Proceedings Paper:

Han, S., Gao, J. orcid.org/0000-0002-3610-8748 and Ciravegna, F. (2019) Neural language model based training data augmentation for weakly supervised early rumor detection. In: Proceedings of 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. 2019 IEEE/ACM International Conference on Social Networks Analysis and Mining (ASONAM 2019), 27-30 Aug 2019, Vancouver, Canada. ACM , pp. 105-112. ISBN 9781450368681

https://doi.org/10.1145/3341161.3342892

Neural Language Model Based Training Data Augmentation for Weakly Supervised Early Rumor Detection

Sooji Han
Department of Computer Science
The University of Sheffield
Sheffield, UK
sooji.han@sheffield.ac.uk

Jie Gao
Department of Computer Science
The University of Sheffield
Sheffield, UK
j.gao@sheffield.ac.uk

Fabio Ciravegna
Department of Computer Science
The University of Sheffield
Sheffield, UK
f.ciravegna@sheffield.ac.uk

***Abstract*—The scarcity and class imbalance of training data are known issues in current rumor detection tasks. We propose a straight-forward and general-purpose data augmentation technique which is beneficial to early rumor detection relying on event propagation patterns. The key idea is to exploit massive unlabeled event data sets on social media to augment limited labeled rumor source tweets. This work is based on rumor spreading patterns revealed by recent rumor studies and semantic relatedness between labeled and unlabeled data. A state-of-the-art neural language model (NLM) and large credibility-focused Twitter corpora are employed to learn context-sensitive representations of rumor tweets. Six different real-world events based on three publicly available rumor datasets are employed in our experiments to provide a comparative evaluation of the effectiveness of the method. The results show that our method can expand the size of an existing rumor data set by 200% with reasonable quality. Preliminary experiments with a state-of-the-art deep learning-based rumor detection model show that augmented data can alleviate over-fitting and class imbalance caused by limited train data and can help to train complex neural networks (NNs). With augmented data, the performance of rumor detection can be improved by 5.6%. Our experiments also indicate that augmented training data can potentially help to generalize rumor detection models on unseen new rumors.**

***Index Terms*—Data augmentation, weak supervision, rumor detection, social media**

I. Introduction

Research areas that have recently been received much attention in using Machine Learning (ML) and Natural Language Processing for automated rumor and fake news detection [1, 2, 3, 4, 5] and fact-checking [6, 7, 8]. In particular, deep learning architectures have been increasingly popular by providing significant improvements to state-of-the-art (SoA) performances. Despite their success, several challenges have yet to be tackled. One major bottleneck of state-of-the-art ML methods for rumor studies is that they require a vast amount of labeled data samples to be trained. However, the manual annotation of large-scale and noisy social media data for rumors is highly labor-intensive and time-consuming [9] as it requires deeper domain knowledge and a more elaborate examination than common annotations like image tagging or named entity annotations do. Due to limited labeled training data, existing NNs for rumor detection usually have shallow architecture [10, 11]. This restricts a further exploration of NNs for representation learning through many layers of nonlinear processing units and different levels of abstraction [12], which results in over-fitting and generalization concerns. The scarcity of labeled data is a major challenge facing for the research of rumors in social media [13, 14]. Another problem is that publicly available datasets for rumor-related tasks such as PHEME data [8] suffer from imbalanced class distributions [15, 8]. Existing methods for handling the class imbalance problem (e.g., oversampling and the use of synthetic data [16]) may cause over-fitting and poor generalization performance. A methodology for rumor data augmentation with the minimum of human supervision is necessary.

Data augmentation is the key to learning with modern deep neural networks (DNNs) as they require a large amount of data for training. The artificial augmentation of training data helps to alleviate data sparseness and class imbalance, reduce over-fitting, and reduce generalization error, thereby sustaining deeper networks and improving their performance. We argue that enriching existing labeled rumor data with duplicated tweets or corresponding variants is a promising attempt for early rumor detection methods [17] that rely on the structure of rumor propagation on social media. Recent findings [18, 10] show that rumors spread via the distribution of original sources. Original sources can quickly evolve into several new variants within the first few minutes in social media. Variations will gradually be increased with more information such as URLs (links) and photos by Twitter users. Links are usually created as new messages without attribution. Although new variations of rumors do not usually have any link or acknowledgement of their original sources, they can increase the credibility of sources with low credibility and the likelihood of rumor spreading. Malicious users leverage users' trust to spread rumors and harmful content on social media [19, 20]. According to previous studies on rumors on social media [18, 21], new variations of rumors posted within the first few peaks in event diffusion are mostly textual variants. 80% of a publicly available rumor tweet corpus consists of duplicated contents on average [10]. Previous studies revealed that variations of rumors share similar propagation patterns, and proposed methods for identifying rumors based on temporal, structural, and linguistic properties of their propagation [15, 2].

In this paper, we propose a novel data augmentation method for automatic rumor detection based on semantic relatedness. The method is based on a publicly available paraphrase identification corpus, context-sensitive embeddings of labeled reference tweets and unlabeled candidate source tweets. Pairwise similarity is used to guide the assignment of pseudo-labels to unlabeled tweets. ELMo [22], a state-of-the-art context-sensitive neural language model (NLM), is fine-tuned on a large credibility-focused social media corpus and used to encode tweets. Our results show that data augmentation can contribute to rumor detection via deep learning with increased training data size and a reasonable level of quality. This has potential for further performance improvements using deeper NNs. We present data augmentation results for six real-world events and the performance of a state-of-the-art DNN model for rumor detection with augmented data in Section VI. We will make the augmented data sets publicly available for further research purposes.

