University of Sheffield

Examining the Limitations of Computational Rumor Detection Models Trained on Static Datasets

@LREC-COLING 2024

Yida Mu, Xingyi Song, Kalina Bontcheva, Nikolaos Aletras

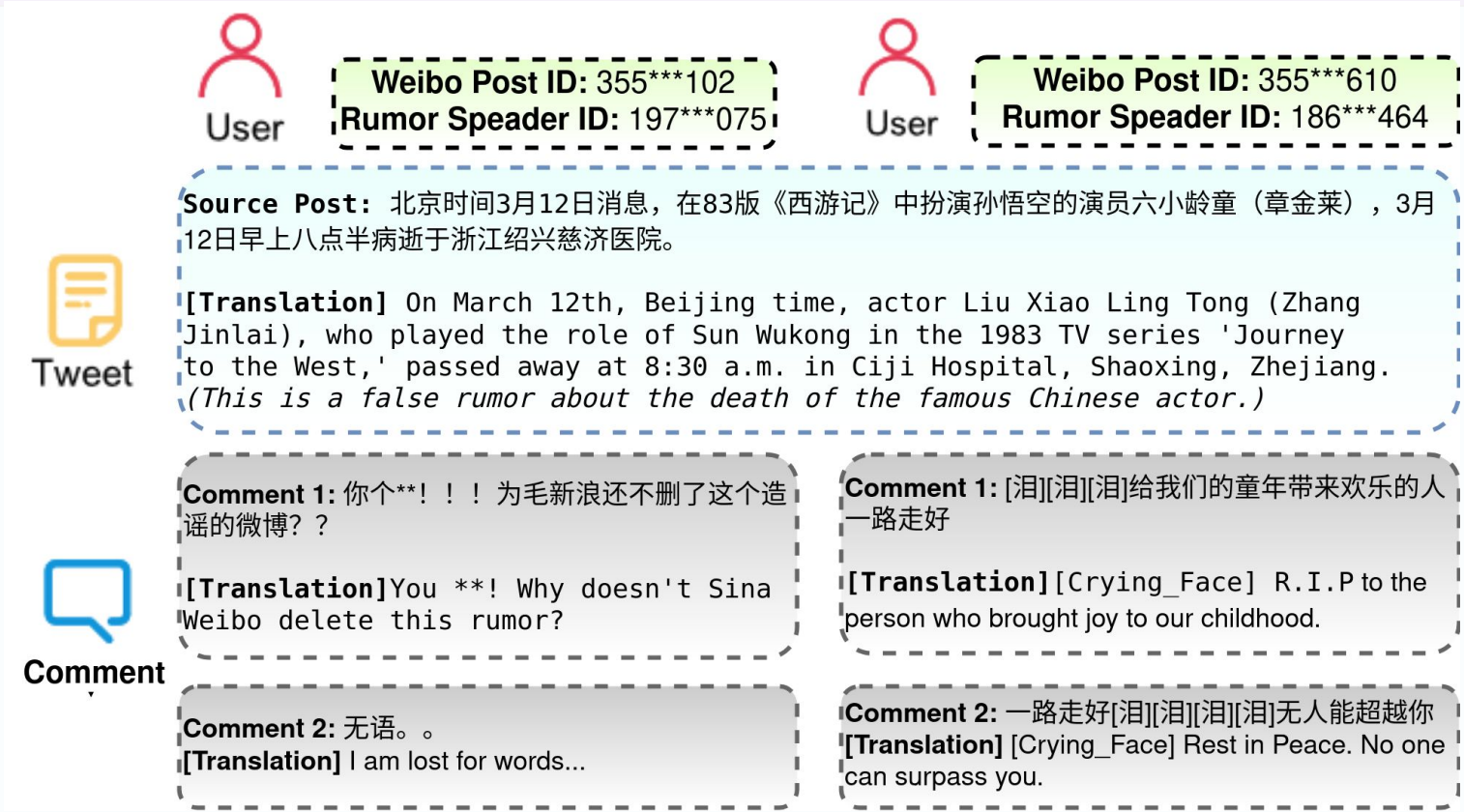
Department of Computer Science, The University of Sheffield

