Modeling Evolution of Message Interaction for Rumor Resolution

Lei Chen[1], **Zhongyu Wei**[1,2]*, **Jing Li**[3], **Baohua Zhou**[4], **Qi Zhang**[5], **Xuanjing Huang**[5]
[1] School of Data Science, Fudan University, China
[2] Research Institute of Intelligent and Complex Systems, Fudan University, China
[3] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[4] School of Journalism, Fudan University, China
[5] School of Computer Science, Fudan University, China
[1,2,5]{chenl18,zywei,qi_zhang,xjhuang}@fudan.edu.cn
[3]jing-amelia.li@polyu.edu.hk; [4]zhoubaohua@yeah.net

Abstract

Previous work for rumor resolution concentrates on exploiting time-series characteristics or modeling topology structure separately. However, how local interactive pattern affects global information assemblage has not been explored. In this paper, we attempt to address the problem by learning evolution of message interaction. We model confrontation and reciprocity between message pairs via discrete variational autoencoders which effectively reflects the diversified opinion interactivity. Moreover, we capture the variation of message interaction using a hierarchical framework to better integrate information flow of a rumor cascade. Experiments on PHEME dataset demonstrate our proposed model achieves higher accuracy than existing methods.

1 Introduction

With increasing openness of social media platforms, unverified messages can be easily disseminated from person to person and result in tremendous rumor cascades which expose huge threat to individuals and society. To resolve rumors, firstly we need to detect statements that are ambiguous at the time of posting, then explore how users share and discuss rumors and finally assess their veracity as true, false or unverified. This can be represented as a pipeline of sub-tasks, including rumor detection, stance classification and rumor verification (Zubiaga et al., 2018a).

Identifying and debunking rumors automatically has been extensively studied in the past few years. State-of-the-art approaches construct sequential representations following a chronological order and then utilize temporal features to capture dynamic signals (Zubiaga et al., 2016; Ma et al., 2016; Wei et al., 2019). Although the source content stays invariable, time-series modeling successfully locates modifiers who might import evidence to correct misinformation or stir up enmity to discredit truth (Zhang et al., 2013). These models generate promising results, however, they ignore local interactions happened during the message diffusion which is deemed to be important for the identification of rumors.

Figure 1 (a) shows a rumor cascade which is identified as false for devilishly suspect Ray Radley's role in the appalling Sydney siege. As can be seen, denial to false rumor tends to evoke affirmative replies which further confuses the factuality of the message. Besides, disagreement and query towards descriptive statements are able to trigger drastic discussion and result in validity modification. Although some researchers explore propagation structure of rumor proliferation (Ma et al., 2017; Kumar and Carley, 2019), they typically rely on rough aggregation of locally successional messages.

Moreover, the evolution of message interaction depicts the global characteristic of rumor cascades which improves the performance of verification. Figure 1 (b) illustrates the intuition using statistics drawn from PHEME dataset (Kochkina et al., 2018). It can be seen that denial tweets with supportive parent posts appear frequently in false rumors especially in an early stage, while unverified rumors constantly stimulate queries behind positive messages along with time. As rumor cascade evolves, with more dialogue context and auxiliary evidence, assessing the message credibility comprehensively becomes possible.

* Corresponding author

Proceedings of the 28th International Conference on Computational Linguistics, pages 6377–6387
Barcelona, Spain (Online), December 8-13, 2020