II. Related Work

Automatic data augmentation has been employed in a wide range of ML tasks as it helps to improve the generalization performance of deep learning models. Data augmentation usually makes use of transformations to which deep learning models invariant to. For example, common transformations for images include flipping, rotating, scaling, cropping, and adding noises. Our work focuses on data augmentation for textual data. The most common approach for augmenting textual data is to replace words or phrases with synonyms [23, 24, 25, 26, 27]. In one work on text classification [23], a WordNet thesaurus [28], in which synonyms for a word or phrase are grouped and ordered by semantic relatedness, is used to replace words in training corpora including reviews, news articles, and DBpedia data sets. The number of words to be replaced and an integer position in the index of synonyms of a given word are randomly determined from a geometric distribution with parameter $p = 0.5$. The authors present that augmented data improves the performance of convolutional neural networks (CNNs) for text classification. In particular, character-level CNNs trained on augmented data achieves the best performance. Recent research [25, 26] applies this method to tweets, and shows that data augmentation can bring performance gains in deep learning tasks on noisy and short social media texts. Vosoughi et al.[25] augment domain-independent English tweets for training an encoder-decoder embedding model built with character-level CNN and long short-term memory (LSTM) [25]. The number of tweets before data augmentation is not presented, but the author report that 3 million tweets in total are available after data augmentation. Another work [26] on tweet stance classification employs the same technique but uses Word2Vec [29] instead of the WordNet thesaurus [28] to replace words in text. Synonyms of a given word are ranked based on cosine similarity between the Word2Vec vector of given word and that of each synonym. The reported number of augmented tweets is 500,000. Despite a wide use of synonyms in text data augmentation and their contribution to performance enhancement, the use of paradigmatic relations can provide a wider range of substitutes for a given word [24]. To this end, Kobayashi [24] proposes methods for context-aware data augmentation based on a conditional bi-directional language model (BiLM). BiLM computes the probability distribution of possible substitutes for a given word in a sentence based on its context (i.e., a sequence of surrounding words). Their method is evaluated for text classification using six different data sets including movie reviews and answer types of questions. Contextual data augmentation makes marginal improvements over performances of synonym-based methods. Recently, a data augmentation method which combines $n-$grams and Latent Dirichlet Allocation (LDA) has been proposed [30]. The method is evaluated on its effectiveness in polarity classification (negative or positive) of reviews using CNNs. LDA is used to extract and rank keywords from positive and negative review corpora separately. Variations of a review are created by combining the original review with its trigrams that contain at least one keyword from the LDA review keywords of the same class type (i.e., positive or negative). Whereas most work on text data augmentation generates variations of a text based on the transformation of words and phrases, a recent work augments tweets by translating a tweet to a different language and then translating it back to the original language [31]. Unlike current artificial data augmentation methods based on modifications to existing data or reliance on limited knowledge bases, our method uses large-scale real-world social media data. It can not only increase the amount of training data, but most importantly help to increase the quality and diversity of original data.

III. Data

We use three publicly available rumor datasets covering a wide range of real-world events on social media, a Twitter paraphrase corpus, and two large-scale Twitter corpora.

PHEME data [8] This is an extension of the PHEME dataset of rumors and non-rumors and contains 9 manually labeled rumor event data set. This data is used as a reference data for data augmentation (see details in Section IV-A).

CrisisLexT26 [32] This data comprises tweets associated with 26 hazardous events happened between 2012 and 2013. A subset of data is manually labeled based on informativeness, information types, and information sources. This data is used as a reference data for data augmentation (see Section IV-A)

Twitter event datasets (2012-2016) [33] This data consists of over 147 million tweets associated with 30 real-world events unfolded between February 2012 and May 2016. We use this data as a pool of candidate source tweets. We choose six events out of 30 available events dataset, for which we can generate references corresponding to the candidate pool including 'Ferguson unrest', 'Sydney siege', 'Ottawa shootng', 'Charliehebdo', 'Germanwings plane crash', and 'Boston marathon bombings'. We refer to five events except the 'Boston marathon bombings' as **'PHEME5'** since the reference set can be generated from *"PHEME data"*. For the references of 'Boston bombings' event, we collect the reference set from *"CrisisLexT26"* and publicly identified rumor sources from fact-checking website 'Snopes.com' [1] since it is not available from *"PHEME data"* (see Section IV-B)

SemEval-2015 task 1 data [34] This data is built for paraphrase identification and semantic similarity measurement. It is employed in our semantic relatedness method to fine-tune a optimum relatedness threshold through a pairwise comparison between the embeddings of labeled reference tweets and those of unlabeled candidates event tweets (see Section V-A).

CREDBANK [35] This data comprises more than 80M tweets grouped into 1049 real-world events, each of which were manually annotated with credibility ratings. This large corpus is leveraged to fine-tune ELMo model in order to provide better representations for rumor-related tasks (refer to Section IV-C).