{y.mu, x.song, k.bontcheva, n.aletras}@sheffield.ac.uk

Research Motivation

A crucial aspect of a rumor detection model is its ability to generalize, particularly its ability to detect emerging, previously unknown rumors. Past research has indicated that content-based (i.e., using solely source post as input) rumor detection models tend to perform less effectively on unseen rumors. At the same time, the potential of context-based models remains largely untapped. The main contribution of this paper is in the in-depth evaluation of the performance gap between content and context-based models specifically on detecting new, unseen rumors. Our empirical findings demonstrate that context-based models are still overly dependent on the information derived from the rumors' source post and tend to overlook the significant role that contextual information can play. We also study the effect of data split strategies on classifier performance. Based on our experimental results, the paper also offers practical suggestions on how to minimize the effects of temporal concept drift in static datasets during the training of rumor detection methods.

Example


User
Weibo Post ID: 355***102
Rumor Speader ID: 197***075
User
Weibo Post ID: 355***610
Rumor Speader ID: 186***464
Tweet
Source Post: 北京时间3月12日消息，在83版《西游记》中扮演孙悟空的演员六小龄童（章金莱），3月12日早上八点半病逝于浙江绍兴慈济医院。
[Translation] On March 12th, Beijing time, actor Liu Xiao Ling Tong (Zhang Jinlai), who played the role of Sun Wukong in the 1983 TV series 'Journey to the West,' passed away at 8:30 a.m. in Ciji Hospital, Shaoxing, Zhejiang. (This is a false rumor about the death of the famous Chinese actor.)
Comment
Comment 1: 你个**！！！为毛新浪还不删了这个造谣的微博？？
[Translation]You **! Why doesn't Sina Weibo delete this rumor?
Comment 1: [泪][泪][泪]给我们的童年带来欢乐的人一路走好
[Translation][Crying_Face] R.I.P to the person who brought joy to our childhood.
Comment 2: 无语。。
[Translation] I am lost for words...
Comment 2: 一路走好[泪][泪][泪][泪]无人能超越你
[Translation] [Crying_Face] Rest in Peace. No one can surpass you.

Main Contribution

• Empirical proof (§ 4.1 & 5) that despite having additional contextual information, rumor detection models still struggle to detect unseen rumors appearing at a future date, with some models performing even worse than random baselines (see Table 3).

• An ablation study (§ 5.3) that removes source posts from the inputs, revealed that current rumor detection approaches rely excessively on information from the source post, while neglecting the contextual information.

• A follow-up similarity analysis (§ 5.4) on content and context-based features, which elucidates the impact of training/test split strategies on model performance.

• Finally, we focus on the issue of effectively utilizing static datasets for rumor detection by providing practical recommendations (§ 6), such as implementing additional cleaning measures for the static dataset and enhancing the current evaluation metrics.

Data

- **Twitter 15 & Twitter 16** (Ma et al., 2017) are two English datasets that include tweets categorized into one of four categories: True Rumor (T), False Rumor (F), Non-rumor (NR) and Unverified Rumor (U).

- **Weibo 16** (Ma et al., 2017) consists of 4,664 Weibo posts in Chinese. It comprises 2,313 false rumors debunked by the official Weibo Fact-checking Platform and 2,351 non-rumors sourced from mainstream news sources.

- **Weibo 20** (Rao et al., 2021) is a Chinese rumor detection dataset similar to Weibo 16. It provides 3,034 non-rumors and 3,034 false rumors from the same Weibo fact-checking platform as Weibo 16.

- **Sun-MM** (Sun et al., 2021) comprises 2,374 annotated tweets (i.e., rumor or non-rumor) that cover both textual (i.e., source post) and visual (i.e., image) information. It is typically used for multi-modal rumor detection.

Models

SVM-HF (Source Post + User Profile) Similar to (Yang et al., 2012; Ma et al., 2015), we use a linear SVM model using source posts represented with TF-IDF and various handcrafted features extracted from user profile attributes e.g., number of followers, account status (i.e., whether a verified account or not), number of historical posts, etc.

BERT (Source Post) In line with previous work (Rao et al., 2021; Tian et al., 2022), we use solely source posts as input to fine-tune the Bert-base model[55]We use bert-base-uncased and bert-base-chinese models from Hugging Face (Wolf et al., 2020) for English and Chinese datasets respectively. (Devlin et al., 2019) by adding a linear layer on top of the 12-layer transformer architecture with a softmax activation. We consider the special token '[CLS]' as the post-level representation.