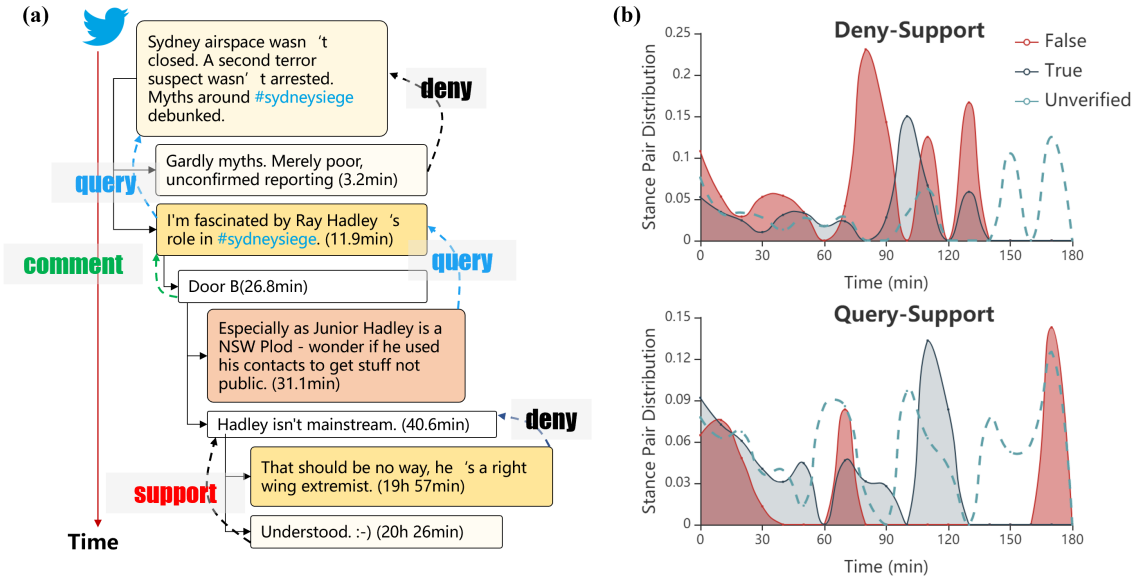

Figure 1: An illustration of how interaction happens during the propagation of messages.

In order to capture local interactive patterns and explore how interactivity dominates global factuality judgment , we propose to learn conversational message interaction and cooperate with propagation structure to improve the performance of rumor resolution. To model message interaction, we learn the latent interactive pattern for a repost toward its original post via discrete variational autoencoders (DVAEs) which has shown great potential in learning categorical latent patterns (interaction patterns in our case). For rumor resolution, latent variables not only represent participant's attitude, but can also control how much literal information is reserved for claim confirmation. We then employ an attention-based hierarchical architecture to capture temporal variation of message interaction.

Our contributions are of three-folds:

- To the best of our knowledge, this is the first study modeling the interactive patterns of messages rather than coarse aggregation for rumor verification. By exploiting interaction between post pairs, we also make it possible to combine propagation structure with time series modeling.
- What's more, we utilize DVAEs to capture the interactive pattern between online conversational discussion and also interpret the latent representation of message interaction associating with stance information.
- Extensive experiments on real-world datasets collected from TWITTER demonstrate our proposed model outperforms state-of-the-art rumor verification methods with large margin.

2 Related Work

Our research is related to two areas including rumor resolution and application of discrete variational autoendocers.

2.1 Rumor Resolution

There have been numerous studies on dismantled tasks of rumor resolution. Traditional approaches (Castillo et al., 2011; Yang et al., 2012; Kwon et al., 2013; Liu et al., 2015) exploit features manually crafted from post text, user profile and media source and use straightforward machine learning algorithms to classify the set of messages. Moreover, rather than only considering properties of individual messages, dynamic time series structure (Ma et al., 2015) and tree model using propagation pattern (Ma et al., 2017) is effective of depicting global difference between rumor and non-rumor claims.

To avoid the effort and bias of feature engineering, methods based on deep neural networks are massively applied and have demonstrated great efficacy of discovering data representation automatically. Ma

et al. (2016) employ recurrent neural networks (RNNs) to capture dynamic temporal signals. Yu et al. (2017) use convolutional neural networks (CNNs) to flexibly extract evidential posts. Recently, Zhou et al. (2019) integrate reinforcement learning to select the minimum number of posts required for early rumor detection. Ma et al. (2019) generate less indicative semantic representation via generative adversarial networks to gain better generalization for rumor detection. Besides, since rumor resolution is a coherent process, researchers also combine detection and stance classification with verification under the framework of multi-task learning (Ma et al., 2018; Kochkina et al., 2018; Kumar and Carley, 2019; Wei et al., 2019).

In summary, deep learning approaches for rumor resolution involves three critical parts: (1) capture local attributes of every single message, (2) integrate information flow to acquire globally coherent representation and (3) explore the synergy effects of local and global information to promote holistic performance. However, it is inadequate to learn interaction between messages via simply sharing model parameters and aggregating information. Our work is closely related to methods based on modeling time-series characteristics (Ma et al., 2016). Different from their work, our proposed model manage to learn the local interactive pattern to assist final verdict and employ attention mechanism to locate messages significantly influence the classification result. Table 1 lists various fundamental modules that latest researches adopt for each part.

Research	Message Modeling	Cascade Modeling	Union Approach
Ma et al. (2016)	-	RNN	-
Yu et al. (2017)	-	CNN	-
Kochkina et al. (2018)	-	BranchLSTM	multi-task learning
Kumar and Carley (2019)	-	TreeLSTM	multi-task learning
Wei et al. (2019)	GCN	RNN	multi-task learning

Table 1: Fundamental components of deep learning approaches for rumor resolution.