SNAP data [36] The SNAP Standford Twitter data set "twitter7" [2] is used as a general purpose Twitter corpus in our experiment. This is a collection of 476 million tweets collected between June-Dec 2009. We use this this to conduct comparative analysis of effectiveness of *CREDBANK* as a rumor task specific dataset for language model training. See Section IV-C for the details of a post-processed corpus.

IV. Methodology

A. Overview of the proposed method

An overview of data augmentation method is presented in Figure 1. Input corpus consist of *"References"* set and *"Candidates"* set. *"References"* are limited ground truth source tweets which are exploited to provide higher level supervision over unlabeled candidate tweet collections (i.e., *"Candidates"*). Candidate tweets refer to any tweets that report about an event of interest. Schemes for constructing references are varying between data sets. For *PHEME5*, we use annotations in the *PHEME* data. References from "Boston marathon bombings" event is generated separately (see IV-B). Specifically, a deep bidirectional language model (biLM) is firstly trained with domain-specific corpus in order to learn representations of rumors. We adopt the ELMo biLM model in this experiment. The leftmost box presents the dataset pre-processing and encoding method. Given corpora that contain pairs of reference and candidate tweets, we apply language-based filtering and perform linguistic pre-processing. The pre-processing includes lowercasing, removal of retweet symbols ('rt @'), URLs[3], and non-alphabetic characters, and tokenization. Tweets with a minimum of 4 tokens are considered to reduce noise [37]. Then, we compute contextual embeddings of tweets with fine-tuned biLM models (see section IV-D). The blue box illustrates our semantic relatedness based rumour variants identification method. Cosine distance between the embeddings of reference source tweets and those of unlabeled candidate tweets is used as a measurement of semantic similarity. Cosine similarity between vector representation of two sentences is a commonly used metric [38, 26, 39]. Two semantically equivalent embeddings have a cosine similarity of 1, and two vectors with no relation have that of 0. To determine whether a reference-candidate pair bears a strong semantic relations, *SemEval-2015 task 1 data* set, a standard short-text similarity benchmark dataset, is adopted to fine-tune relatedness thresholds. Two thresholds are learned from this fine-tuning process including rumor candidate threshold (θ_1) and non-rumor candidate threshold (θ_2). See Section V-A and V-B for the details of experiment and related class balancing strategy. Having an optimum threshold, we then perform semantic similarity computation for reference-candidate pairs (KxN) from the reference and candidate dataset. The next step is to select rumors and non-rumors from candidate tweets based on the optimum relatedness thresholds. In the final step, data collection [4] is performed to retrieve social-temporal context data (typically retweets and replies) for selected candidate tweets. Source tweets without context data are filtered out.

B. Reference Generation

We present how reference data is generated using already available labeled data. For the *PHEME5*, annotated rumor categories in the *PHEME data* are used. Rumor source tweets are categorized by their topics, and the authors create clean texts for each rumor category. For example, a rumor category for the Sydeny siege event,"The Sydney airspace has been closed", includes several rumor source tweets related to airspace over Sydney. Some examples are as follows: **(1)** "*CORRECTION: We reported earlier Sydney air space was shut down. That is not correct. No Sydney air space has been shut down. #SydneySiege*", and **(2)** "*DEVELOPING: Airspace shutdown over Sydney amid chocolate shop hostage situation; Islamic flag shown in shop's window.*" We understand that using raw tweets as references may help to capture more various patterns of rumor variations. However, tweets are very noisy and contain a large amount of non-standard spelling. To ensure high quality references and reduce the computation time of pairwise similarity between candidates and references, we use clean rumor categories as rumor references.

As the 'bostonbombings' event is not available in the *PHEME data*, we refer to *CrisisLexT26* as well as the Boston marathon bombings rumor archive created by Snopes.com. Any rumors investigated by Snopes.com are included in the reference set for 'bostonbombings' regardless of their veracity. In the *CrisisLexT26*, tweets are categorized by their *informativeness* (related to the crisis and informative, related but not informative, and unrelated), *information type* (affected

[1] A collection of rumors source tweets associated with the Boston Marathon bombing are available via https://www.snopes.com/fact-check/boston-marathon-bombing-rumors/, last access in April, 2019

[2] We downloaded the dataset from https://snap.stanford.edu/data/twitter7.html (accessed on March, 2019)

[3] Although embedded links can be considered as critical metadata about a potential rumour tweets citing a particular source for rumour detection, the external textual information are not exploited for our data augmentation work.

[4] We will make our source code and augmented data publicly available.