Bi-GCN (Comment Network) To model the network of comment propagation, we use Bi-Directional Graph Convolutional Networks (Bi-GCN) (Bian et al., 2020). Bi-GCN employs two separate GCNs with (i) a top-down directed graph representing rumor spread to learn the patterns of rumor propagation; and (ii) another GCN with an opposite directed graph of rumor diffusion.

Hierarchical Transformers (Source Post + Comment Sequence) Similar to prior work (Rao et al., 2021; Tian et al., 2022), we use a hierarchical transformer-based network to encode separately the source post and its sequence of comments.[66]Given that the total number of tokens of the source post and all comments exceeds the maximum input length (i.e., 512 tokens) of most Bert-style models. We then add a self-attention and a linear projection layer with softmax activation to combine the hidden representation of posts and comments.

Hybrid Vision-and-Language Representation (Source Post + Image) We use visual transformer[77]https://huggingface.co/google/vit-base-patch16-224 (ViT) (Dosovitskiy et al., 2020) and BERT (Devlin et al., 2019) to represent images and source posts of rumors for the Sun-MM dataset. We then combine the two hidden representations by adding a fully connected layer with softmax activation for rumor classification.

Data Splits

• **Forward Chronological Splits** For each dataset, we initially sort all rumors chronologically, from the oldest to the newest. We then divide them into three subsets: a training set (containing 70% of the oldest rumors), a development set (10% of the rumors that were posted after those in the training set but before those in the test set), and a test set (containing the 20% most recent rumors). This data split strategy allows the model to be trained and fine-tuned on older rumors and then be evaluated on the most recent ones.

• **Backward Chronological Splits** In contrast, here all rumors are sorted starting from the most recent ones to the oldest ones, and then are split in the same way as the forward chronological splits. This allows the model to be trained on the newest rumors and evaluated on the oldest ones.

• **Random Splits** This is the most commonly adopted data split strategy in prior work. All datasets are divided into three subsets using a stratified random split approach

Data Splits

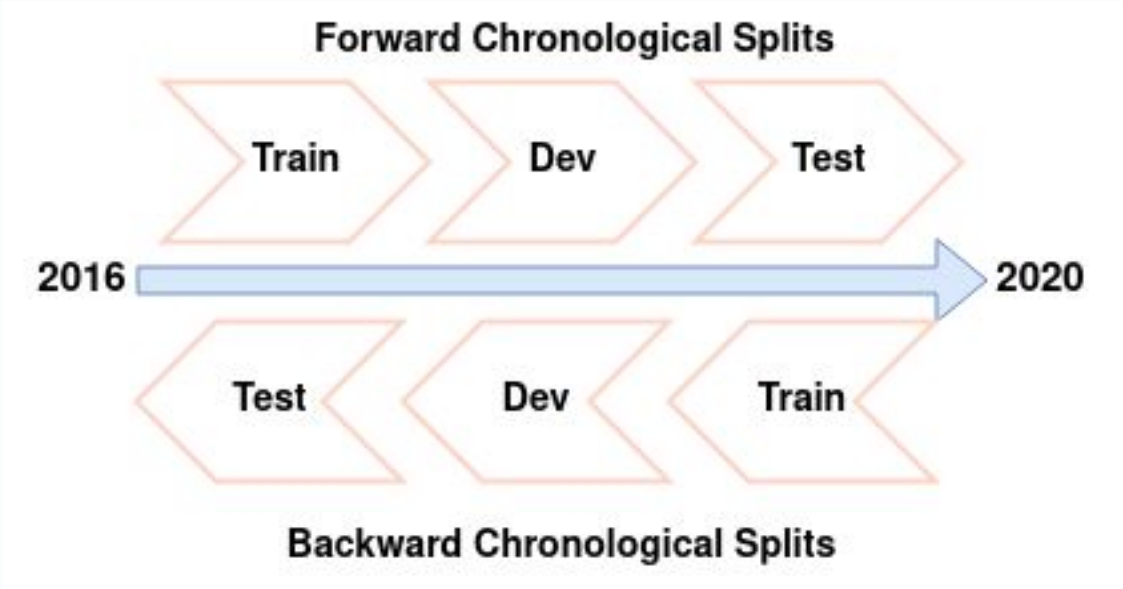


Figure 2: An example of using forward and backward chronological data splits on Weibo 20 dataset (including rumors from 2016 to 2020). There is no overlap among the three subsets.