2.2 Application of Discrete Variational Autoencoders

Variational Autoencoders (VAEs) are devised to learn low-dimensional latent variables strongly linked with fundamental attributes (Kingma and Welling, 2013) and has shown great promise in smoothly generating diversified sentences from a continuous space (Bowman et al., 2015).

In the setting of VAE, the latent variables are considered independent and continuous in Gaussian latent space. As for datasets composed of discrete classes, discrete latent variables are more suitable to capture the different distribution over the disconnected manifolds. To overcome the problem of training discrete latent variables, Rolfe (2016) proposes the discrete variational autoencoders (DVAEs) which assume that the corresponding prior distribution over the latent space is characterized by independent categorical distributions.

Especially for text mining, discrete variables are adaptive to holistic properties of text and much more friendly for interpreting categories of natural language such as style, topic and high-level syntactic features. For instance, in neural dialog generation, DVAE is able to learn underlying dialogue intentions that can be interpreted as actions guiding the generation of machine responses (Wen et al., 2017; Zhao et al., 2018). In this paper, we learn discrete latent variables between inherited post pairs and incorporate them with textual information to model message interaction.

3 Proposed Model

Resolution of rumor cascades can be formulated as a supervised classification problem. Given a tree-stuctured TWITTER cascade $\mathcal{C}$ which corresponds to a root tweet r_0 and its relevant responsive tweets $\{r_1, r_2, ..., r_T\}$, the goal is to recognize the stance of each tweet $\mathcal{Y}_i^s$ as *support*, *comment*, *deny* or *query*, as well as determine the class of the cascade $\mathcal{Y}_v$ as *true*, *false* or *unverified*. From our dataset, for each tweet r_i, its post time t_i and parent post r_i^p from which it retweets is also available.

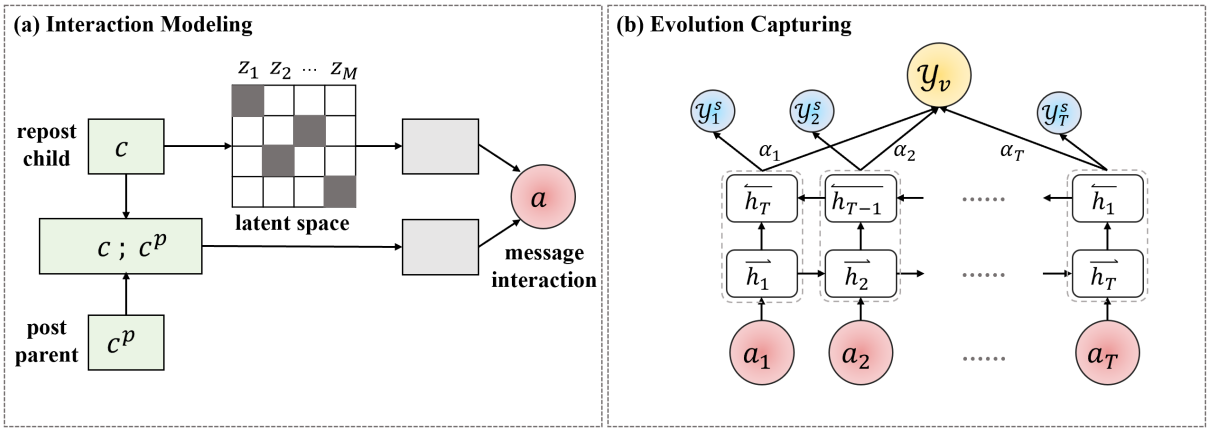

Figure 2: Sub-modules of our proposed model: (a) modeling message interaction via DVAEs and (b) the attention-based hierarchical structure for integration and classification.

Our model is based on a hierarchical architecture which consists of two components: (1) interaction modeling which cooperates child post with its parents via DVAEs to generate message interaction and (2) evolution capturing that employs attention-based recurrent neural networks to capture temporal variation and make prediction, as shown in Figure 2.

3.1 Interaction Modeling

We use mean of glove word vectors to encode the textual information for each post and then employ DVAEs to explore the relationship between post pairs so as to generate representation for message interaction.

Post Representation. For each tweet r, we represent the textual information as a sequence of words $\{w_1, w_2, ..., w_n\}$. Besides, we extract its post time t and look up corresponding parent post r^p for further use.

Given a sequence of words $\{w_1, w_2, ..., w_n\}$, an embedding layer map each w_i into a dense vector $\boldsymbol{x}_i$,

$$\boldsymbol{x}_i = \boldsymbol{E}w_i,\ i = 1, 2, ..., n \tag{1}$$

where $\boldsymbol{E}$ is the embedding matrix, $\boldsymbol{x}_i$ is the embedding form of the word w_i.

