\text{mean}(\beta_k) + \text{std}(\beta_k)$. Fig. 5 shows the box plot of locations where the strongest association happens. It can be seen that modes 2 and 6 usually have the strongest association at the beginning of a comment. For modes 3, 8, 15 and 16, the strongest associations occur over a wider span in comments. In addition to the interpretability of

Mode-2	• Oh come on! Really? One can't make that trip and spend maybe half and save the other for milk, bread and things that do spoil? … • Remind me. How many filibusters did Harry Reid conduct this year? … • Feckless tyrant? How did you do that with your brain? … • Seriously? You have to be registered to vote. … • Holy f*, seriously? This is some heavy duty shit. …
Mode-6	• Bull. Plenty of individuals influence policy by never missing a single chance to vote, no matter how minor the election. … • Bull. Conservatives hate Obamacare so much because if their constituents got mental health treatment, they'd stop voting Republican. • Utter bull s*. Where was the compromise from Obama and the Dems when they pushed through Obamacare without ONE Republican vote. … • psst…it's college • psst- he's "black" - meaning that one of his ancestors is black (as if it's pollutant of some sort).
Mode-7	• …, I used to work 55+ hours a week, salaried, lower quartile salary to boost. … • Or possibly that the standard of living between unemployment and the "jobs" that are out there is really insignificant. … • Where on earth is 7.25 a living wage? If by some miracle you get 40 hours a week that's only $1,160 before taxes. … • If you have to work 40 hours a week to pay your bills that means you are controlled in your fight for survival. … • …Working 15 hours a week for extra pocket money when you're a teen is easy. Working 50 hours a week at fast food to cover rent, food, …
Mode-10	• …Had a guy stalk a trans friend of mine for months trying to terrorize her because … • A co-worker of mine got audited by the IRS because … • …Some conservative friends of mind wanted to meet up at a coffee house with shittier coffee because the other one was too "liberal". … • …Friend of mine works with mentally unstable and aggressive people as part of some social service. … • …A student of mine asked our own AP about an atheist group and he just flat out said "You kidding me?" …
Mode-16	• …These same people will continue to listen to the bullshit that is the Republican Party. And when that happens, they have this twisted reality … • …After spending almost my entire life in Texas and as a Born gain evangelical conservative Republican, I learned my lessons about how completely dishonest and corrupted that entire culture is the hard way. …I will never gain ever vote for or support any kind of conservative. … • …has been our greatest embarrassment, but what makes matter even worse is the support he has for re-election. I would not be surprised … • Well, it is entirely possible that …the underlying cause of Limbaugh's attack was that this guy was playing the type of dirty politics … • …this was more of a referendum on the GOP leadership in Congress by Republican voters, because let's face it, they haven't done anything.…

Table 3: Examples of comments associated with the learned global modes for `Politics`.

the learned modes as one can get from LDA, these observations suggest that our model may further capture word location effects which may help predicting community endorsement.

Next, we analyze the response characteristics by examining the response trigger vectors associated with the onset of comment responses, which is a special start-of-reply token. These response trigger vectors are clustered into 8 classes via k-means and visualized in Fig. 6 using t-SNE (van der Maaten and Hinton, 2008). For each cluster, we study the karma distribution, as well as comments together with the first reply. Related data statistics and examples are shown in Fig. A-4 and Tables A-2&A-3 in the supplementary materials. The horizontal dimension seems to be associated with how many replies a comment elicits. The vertical dimension is less interpretable but most clusters have identifiable traits. The far left classes (Class-1&4) both have few replies and low karma, often two-party exchanges where Class-4 has more negative sentiment. Class-2 comments tend to involve complements, whereas comments in Class-3 usually trigger a reply with *but*-clause for a contrast and disagreement intent. Comments in Class-5 mostly receive responses expanding on the original comments. Class-6 has a lot of sarcastic and cynical comments and replies. Comments in Class-7 are mostly anomalous since their first responses were usually [deleted]. It seems there are multiple response trigger factors in the proposed embedding model, some may reflect dialog acts and others sentiment, any of which may be helpful in predicting community endorsement.