Results

Models & Splits		Twitter 15					Twitter 16				
		Acc.	*NR* F1	*F* F1	*T* F1	*U* F1	*Acc.*	*NR* F1	*F* F1	*T* F1	*U* F1
Weak Baseline		0.240	0.224	0.246	0.238	0.254	0.248	0.174	0.250	0.300	0.264
SVM-HF	*Random*	**0.739**	**0.727**	**0.701**	**0.803**	**0.728**	**0.709**	**0.697**	**0.602**	**0.858**	**0.663**
	Forward	0.413	0.589	0.366	0.092	0.304	0.373	0.523	0.226	0.297	0.214
	Reverse	0.353	0.590	0.462	0.063	0.062	0.380	0.520	0.103	0.411	0.368
BERT	*Random*	**0.615**	**0.561**	**0.593**	**0.692**	**0.599**	**0.598**	0.381	**0.615**	**0.698**	**0.625**
	Forward	0.366	0.382	0.226	0.457	0.328	0.380	**0.446**	0.306	0.110	0.489
	Reverse	0.367	0.430	0.256	0.455	0.292	0.428	0.371	0.210	0.662	0.483
Bi-GCN	*Random*	**0.838**	**0.785**	**0.841**	**0.886**	**0.785**	**0.854**	**0.745**	**0.861**	**0.939**	**0.847**
	Forward	0.415	0.509	0.386	0.311	0.319	0.489	0.551	0.381	0.401	0.511
	Reverse	0.498	0.584	0.339	0.786	0.118	0.517	0.502	0.413	0.667	0.419

Table 3: Experimental results of Twitter 15 & 16 datasets across three different data split strategies. Cells in **bold** indicate the best results from all models. Cells in gray indicate that the model trained using random splits achieves significantly better performance than using both forward and backward chronological splits. ($p < 0.05$, t-test).

Models	Splits	*Weibo 16*				*Weibo 20*				Sun-MM			
		Acc.	*P*	*R*	*F1*	*Acc.*	*P*	*R*	*F1*	*Acc.*	*P*	*R*	*F1*
Weak Baseline		0.493	0.493	0.492	0.493	0.501	0.501	0.501	0.501	0.514	0.512	0.514	0.512
SVM-HF	*Random*	**0.906**	**0.907**	**0.906**	**0.906**	**0.870**	**0.870**	**0.868**	**0.870**	**0.783**	**0.742**	**0.758**	**0.749**
	Forward	0.823	0.855	0.822	0.819	0.680	0.691	0.680	0.676	0.689	0.635	0.630	0.635
	Backward	0.752	0.757	0.752	0.752	0.801	0.802	0.801	0.801	0.771	0.740	0.676	0.692
BERT	*Random*	**0.918**	**0.918**	**0.917**	**0.918**	**0.920**	**0.921**	**0.920**	**0.920**	**0.839**	**0.807**	**0.806**	**0.806**
	Forward	0.889	0.892	0.888	0.888	0.738	0.756	0.738	0.732	0.708	0.682	0.708	0.680
	Backward	0.809	0.812	0.809	0.808	0.898	0.899	0.898	0.898	0.807	0.783	0.735	0.748
Bi-GCN	*Random*	**0.892**	**0.893**	**0.885**	**0.887**	-	-	-	-	-	-	-	-
	Forward	0.843	0.843	0.834	0.835	-	-	-	-	-	-	-	-
	Backward	0.762	0.783	0.762	0.747	-	-	-	-	-	-	-	-
H-Trans / Hybrid	*Random*	**0.955**	**0.956**	**0.955**	**0.955**	**0.959**	**0.960**	**0.959**	**0.959**	**0.853**	**0.818**	**0.829**	**0.823**
	Forward	0.946	0.949	0.946	0.946	0.850	0.860	0.849	0.850	0.707	0.687	0.725	0.685
	Backward	0.792	0.833	0.785	0.793	0.940	0.938	0.935	0.938	0.821	0.782	0.805	0.791

Table 4: Experimental results of Weibo 16 & 20 and Sun-MM across three different data split strategies. Cells in **bold** indicate the best results from all models. Cells in gray indicate that the model trained using random splits achieves significantly better performance than using both forward and backward chronological splits. ($p < 0.05$, t-test).