Then we take the average of these word embeddings to obtain the sentence-level representation $\boldsymbol{c}$. Similarly, we can obtain representation of the corresponding parent post $\boldsymbol{c}^p$ according to r^p. Besides, we have also tried other complex methods of sentence representation, including CNNs, RNNs and pre-trained BERT embeddings. They are not as effective as in other tasks since text in TWITTER contains numerous informal expressions and they are likely to intensify the semantic gap under the setting of cross-event validation.

Latent Interaction Modeling. To model message interation, we propose to explore the relationship between three random variables: the repost tweet $\boldsymbol{c}$, the parent post $\boldsymbol{c}^p$ and the latent interactive pattern $\boldsymbol{z}$. Before introducing our adaption of DVAEs, we identify two key properties of tweet claim formulation in the first place.

On one hand, the latent meaning of $\boldsymbol{z}$ should be independent of $\boldsymbol{c}^p$ since there is high probability for contradictory opinions to appear after the same original post. On the other hand, different from text generation, the latent action $\boldsymbol{z}$ is the product of interaction between $\boldsymbol{c}$ and $\boldsymbol{c}^p$ and should reciprocate with textual information to guide rumor discrimination. Thus, our DVAEs include two critical modules, (1) a recognition network $\mathcal{R}$: $q_{\mathcal{R}}(\boldsymbol{z}|\boldsymbol{c})$ that recognizes attitude of a retweet post; (2) a policy network $\boldsymbol{\pi}$: $p_{\boldsymbol{\pi}}(\boldsymbol{a}|\boldsymbol{z}, \boldsymbol{c}, \boldsymbol{c}^p)$ that constrains the distribution of $\boldsymbol{z}$ and incorporates textual information to form interaction $\boldsymbol{a}$, as shown in Figure 2(b).

In the setting of DVAEs, the latent action $\boldsymbol{z}$ is a series of K-way categorical variables $\{z_1, z_2, ..., z_M\}$, where z_i is independent with each other and M is the number of latent variables. Conditioning on the retweet post $\boldsymbol{c}$, the recognition network calculates the temporary logits of latent space $\boldsymbol{\ell}$ by a single full-connected layer,

$$\boldsymbol{\ell}_i = \tanh(\boldsymbol{W}_{\ell i}\boldsymbol{c} + \boldsymbol{b}_{\ell i}),\ i = 1, ..., M \tag{2}$$

where $\boldsymbol{W}_{\ell i}$ and $\boldsymbol{b}_{\ell i}$ are weight matrix and bias vector.

As simulating the distribution of $\boldsymbol{z}$ from $\boldsymbol{\ell}$ by softmax operation presents great challenge for back propagation, we apply Gumbel-Softmax trick to create a derivable estimator for categorical variables (Maddison et al., 2016; Jang et al., 2016). A random variable g has a standard Gumbel distribution if $g = -\log(-\log(u))$, with $u \sim U(0, 1)$. Let $\{g_1, g_2, ..., g_k\}$ be an *i.i.d* sequence of Gumbel random variables, by adding the Gumbel noise g_k to $\log \ell_{ik}$, the categorical distribution could be appropriately reparameterized. Then a relaxation by introducing a temperature parameter τ makes it possible to implement a continuous approximation and provides guarantee for optimization.

With Gumbel-Softmax trick, we obtain separated elements of the posterior distribution $q_{\mathcal{R}}(z_i|\boldsymbol{c})$ as,

$$d_{ik} = \frac{e^{(\ell_{ik}+g_k)/\tau}}{\sum\limits_{k} e^{(\ell_{ik}+g_k)/\tau}} \tag{3}$$

with higher τ, the vector d_i is much smoother and even seems continuous.

Then, the discrete code of each z_i can be acquired.

$$z_i = \underset{k\in[1,2,...,K]}{\arg\max}\ d_{ik} \tag{4}$$

In the policy network, we concatenate $\boldsymbol{c}$ and $\boldsymbol{c^p}$ to form semantic signal and combine the signal with the learned latent interactive pattern $\boldsymbol{z}$ to generate a control vector $\boldsymbol{a}$ which represents message interaction,

$$\boldsymbol{a} = \boldsymbol{W}_a^0\boldsymbol{z} \oplus [\text{sigmoid}(\boldsymbol{W}_a^1\boldsymbol{z} + \boldsymbol{b}_a^1) \cdot (\boldsymbol{c} \oplus \boldsymbol{c}^p)] \tag{5}$$

where $\boldsymbol{W}_a^0$ and $\boldsymbol{W}_a^1$ are weight matrix, $\boldsymbol{b}_a^1$ is bias vector, and $\oplus$ denotes the concatenate operation. In Equation 5, sigmoid gate allows $\boldsymbol{z}$ to control the degree of semantic information flowing from the post representation.