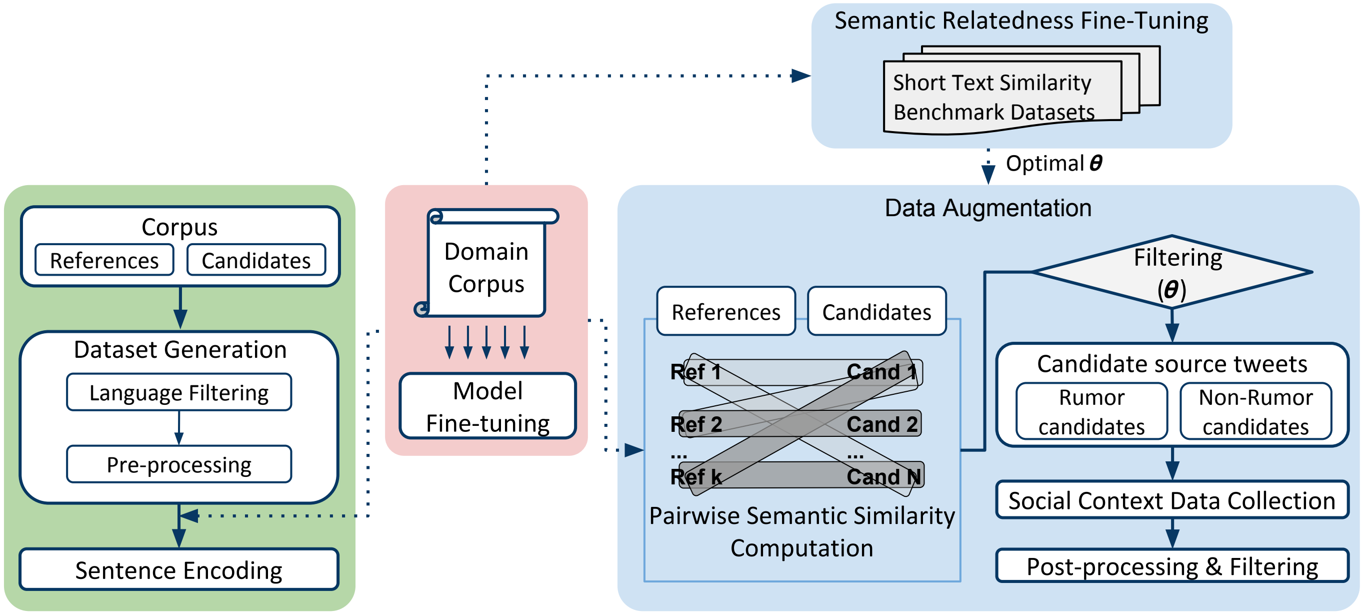

Fig. 1. Data augmentation architecture. The green box (i.e., the leftmost box) shows our method for tweet sentence pair encoding using fine-tuned language model. The blue box shows the key idea of the method employed to fine-tune relatedness thresholds for new source tweets variants identification and rumors dataset generation.

individuals, affected infrastructure, donations & volunteers, caution & advice, emotions, and other useful information), and *information sources* (e.g., eyewitness and media). The original data includes 1000 annotated tweets for the Boston marathon bombings. As the CrisisLexT26 is not annotated under an annotation scheme for social media rumors, we map its labels to binary labels (i.e., rumors/non-rumors).To this end, tweets with "related and informative" *informativeness* label are selected. Next, tweets, the *information type* of which is any of "affected individuals", "infrastructure and utilities", and "other useful information", are chosen. After sampling, 335 annotated tweets remain. We manually inspect and categorise them into rumors and non-rumors according the a rumor tweet annotation scheme proposed by Proter et al. [40]. To match the format of references generated using the *PHEME data*, we generate clean reference sentences from rumor tweets obtained after mapping the CrisisLexT26 labels and texts available in the Snopes.com's archive. Some examples of referecens for the 'bostonbombings' are as follows:

- The third explosion at the JFK library (unknown connection)
- Bombs were pressure cookers and placed in black duffel bags
- Suspect in Boston bombing described as dark skinned male

C. Data Collection

We download source tweets for six selected events in *Twitter events 2012-2016* and *CREDBANK* using an open source tweet collector called *Hydrator* [5]. Table I shows the number of tweet ids in the original *Twitter events 2012-2016* data, that of downloaded tweets, that of candidate source tweets which remained after language-based filtering and linguistic pre-processing (see Section IV-A), and that of references. For *CREDBANK*, *77,954,446* out of *80,277,783* tweets (i.e., 97.1% of the original data) are downloaded. After deduplication, the train corpus contains *6,157,180* tweets with *146,340,647* tokens and *2,235,075* vocabularies. We collect retweets using a Python library *tweepy* [6]w. Replies are collected via screen scraping technique implemented using Python libraries *Selenium* [7] and *BeautifulSoup* [8].

TABLE I
STATISTICS OF THE TWITTER EVENTS 2012-2016 DATA.

Event	# of tweets (original)	downloaded tweets	after preprocessing	# of references
germanwings	2,648,983	1,726,981	702,864	19
sydneysiege	2,157,879	1,376,218	1,211,295	61
fergusonunrest	8,782,071	5,743,959	5,504,692	41
ottawashooting	1,075,864	737,136	669,734	51
bostonbombings	3,430,387	1,886,632	1,259,857	88
charlihebdo	1,894,0619	12,253,734	4,276,112	60

D. Rumor-Oriented Embeddings (ELMo)

ELMo is adopted to learn effective representation of tweets. ELMo provides deep, contextualised, and characer-based word representations by using bidirectional language models (biLMs) [22]. Previous research shows that fine-tuning Neural Language Models (NLMs) with domain-specific data allows them to learn more meaningful word representations and provides a performance gain [41, 22]. To fine-tune pre-trained

[5] available via http://github.com/DocNow/hydrator

[6] available via https://www.tweepy.org/

[7] available via http://selenium-python.readthedocs.io/

[8] available via http://www.crummy.com/software/BeautifulSoup/bs4/doc/

TABLE II
STATISTICS OF TWO CORPUS FOR FINE-TUNING ELMO.