Discussion

Model Performance on Random Splits

The experimental results for all rumor detection approaches and data split strategies are shown in Tables 3 and 4. We can observe that training on random splits always leads to significant overestimation (t-test, $p < 0.01$) of model accuracy as compared to training on both forward and backward chronological splits.

Taking the best performing Bi-GCN model on Twitter 15 as an example, we observe a decrease in model accuracy of at least 39.4% when comparing test results on random splits against the two chronological splits. Furthermore, we find that some models (e.g., SVM-HF and Bi-GCN on Twitter 15) perform even worse than a weak baseline (e.g., the F1-measure results for the false rumor category (F) across two chronological splits in comparison with the weak baseline) that uses random predictions. As expected, our empirical findings align with previous studies of temporal impact in other downstream NLP tasks (Huang and Paul, 2019; Chalkidis and Søgaard, 2022; Mu et al., 2023b).

The results indicate that models learn to classify accurately rumor posts in the test set only when they are highly similar to posts in the training data, even though the remaining contextual information (such as user profile attributes, comments, and sometimes images) are different. To further investigate the impact of this semantic overlap, we conduct an ablation study (Section 5.3) and a similarity analysis (Section 5.4).

Discussion

Forward v.s. Backward Chronological Splits

Our experimental results show that models trained using backward chronological splits achieve higher accuracy on all datasets (except Weibo 16) as compared to those on forward chronological splits. This suggests that the models have the tendency to learn recurrent rumors. This observation is consistent across datasets. For instance, the accuracy of all models on the Twitter 16 dataset is higher when random splits are used for training as compared to forward splits, but lower when compared to backward splits. This may be attributed to similarities between the training and test sets. This is investigated further in Section 5.4.

How do we properly use static datasets?

we make the following practical suggestions for developing new rumor detection systems on static datasets:

For practical applications that aim to detect unseen rumors, it is essential to consider chronological splits when evaluating all rumor detection approaches on static datasets, in addition to standard random splits. By using forward and backward chronological splits, we can assess the ability of the rumor classifiers to handle both earlier and older unseen rumors.

How do we properly use static datasets?

Considering that temporalities (i.e., the temporal concentration of rumor topics) typically occur in widely used rumor detection datasets (e.g., Twitter 15&16 and Weibo 16 (Ma et al., 2016, 2017)), one can apply an additional data pre-processing measure to filter out rumor events with multiple posts. For instance, using out-of-the-box methods such as Levenshtein distance (Levenshtein et al., 1966) and BERTopic (Grootendorst, 2022), we identified a total of 9 similar rumors that resemble the false rumor depicted in Figure 1. After conducting a more in-depth error analysis on the predictions generated by the H-Trans model, which has demonstrated the highest predictive performance on Weibo 16, we discovered that the models can accurately classify all of these rumors in the test set when employing random data splits.

How do we properly use static datasets?

Current evaluation metrics, such as accuracy and F1-measure, are unable to accurately assess the true capability of rumor classifiers in detecting unseen rumors. Therefore, there is a need for new measures to evaluate the accuracy of model predictions for unknown rumors. For example, one can calculate the accuracy of a rumor detection system by excluding known rumors (i.e., similar rumors appearing in the training set) from the test set.

How do we properly use static datasets?

Given the limitations of the current pipeline that relies solely on static datasets, we argue that evaluation models should not be restricted to such datasets. By leveraging the consistent format of datasets collected from the same platform (as shown in Table 1), for example, one can explore broader temporalities by training a rumor classifier on Twitter 15 and evaluating its performance on Twitter 16. This protocol enables a more comprehensive examination of the generalizability of rumor detection systems, which is crucial for their practical applications in the real world (Moore and Rayson, 2018; Yin and Zubiaga, 2021; Kochkina et al., 2023).