In order to demonstrate discrete latent variables are more effective than continuous, we also compare the performance while following the framework proposed by Bowman et al. (2015) to obtain the continuous latent variables $\boldsymbol{z}$.

3.2 Evolution Capturing

After exploring the interactivity between messages, we employ and modify the dynamic time series model (Ma et al., 2016) to capture temporal variation of these interactive information, as shown in Figure 2(b). Different from their preprocessing procedure, we remove the tedious process of time series partitioning since the average cascade size of the dataset we use is relatively small and simplifying data storage structure is more friendly for batch training. Then bidirectional LSTM layers are employed on these sequential message interactions to obtain the intermediate hidden states $\boldsymbol{h}_i^j$.

$$\boldsymbol{h}_i^j = \text{BiLSTM}(\boldsymbol{h}_{i-1}^j, \boldsymbol{h}_i^{j-1}) \tag{6}$$

where j means the jth LSTM layer and h_i^{j-1} equals to a_i at the first layer.

Then we utilize the inner hidden states to output stance labels $\hat{y}_1^s, \hat{y}_2^s, ..., \hat{y}_T^s$ in the framework of multi-task learning. Although the bidirectional LSTM networks could have several layers, we use the first layer of hidden states as the source of stance output because they are closer to the original local representation.

After obtaining coherent global representation of each message, an attention pooling layer is used as a last step of integration in order to capture contribution imbalance. For the last layer of hidden states

$\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_T$, we calculate the cascade representation $\boldsymbol{s}$ as follow,

$$\boldsymbol{m}_i = \tanh(\boldsymbol{W}_m \boldsymbol{h}_i + \boldsymbol{b}_m) \tag{7}$$

$$u_i = \frac{e^{\boldsymbol{w}_u \boldsymbol{m}_i}}{\sum_j e^{\boldsymbol{w}_u \boldsymbol{m}_j}} \tag{8}$$

$$\boldsymbol{s} = \sum_i u_i \boldsymbol{h}_i \tag{9}$$

where $\boldsymbol{W}_m$ and $\boldsymbol{w}_u$ are weight matrix and vector, $\boldsymbol{b}_s$ is the bias vector and u_i represents the attention weights.

Finally, one linear layer is applied on the cascade representation $\boldsymbol{s}$ to get the prediction result $\hat{y}$.

3.3 Joint Learning

For one thing, our proposed model aims at modeling the interactivity between messages, and for another, the ultimate goal is to make precise discrimination for rumor claims. As a result, the objective of the overall framework has to consider effects from two aspects. We define the loss function as,

$$\mathcal{L} = \mathcal{L}_v + \mathcal{L}_s + \lambda \mathcal{L}_{\text{DVAE}} \tag{10}$$

where λ is a tradeoff hyperparameter to balance the task-oriented loss and DVAE loss.

The first two loss term is defined on the rumor resolution task. We adopt the well-known cross entropy loss,

$$\mathcal{L} = -\frac{1}{N} \sum_i^N \sum_j^L y_i^j \log \hat{y}_i^j \tag{11}$$

where N is the number of instances, L is the number of considered classes.

The last term is defined on the generation validity of DVAEs. To carry out inference for interaction modeling, we introduce a parameterized network $q_\Phi(\boldsymbol{z}|\boldsymbol{c}, \boldsymbol{c}^p)$ to approximate the posterior distribution $p_{\boldsymbol{\pi}}(\boldsymbol{z}|\boldsymbol{c}, \boldsymbol{c}^p)$. Since it is a trainable parameter space, we simplify the expression as $q_\Phi(\boldsymbol{z})$. Then we can write the objective of DVAEs as follow.

$$\mathcal{L}_{\text{DVAE}} = \mathbb{E}_{q_\Phi(\boldsymbol{z})}[\log p_{\boldsymbol{\pi}}(\boldsymbol{a}|\boldsymbol{z}, \boldsymbol{c}, \boldsymbol{c}^p)] - D_{\text{KL}}(q_\Phi(\boldsymbol{z})||q_{\mathcal{R}}(\boldsymbol{z}|\boldsymbol{c})) \tag{12}$$

Inspired by the decomposition work from Zhao et al. (2018), we use cross entropy to approximate the reconstruction loss and derive the KL-divergence through customary calculation.