Corpus	Item	Train	Hold-out
CREDBANK	tweets	6,155,948	1,232
	tokens	146,313,349	27,298
	vocabs	2,234,861	6,517
SNAP	tweets	13,928,924	6,000
	tokens	193,192,322	99,758
	vocabs	11,696,602	24585

TABLE III
IMPROVEMENTS IN PERPLEXITY AFTER FINE-TUNING ON TWO CORPUS.

Data	Before tuning	After tuning (CREDBANK)	After tuning (SNAP)
Hold-out (CREDBANK)	883.06	18.24	360.47
Hold-out (SNAP)	476.42	N/A	64.92
Test	475.06	32.02	304.07

TABLE IV
COMPARISON OF THE PARAPHRASE IDENTIFICATION PERFORMANCE OF DIFFERENT MODELS FOR SENTENCE REPRESENTATION.

Model	F	P	R	Threshold
ELMo+CREDBANK (average)	**0.6507**	**0.6088**	0.6986	0.6526
ELMo+CREDBANK (top)	0.6270	0.5660	0.7027	0.6470
ELMo Original 5.5B (average)	0.6281	0.5872	0.6752	0.6305
ELMo Original 5.5B (top)	0.6047	0.5554	0.6635	0.6875
GloVe (twitter.27B.200d)	0.5079	0.3417	**0.9890**	0.5017
Word2Vec (Google News)	0.4223	0.4796	0.3772	0.5003
ELMo Original 5.5B (top)*	0.5868	0.5112	0.6887	0.6752
GloVe (twitter.27B.200d)*	0.5117	0.3565	0.9062	0.5070
Word2Vec (Google News)*	0.4715	0.4473	0.4985	0.5000

*Models are applied to normalized tweets.

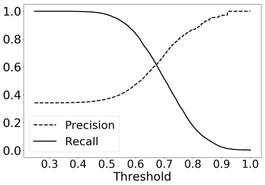

Fig. 2. Precision-recall curve

TABLE V
FINE-TUNING THRESHOLDS BY PRECISION.

P	F	R	THOLD
0.6088	0.6507	0.6986	0.6526
0.7000	0.6176	0.5526	0.6911
0.7500	0.5907	0.4871	0.7083
0.8502	0.4421	0.2987	0.7602
0.9003	0.2832	0.1681	**0.8018**

ELMo [9] for our task, we generate a dataset from *CREDBANK*. Sentences in original corpus are shuffled and split into training and hold-out sets. About 0.02% of the original data is used as the hold-out set. We also generate a test set using the PHEME data containing 6,162 tweets related to 9 events in the hope that it will offer an independent and robust evaluation of our hypothesis (refer to Section I). For SNAP ”Twitter-7” corpus, we use June tweets as training set to fine-tune pre-trained ELMo model and use a sample of 6000 tweets from November tweets as hold-out set. Table II shows the number of tweets, tokens and vocabularies in the training and hold-out sets of the CREDBANK and SNAP Twitter7 corpus after language filtering and deduplication. Following the practice in [42], a linear combination of the states of each LSTM layer and the token embeddings is adopted to encode tweets. Since the *CREDBANK* training set is still relatively small for NLMs, we only fine-tune the pre-trained ELMo with 1 epoch on two corpus respectively to avoid over-fitting. The model fine-tuned on Credbank dataest (refered as ”ELMo_Credbank”) was trained more than 800 hours on a Intel E5-2630-v3 CPU with maximum 50GiB RAM used. Model fine-tuned on SNAP corpus (refered as ”ELMo_SNAP”) was trained more than 500 hours on a NVIDIA Kepler K40M GPU. Table III shows a large improvement in perplexity on both hold-out set and test set with CREDBANK in comparison to the fine-tuned model with SNAP corpus. Reported values are the average of the forward and backward perplexity. Once fine-tuned, the biLM weights are fixed and used for computing the sentence representation of tweets in our experiments.

V. EXPERIMENTS

A. Semantic Relatedness Fine-Tuning

We are interested in exploring the effect of the distance between embeddings of pairs of reference and candidate tweets on the quality of augmented data which will eventually affect the rumor detection model’s capability to predict unseen rumors. Table IV compares different models for word representation on the SemEval-2015 data. We show the results based on the maximum F-score each model achieved. Our experimental results show the effectiveness of our CREDBANK fine-tuned ELMo over pre-trained model (”Original (5.5B)”) and SoA word embedding models. We applied different models to normalized texts. Normalization methods we used in the experiments include removing English stopwords and punctuations, and lemmatization using ‘WordNetLemmatizer’ in a Python library NLTK [10]. As shown in Table IV, text normalization actually degenerates the performance of the ELMo in terms of F-score, while it improves the performance of the other word embeddings. In fact, state-of-the-art NLMs like ELMo do not need much text normalization. A pre-trained ELMo model only needs tokenization. As for the output of ELMo models, using the average of representations from all layers outperforms using only the top layer representation. This finding is consistent with results presented in Perone et al.’s work [42]. To ensure higher quality (i.e., less false positives in a selected sample), we argue that a higher precision is required [26]. Therefore, relatedness thresholds are fine-tuned based on precision achieved by the best-performing model. Table V shows a part of fine-tuning results. We should choose a threshold which can achieve a reasonably high precision and sample an adequate number of tweets.