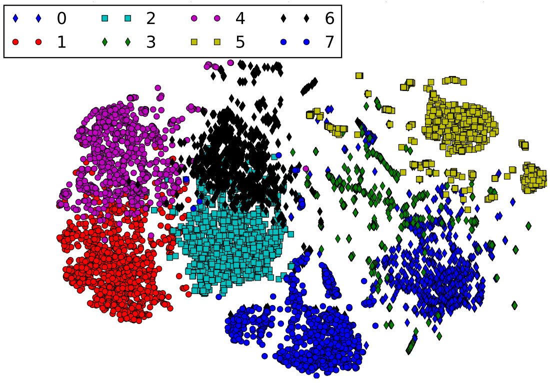


Figure 6: t-SNE visualization of response trigger vectors clustered using k-means.

7 Related Work

The skip-thought vector method (Kiros et al., 2015) is most closely related to our work in terms of utilizing context for unsupervised sequence modeling under the sequence-to-sequence framework (Sutskever et al., 2014). A key difference is the context being exploited. The skip-thought vector method uses surrounding sentences by abstracting the skip-gram structure (Mikolov et al., 2013a) from word to sequence. In our model, we exploit two types of context that are unique in online discussions: 1) the global context such as community topic and style which is learned in the *mode* vectors, and 2) the responses to a comment modeled as the *response trigger* vectors. Moreover, we augment our model with the attention mechanism (Bahdanau et al., 2015) to push the model to distill the relevant information from context.

The neural attention mechanism has been used for a variety of natural language processing tasks, e.g., machine translation (Bahdanau et al., 2015), question answering (Sukhbaatar et al., 2015), constituency parsing (Vinyals et al., 2015), social media opinion mining (Yang and Eisenstein, 2017). and dependency parsing (Cheng et al., 2016). The attention mechanism developed in this paper for exploiting global modes differs from previous work in that the global modes being attended over are *latent* rather than explicitly observed, and in that they are learned jointly with the full model.

Predicting the community endorsement has been studied by using either hand-crafted features (Jaech et al., 2015) or neural models (Fang et al., 2016; Zayats and Ostendorf, 2017), but all of them focus on supervised learning. Unsupervised learning strategies have been explored for characterizing different factors in language. A hierarchical Dirichlet process model was originally proposed for topic variations but has been extended to characterize multiple factors in (Huang and Renals, 2008). While much of the Dirichlet modeling work uses multinomial distributions, a loglinear version is introduced in (Eisenstein et al., 2011). Multi-dimensional structure latent factors in text are modeled by extending the sparsity-promoting topic model in (Paul and Dredze, 2012). Our model instead uses a neural network to characterize latent language factors, where the learned latent language factors could have a dependency on word positions.

8 Conclusion

This paper introduces a new factored neural model for unsupervised learning of comment embeddings leveraging two different types of context in online discussions. By extending the attention mechanism and using residual learning, our method is able to jointly model global context, comment content and response generation. Quantitative experiments on three different subreddits show that the factored embeddings achieve consistent improvement in predicting quantized karma scores over other standard document embedding methods. Analyses on the learned global modes show community-related style and topic characteristics are captured in our model. Also, we observe that response trigger vectors characterize certain aspects of comments that elicit different response patterns.

A potential future direction is to explore whether the comment embeddings derived from the unsupervised factored neural model can be useful across multiple tasks. Recently, a dataset with dialogue act annotations on Reddit discussions is published and can be used for a dialogue act prediction task (Zhang et al., 2017). Identifying or ranking persuasive arguments in the `ChangeMyView` subreddit (as studied in (Tan et al., 2016; Wei et al., 2016)) and asking for favors in the `RandomActsOfPizza` subreddit (used in (Althoff et al., 2014)) are also interesting for future work.

Acknowledgments

This paper is based on work supported by the DARPA DEFT Program. Views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

Tim Althoff, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2014. How to ask for a favor: A case study of the success of altruistic request. In *Proc. Int. AAAI Conf. Web and Social Media (ICWSM).*

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. Int. Conf. Learning Representations (ICLR).*

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Machine Learning Research*, 3:993–1022.