4 Experiments

4.1 Data Set

We evaluate our interaction-aware model on real-world dataset collected from TWITTER which is developed by Kochkina et al. (2018). It contains rumor and non-rumor claims related to 5 breaking news and each of the rumor claims is annotated with its credibility, either true, false or unverified. In addition, the dataset constructor supplement sparse stance information (Zubiaga et al., 2018b) so that multi-task learning is able to show its validity and we can implement further analysis to confirm the effectiveness of message interaction. Among these two tasks, verification is labeled on cascade-level while stance belongs to tweet-level annotations. PHEME is undoubtedly suitable for our exploration of message interaction as it is constructed by a large amount of conversational threads in which participants tend to launch discussion other than judge on the source tweet.

4.2 Preprocessing and Training Details

We preprocess each tweet by the NLTK toolkit (Bird et al., 2009) and follow a procedure of removing *url* and @, tokenizing, lemmatizing, and removing all the stop words. Glove (Pennington et al., 2014) word embeddings with dimension of 300 are adopted without being fine-tuned. As for training process, we

perform leave-one-event-out (LOEO) cross validation (Kochkina et al., 2018). Although it suffers a lot to handle problems such as evil-balanced instances for each event and semantic inconsistency between events, LOEO is much more representative of real world and has been adopted by latest researches (Kumar and Carley, 2019; Wei et al., 2019).

Hyperparameters performing best in development set are fixed and recorded. The network is trained with back propagation using the Adagrad update rule (Duchi et al., 2011). Following is the final hyperparameters of best performed network. For the module of DVAEs, the number of disrete variables M is set as 4, the possible number of each varible K is 4 and the temperature τ equals to 10. For the integration part, the number of hidden unit is 200, with a dropout rate of 0.3. While training, the batch size is set as 32, the maximum number of training epochs is 50, and the tradeoff parameter of loss terms is 0.4. We assign verification and stance classification tasks with different start learning rate, namely 1e-5 and 1e-4 respectively, because these two tasks share the same input while most of the stance labels are missing which requires larger learning rate to catch up. We have made our code and preprocessed data publicly available [1].

4.3 Models for Comparison

We compare our model with the following models:

RNN: A RNN-based model (Ma et al., 2016) with GRU to capture dynamic textual variation.

CNN: A CNN-based rumor detection model (Yu et al., 2017) to locate key information.

BranchLSTM: A branchLSTM-based network (Kochkina et al., 2018) that cooperates detection and stance classification task to boost verification.

TreeLSTM: A treeLSTM-based networ (Kumar and Carley, 2019) to encode cascade information with multi-task learning.

GCN-RNN: A combination model (Wei et al., 2019) which uses GCN to update message and employs RNN to acquire cascade representation.

VAE-RNN: Our proposed model alternating discrete latent variables as continuous.

DVAE-RNN: Our proposed model that considers time series effect and propagative interactivity at the same time.

4.4 Overall Performance

We implement the task of rumor verification and stance classification to evaluate the performance of our proposed model.

Rumor verification. The overall results for rumor verification are shown in Table 2. We can see that our interaction-aware model significantly outperforms all the models nearly across all the metrics, especially recognizing misleading messages (false rumor) which is extremely important for practical use. On the whole, methods using multi-task learning are more robust than others that don't. Compared with plain RNN model, introducing local information modeling brings about performance improvement which illustrates that to measure the whole cascade's attribute, the local attribute of interaction needs to be considered. Comparing with VAE-RNN, the discrete variables is more representative of the latent interactive patterns as the input of neural networks is already a form of continuous dense vectors.

Stance classification. Under the framework of multi-task learning, we also test the performance of stance classification, as shown in Table 3. Our proposed model achieves the highest accuracy and macro F1-score, even though some other methods reach a sudden performance boost testing on certain event or stance. The main reason is that stance classification is much more dependent on the semantics of the tweet and its surrounding claims, and the huge semantic gap between the event-related corpus brings about the drastic fluctuation. Compared with VAE-RNN that converts the discrete variable into continuous, the exceedance indicates that using discrete latent variables are more suitable to represent categorical information.