B. Data Augmentation

We follow our data augmentation procedure described in Section IV. After pairwise similarity computation on all references and candidates, we apply relatedness thresholds to the

[9] The pre-trained model and the Tensorflow training checkpoints are obtained from Tensorflow implementation of ELMo, available via github.com/allenai/bilm-tf

[10] available via https://www.nltk.org

results for selecting rumor and non-rumor source tweets from a pool of candidates. For sampling rumor sources, we use $\theta_1 = 0.8018$, which achieves a precision of 0.9 in the benchmark task illustrated above. If a semantic similarity score between a candidate and one or more references is greater than or equal to θ_1, the candidate is included in a rumor source collection. If a candidate is identified as a rumor for any of rumor references, this candidate is included in a rumor. For non-rumor sources, we assume that low semantic relatedness to rumor references indicate the high likelihood of being a non-rumor. The minimum semantic similarity score for positive paraphrase pairs in the SemEval-2015 task is 0.248. We set a threshold (θ_2) for sampling non-rumor samples to 0.266, which is the second smallest semantic similarity score for the SemEval-2015 task and achieves the same precision, recall, and F-measure as the minimum score 0.248. If a semantic similarity between a candidate and every rumor reference is less than (θ_2), the candidate is included in a non-rumor source collection. Data augmentation results after applying thresholds show high class imbalance for all event except the 'germanwings'. To overcome this problem, random sampling is applied to the non-rumor source collection. Specifically, we randomly sample $(3 * (\texttt{number of augmented rumor sources}))$ non-rumors from the collection. Given augmented and initially balanced rumor and non-rumor source tweets, replies for each source tweet are collected (see Section IV-C) and source tweets without replies are removed from the augmented data. We observe a considerable reduction in augmented data size because a large number of source tweets do not have replies. Next, we apply sampling again. $(2 * (\texttt{number of rumor source tweets}))$ non-rumor source tweets are randomly sampled to balance class distributions in each event data set. In order to keep source tweets which are rich in conversational threads, we include all source tweets that have more than 10 replies. The remainder is randomly chosen. Finally, augmented rumor and non-rumor source tweets with replies are merged with the *PHEME5*.

C. Rumor Detection

We conduct rumor detection experiments using the original PHEME5 and two augmented data sets: *PHEME5+Aug* and *PHEME5+Aug+boston*. "*PHEME5+Aug*" is augmented data for the five events in the PHEME5. "*PHEME5+Aug+boston*" is "*PHEME5+Aug*" combined with the 'bostonbombings'. We employ Kochkina et al.'s method as a SoA baseline model of rumor detection with slight modifications [8]. In their model, source tweets and replies are represented as 300-dimensional word2vec word embeddings pre-trained on the Google News data set [11]. For the sake of simplicity, we modify the implementation of MTL2 Veracity+Detection [12] for rumor detection only. Another modification we made is data input. In the original models, a conversation consists of a source tweet and replies to it and conversations are decomposed into branches. In our experiments, we are unable to obtain the conversation structure and decompose it into several branches with our augmented datasets. For example, if tweet B is a reply of a source tweet A and tweet C is a reply of B, Twitter objects represent that C is a reply of A. To overcome this limitation but still take contexts into consideration, we consider the entire conversation of a source tweets as a single branch. We construct input by using source tweet and the top (i.e., most recent) 24 replies of each source tweet in this task. The original models require input with shape: (the number of branches in each event dataset, the maximum length of branches, 300). Therefore, the modified models require input with shape: (the number of source tweets in each event data, 25, 300). As for hyperparameter optimization, we implement a grid search with the parameter space defined by Kochkina et al. Parameter combinations are optimized based on accuracy on the validation set over 20 trials. For the *PHEME5* and *PHEME5+Aug*, we use 'fergusonunrest', 'ottawashooting', and 'sydneysiege' as training data, 'charliehebdo' as a validation set as proposed in the original implementation. For *PHEME5+Aug+boston*, we use 'bostonbombings' as a training set on top of the three events. For evaluation, leave-one-out cross-validation (LOOCV) is performed, which means that one event is used as a test set and the remaining events are used as a training set on each iteration. This setup makes it possible to evaluate rumor detection in real-world scenarios in which detection models are required to identify unseen rumors [8].

VI. Results and Discussion

A. Data Augmentation

We augment rumor and non-rumor source tweets for the six selected events in the Twitter events 2012-2016 data. Then, the augmented tweets for the PHEME5 events are merged with the original PHEME5. Table VI shows the number of source tweets and replies obtained via our data augmentation method and those after balancing augmented data and merging the balanced data with the original PHEME5. The values in the parentheses are the number of tweets in the original PHEME5. Overall, the number of source tweets for rumors and non-rumors increased by 216% and 192%, respectively. There are 52% and 149% increases in the number of replies for rumor sources and that for non-rumor sources, respectively. The standard deviation of imbalance ratios of non-rumor sources to rumor source improved from 1.24% to 0.61%, respectively. In particular, significant class imbalances in two largest events–'fergusonunrest' and 'charliehebdo'–have become moderate as a result of data augmentation.