Hao Cheng, Hao Fang, Xiaodong He, Jianfeng Gao, and Li Deng. 2016. Bi-directional attention with agreement for dependency parsing. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2204–2214.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahadanau, Fethhi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1724–1734.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *J. American Society for Information Science*, 41(6):391–407.

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. 2011. Sparse additive generative models of text. In *Proc. Int. Conf. Machine Learning (ICML)*.

Hao Fang, Hao Cheng, and Mari Ostendorf. 2016. Learning latent local conversation modes for predicting community endorsement in online discussions. In *Proc. Int. Workshop Natural Language Process. for Social Media*.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, , and Noah A. Smith. 2015a. Retrofitting word vectors to semantic lexicons. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics (NAACL)*.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015b. Sparse overcomplete word vector representations. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*.

Marco Guerini, Carlo Strapparava, and Gozde Ozba. 2011. Exploring text virality in social networks. In *Proc. Int. AAAI Conf. Web and Social Media (ICWSM)*.

Ji He, Mari Ostendorf, Xiaodong He, Jiansu Chen, Jianfeng Gao, Lihong Li, and Li Deng. 2016a. Deep reinforcement learning with a combinatorial action space for predicting popular Reddit threads. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 195–200.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016b. Deep residual learning for image reconigtion. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Liangjie Hong, Ovidiu Dan, and Brian D. Davison. 2011. Predicting popular messages in Twitter. In *Proc. WWW*.

Songfang Huang and Steve Renals. 2008. Modeling topic and role information in meetings using the hierarchical Dirichlet process. In *Machine Learning for Multimodal Interaction V*, Springer Lecture Notes in Computer Science.

Aaron Jaech, Vicky Zayats, Hao Fang, Mari Ostendorf, and Hannaneh Hajishirzi. 2015. Talking to the crowd: What do people react to in online discussions? In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2026–2031.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learning Representations (ICLR)*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3294–3302.

Himabindu Lakkaraju, Julian McAuley, and Jure Leskovec. 2013. What's in a name? Understanding the interplay between titles, content, and communities in social media. In *Proc. Int. AAAI Conf. Web and Social Media (ICWSM)*.

Quo Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 3104–3112.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 302–308.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Machine Learning Research*, 9.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proc. Workshop at Int. Conf. Learning Representations*.

Tomas Mikolov, Wen-Tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proc. Conf. North American Chapter Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 746–751.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *Proc. IEEE Spoken Language Technologies Workshop*.

Michael J. Paul and Mark Dredze. 2012. Factorial lda: Sparse multi-dimensional text models. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proc. LREC Workshop New Challenges for NLP Frameworks*.

Jason D. M. Rennie and Nathan Srebro. 2005. Loss functions for preference levels: regression with discrete ordered labels. In *Proc. Int. Joint Conf. Artificial Intelligence*.

Bongwon Suh, Lichan Hong, Peter Pirolli, and Ed H. Chi. 2010. Want to be retweeted? large scale analytics on factors impacting retweet in Twitter network. In *Proc. IEEE Second Intern. Conf. Social Computing*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 2431–2439.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3104–3112.

Chenhao Tan, Lillian Lee, and Bo Pang. 2014. The effect of wording on message propagation: Topic- and author-controlled natural experiments on Twitter. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*.

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

Trang Tran and Mari Ostendorf. 2016. Characterizing the language of online communities and its relation to community reception. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1030–1035.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 2755–2763.

Zhongyu Wei, Yang Liu, and Yi Li. 2016. Ranking argumentative comments in the online forum. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 195–200.

Yi Yang and Jacob Eisenstein. 2017. Overcoming language variation in sentiment analysis with social attention. *Trans. Assoc. for Computational Linguistics (TACL)*.

Vicky Zayats and Mari Ostendorf. 2017. Conversation modeling on reddit using a graph-structured LSTM. *arXiv:1704.02080 [cs.CL]*.

Amy Zhang, Bryan Culbertson, and Praveen Paritosh. 2017. Characterizing online discussion using coarse discourse sequences. In *Proc. Int. AAAI Conf. Web and Social Media (ICWSM)*.