[1] https://github.com/lchen96/rumor_interaction

Method	Acc.	MaF	FG	SS	GC	OS	CH	F_F	F_T	F_U
RNN	0.542	0.353	0.305	0.337	0.400	0.358	0.363	0.204	0.473	0.381
CNN	0.516	0.344	0.327	0.320	0.406	0.312	0.355	0.224	0.480	0.328
BranchLSTM*	0.470	0.329	0.189	0.350	0.429	0.352	0.327	0.181	0.524	0.278
TreeLSTM	0.552	0.369	0.314	0.348	0.443	0.360	0.379	0.210	0.511	0.385
GCN-RNN	0.609	0.382	0.338	0.361	**0.455**	0.388	0.370	0.237	0.524	0.386
VAE-RNN	0.533	0.367	0.313	0.321	0.441	0.371	0.389	0.208	0.503	**0.391**
DVAE-RNN	**0.610**	**0.400**	**0.401**	**0.362**	0.441	**0.398**	**0.400**	**0.286**	**0.535**	0.380

Table 2: Results for rumor verification. MaF: the value of macro F1-score, **Bold**: the best performance in each column. 5 columns in the middle represent the macro F1-score using different event data as the test data. 3 columns on the right show the averaged F1-score of false,true and unverified rumors. '*' denotes values taken from the original publication.

Method	Acc.	MaF	FG	SS	GC	OS	CH	F_S	F_C	F_D	F_Q
RNN	0.647	0.447	0.368	0.441	0.520	0.436	**0.468**	0.446	0.761	0.137	**0.442**
CNN	0.681	0.425	0.377	0.421	0.477	0.426	0.422	0.474	**0.799**	0.106	0.319
BranchLSTM*	-	0.460	0.373	0.446	0.543	0.475	0.465	-	-	-	
TreeLSTM	0.681	0.464	**0.401**	0.431	**0.580**	0.446	0.462	0.513	0.790	0.127	0.425
GCN-RNN	0.633	0.433	0.364	0.419	0.528	0.420	0.433	0.489	0.757	0.141	0.345
VAE-RNN	0.644	0.444	0.367	0.456	0.520	0.413	0.464	0.433	0.761	**0.158**	0.425
DVAE-RNN	**0.689**	**0.471**	0.400	**0.487**	0.521	**0.480**	0.466	**0.532**	0.780	0.155	0.400

Table 3: Results for stance classification. MaF: the value of macro F1-score, **Bold**: the best performance in each column. 5 columns in the middle represent the macro F1-score using different event data as the test data. 4 columns on the right show the averaged F1-score of classifying supporting, commenting, denying and querying messages. '*' denotes values taken from the original publication.

4.5 Further Analysis on Interaction Modeling

In order to analyze the effectiveness of DVAEs for interaction modeling, we propose to use the stance information as assistance. Our model attempts to learn the latent vector aroused by a specific post but constrained by the parent-relevant distribution which means the interaction we modeled is heavily depend on the pair relationship between parent post and its repost. Besides, with the design of attention-based integration strategy, we are able to locate what kind of message interaction dominantly determine the classification of rumor cascades.

Using the model with best performance, we calculate the average attention weights of different stance pairs to estimate if interactive patterns assist in verifying rumors. The distribution of attention weights of different interaction patterns can be seen in Figure 3. It is obvious that supportive or denial posts with parent holding the same stance play a critical part in verifying rumors, and discussion aroused by judgemental (supporting/denying) tweets immensely promote the process of identification.

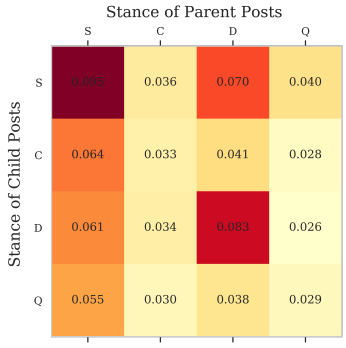

Figure 3: Average attention weights of stance pairs.

4.6 Hyperparameter Sensitivity

In this section, we explore the influence of three hyper-parameters, namely the trade-off factor λ, the number of discrete latent variables M and categories for each latent variable K.

Impact of λ. In order to investigate the influence level of interactive effect, we set the tradeoff factor λ as 0, 0.2, 0.4, 0.6, 0.8, 1 respectively to control the dominance of message interaction modeling. As shown in Figure 4, we observe that with λ set to 0.4, our model achieves the highest accuracy for rumor detection and verification. Even when the value of λ descends to 0, the model is still robust as a result of plain integration of message pairs. Nevertheless, we figure that with the increase of λ, our proposed model gradually presents the effectiveness for rumor verification. As the assessment criteria is task-oriented, thereupon, with larger λ, the generation of DVAEs is likely to become discretionary so that the test accuracy decreases rapidly.

Impact of M and K. Furthermore, We explore the optimal scope for the latent space $\boldsymbol{z}$ by tuning M and K. With a mass of experimental practice, we confirm when setting M and K both at 4, our framework works best. Figure 4 illustrates the result of varying M and K compared with plain hierarchical structure. Varying M affects little for the classification result. The reason probably lies in the independence of each z_i. However, the augment of K brings about disastrous decline of prediction exactitude. This is principally because a large K makes it more difficult to approximate the complex posterior distribution.