Manual inspection of sampled source tweets shows that augmented data contains tweets identical to references and several variations of references. It is worth noting that our data augmentation with weak supervision can even capture rumors which are related but not technically identical to reference tweets. Some examples of rumor tweets in our augmented data are as follows:

(1) ***A 20-year-old student*** *is among the hostages at the kosher shop in Paris http://t.co/orBfH8MK1J*: This tweet is almost

[11] https://code.google.com/archive/p/word2vec/

[12] available via http://github.com/kochkinaelena/Multitask4Veracity

identical to a reference tweet, “**A baby** is among the hostages in the Kosher market”, for the Charlie Hebdo attack, except for subjects of sentences. The semantic similarity score between two sentences is 0.8123.
(2) *Uber Promises **Free Rides** in Sydney after Surge Pricing Kicks in During Hostage Crisis http://t.co/7NAO9HSxEA*: This tweet is a variation of a reference tweet, “Uber introduced **surge pricing** in downtown Sydney during hostage crisis.”. Two sentences report contradictory sub-events related to a taxi booking company called Uber, but their semantic similarity score is 0.8238.
Using raw annotated tweets as references rather than refined categories of rumors may help to retrieve more positive examples. In the original PHEME, for example, a tweet, “Ray Hadley says he spoke with hostage, and could hear the gunman in the background barking orders and demanding to go live on air”, is annotated as a rumor, “The gunman and/or hostages have made contact with Sydney media outlet(s) (radio station, etc.)”. Without a background knowledge that Ray Hadley is an Australian radio broadcaster, data augmentation methods based on semantic relatedness fail to identify such rumors.

B. Rumor Detection

We conduct rumor detection experiments on three data sets: (1) *PHEME5*, (2) *PHEME5+Aug*, (3) *PHEME5+Aug+boston*. We employ Kochkina et al.’s method as state-of-the-art baseline model of rumor detection with slight modifications [8] (refer to Section V-C for details). The main results of our rumor detection experiments are presented in Table VII. It shows that data augmentation helps to boost performance on rumor detection in terms of F-score, precision, and accuracy. On the other hand, recall decreases when using augmented data. Such results indicate that a rumor detection model provides substantially more rumors than non-rumors with augmented data. Although a difference is marginal, *PHEME5+Aug* is more effective than *PHEME5+Aug+boston*. Table VIII shows LOOCV results described in Section V-C. ‘Event’ column in Table VIII is the event used as the test set on each iteration of cross-validation. Overall, augmented data is helpful to improve precision for all the events. The ’fergusonunrest’ is the most difficult event in the *PHEME5* for a rumor detection model as it has a unique class distribution distinguished from all other events [8]. Our data augmentation alleviates class imbalance and improves the F-score of rumor detection on the ’fergusonunrest’ by 10.5%. The cross-validation results on the ‘bostonbombings’ also show a high imbalance between precision and recall, which results in low F-score compare to all the other events.

TABLE VII
RUMOR DETECTION RESULTS FOR DIFFERENT DATASETS.

Data	F	P	R	Acc.
PHEME5	0.5285	0.5466	**0.5117**	0.6898
PHEME5+Aug	**0.5846**	**0.7776**	0.4683	**0.7779**
PHEME5+Aug+boston	0.5701	0.7335	0.4662	0.7653

TABLE VIII
LOOCV RESULTS FOR THE PHEME5 AND AUGMENTED DATA SETS.

Event	Data	F	P	R	Acc.
germanwings	PHEME5	**0.519**	0.658	**0.429**	0.597
	PHEME5+Aug	0.369	**0.855**	0.235	0.635
	PHEME5+Aug+boston	0.498	0.795	0.363	**0.668**
sydneysiege	PHEME5	**0.571**	0.671	**0.496**	0.681
	PHEME5+Aug	0.491	**0.881**	0.341	0.752
	PHEME5+Aug+boston	0.538	0.814	0.402	**0.757**
fergusonunrest	PHEME5	0.217	0.436	0.144	**0.741**
	PHEME5+Aug	0.258	**0.636**	0.161	0.736
	PHEME5+Aug+boston	**0.322**	0.574	**0.224**	0.732
ottawashooting	PHEME5	0.631	0.731	0.555	0.657
	PHEME5+Aug	**0.769**	**0.776**	**0.762**	**0.823**
	PHEME5+Aug+boston	0.659	0.745	0.591	0.763
charliehebdo	PHEME5	0.527	0.405	**0.756**	0.702
	PHEME5+Aug	0.691	**0.726**	0.658	0.837
	PHEME5+Aug+boston	**0.722**	0.693	0.754	**0.840**
bostonbombings	PHEME5+Aug+boston	0.158	0.731	0.089	0.685

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a new paradigm of data augmentation for effectively enlarging existing rumor data sets using publicly available large-scale unlabeled data for real-world events on social media. Semantic relatedness is exploited to apply weak labels to unlabeled data based on limited labeled rumor source tweets. Our experiments show the potential efficiency and effectiveness of semantically augmented data for combating the scarcity of labeled data and class imbalance of existing publicly available rumor data sets. Our augmented data is highly realistic and can potentially increase the diversity of existing labeled data and improve its quality. Preliminary results achieved using a SoA DNN model indicate that augmented data is helpful to train deep neural networks. We release this augmented data in the hope that it will be useful for further research in the field of rumor detection and general studies of rumor propagation on social networks. In the future, we plan to extend our method to other rumor event data sets and training tasks in order to build more comprehensive data for rumor detection. A more extensive evaluation will be conducted to examine the effectiveness of augmented data in handling over-fitting and its usefulness in facilitating deeper NNs for rumor detection. Further research will also look into more advanced techniques for rumor variation identification. In addition, it is arguable that different types of rumor events may expose different propagation patterns. We will look into whether data augmentation create a bias towards detecting the same sort of rumors. Increasing diversity and reducing bias in training data will be a future direction of our work.