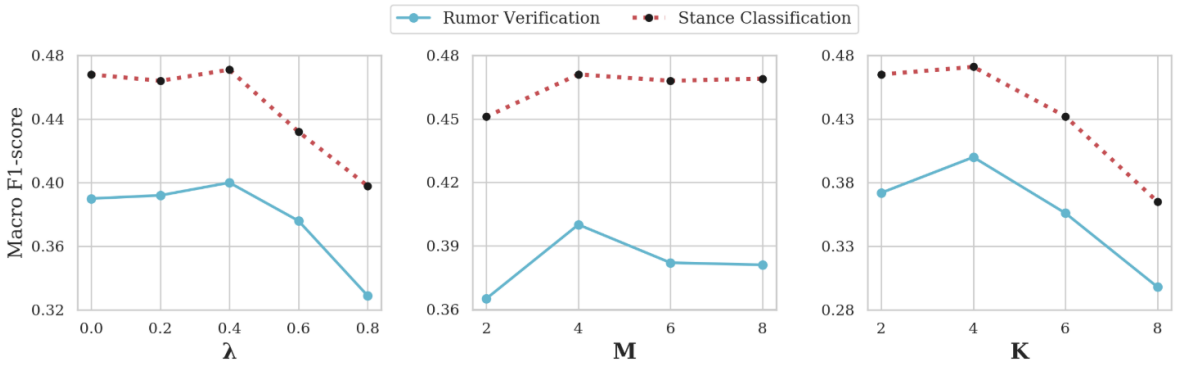

Figure 4: The macro F1-score fluctuation of varying the tradeoff factor λ the number of discrete latent variables M and categories for each latent variable K.

5 Conclusion and Future work

In this paper, we propose to model the evolution of message interaction for rumor resolution. The interaction pattern between post repost pairs is modeled via via discrete variational autoencoers. And an attention-based hierarchical architecture is employed to capture the evalution of message interactions. Experimental results on PHEME dataset show that our framework significantly outperforms the baselines for rumor verification. Further analysis shows that DVAEs is able to model interaction features for better interaction pattern identification. Besides, a closer look at attention weights present that some specific types of interactions contribute more on rumor resolution.

In the future, we would like to explore the task of interaction type classification to further analyze the influence of various interaction types on rumor resolution. In addition, it would be interesting to identify those change points along the timeline when misinformation emerges.

Acknowledgement

This work is partially supported by National Natural Science Foundation of China (No.71991471), National Social Science Foundation (No.20ZDA060), National Key Research and Development Plan (No.2018YFC0830600), Science and Technology Commission of Shanghai Municipality Grant (No.20dz1200600, No.18DZ1201000, No. 17JC1420200).

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684. ACM.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task learning for rumour verification. *arXiv preprint arXiv:1806.03713*.

Sumeet Kumar and Kathleen M Carley. 2019. Tree lstms with convolution units to predict stance and rumor veracity in social media conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5047–5058.

Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *2013 IEEE 13th International Conference on Data Mining*, pages 1103–1108. IEEE.

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1867–1870. ACM.

Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1751–1754. ACM.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *Ijcai*, pages 3818–3824.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 708–717.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Detect rumor and stance jointly by neural multi-task learning. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 585–593. International World Wide Web Conferences Steering Committee.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect rumors on twitter by promoting information campaigns with generative adversarial learning.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jason Tyler Rolfe. 2016. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*.

Penghui Wei, Nan Xu, and Wenji Mao. 2019. Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity. *arXiv preprint arXiv:1909.08211*.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. 2017. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3732–3741. JMLR. org.

Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 13. ACM.

Feng Yu, Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan, et al. 2017. A convolutional approach for misinformation identification.

Yichao Zhang, Shi Zhou, Zhongzhi Zhang, Jihong Guan, and Shuigeng Zhou. 2013. Rumor evolution in social networks. *Physical Review E*, 87(3):032133.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. *arXiv preprint arXiv:1804.08069*.

Kaimin Zhou, Chang Shu, Binyang Li, and Jey Han Lau. 2019. Early rumour detection. In *Naccl*.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2016. Learning reporting dynamics during breaking news for rumour detection in social media. *arXiv preprint arXiv:1610.07363*.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018a. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):32.

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein. 2018b. Discourse-aware rumour stance classification in social media using sequential classifiers. *Information Processing & Management*, 54(2):273–290.