REFERENCES

[1] S. Helmstetter and H. Paulheim, “Weakly supervised learning for fake news detection on twitter,” in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, vol. 00, Aug. 2018, pp. 274–277. [Online]. Available: doi.ieeecomputersociety.org/10.1109/ASONAM.2018.8508520

[2] S. Kwon, M. Cha, and K. Jung, "Rumor Detection over Varying Time Windows," *PLOS ONE*, vol. 12, no. 1, pp. 1–19, 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0168344

[3] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[4] K. Shu, S. Wang, and H. Liu, "Understanding user profiles on social media for fake news detection," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2018, pp. 430–435.

[5] K.-F. Wong, W. Gao, and J. Ma, "Rumor detection on twitter with tree-structured recursive neural networks," in *ACL*, 2018.

[6] C. Boididou, S. E. Middleton, Z. Jin, S. Papadopoulos, D.-T. Dang-Nguyen, G. Boato, and Y. Kompatsiaris, "Verifying information with multimedia content on twitter," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 15 545–15 571, 2018.

[7] S. Vosoughi, M. Mohsenvand, and D. Roy, "Rumor gauge: Predicting the veracity of rumors on twitter," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 11, no. 4, p. 50, 2017.

[8] E. Kochkina, M. Liakata, and A. Zubiaga, "All-in-one: Multi-task learning for rumour verification," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3402–3413.

[9] A. Zubiaga, M. Liakata, and R. Procter, "Learning reporting dynamics during breaking news for rumour detection in social media," *CoRR*, vol. abs/1610.07363, 2016.

[10] T. Chen, X. Li, H. Yin, and J. Zhang, "Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 40–52.

[11] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, pp. 3818–3824. [Online]. Available: http://dl.acm.org/citation.cfm?id=3061053.3061153

[12] G. Zhong, L.-N. Wang, X. Ling, and J. Dong, "An overview on data representation learning: From traditional feature learning to recent deep learning," *The Journal of Finance and Data Science*, vol. 2, no. 4, pp. 265–278, 2016.

[13] A. Aker, A. Zubiaga, K. Bontcheva, A. Kolliakou, R. Procter, and M. Liakata, "Stance classification in out-of-domain rumours: A case study around mental health disorders," in *International Conference on Social Informatics*. Springer, 2017, pp. 53–64.

[14] S. Vosoughi, "Automatic detection and verification of rumors on twitter," Ph.D. dissertation, Massachusetts Institute of Technology, 2015.

[15] Y. Liu, X. Jin, H. Shen, and X. Cheng, "Do rumors diffuse differently from non-rumors? a systematically empirical analysis in sina weibo for rumor identification," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 407–420.

[16] W. Xu and H. Chen, "Scalable rumor source detection under independent cascade model in online social networks," in *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*. IEEE, 2015, pp. 236–242.

[17] A. Zubiaga, M. Liakata, and R. Procter, "Exploiting context for rumour detection in social media," in *International Conference on Social Informatics*. Springer, 2017, pp. 109–123.

[18] J. Maddock, K. Starbird, H. J. Al-Hassani, D. E. Sandoval, M. Orand, and R. M. Mason, "Characterizing online rumoring behavior using multi-dimensional signatures," in *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. ACM, 2015, pp. 228–241.

[19] C. Baxter, P. Barratt, and M. Thomson, "Social media and the generation, propagation, and debunking of rumours," *Report on behalf of Department of National Defence, Canada. Ontario: Human Systems Incorporated*, 2015.

[20] A. Arif, K. Shanahan, F.-J. Chou, Y. Dosouto, K. Starbird, and E. S. Spiro, "How information snowballs: Exploring the role of exposure in online rumor propagation," in *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 2016, pp. 466–477.

[21] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. International World Wide Web Conferences Steering Committee, 2015, pp. 1395–1405. [Online]. Available: https://doi.org/10.1145/2736277.2741637

[22] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[23] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[24] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 452–457. [Online]. Available: https://www.aclweb.org/anthology/N18-2072

[25] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2vec: Learning tweet embeddings using character-level cnn-

lstm encoder-decoder," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 1041–1044.

[26] P. Vijayaraghavan, I. Sysoev, S. Vosoughi, and D. Roy, "Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns," *arXiv preprint arXiv:1606.05694*, 2016.

[27] O. Kolomiyets, S. Bethard, and M.-F. Moens, "Model-portability experiments for textual temporal analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, 2011, pp. 271–276.

[28] G. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[30] M. Abulaish and K. Sah, Amit, "A text data augmentation approach for improving the performance of cnn," in *Proceedings of the MINDS Workshop,the 11th International Conference on Communication Systems and Networks (COMSNETS)*, Banglore, India, 2019, pp. 1–6.

[31] F. M. Luque and J. M. Pérez, "Atalaya at tass 2018: Sentiment analysis with tweet embeddings and data augmentation," *Proceedings of TASS*, vol. 2172, 2018.

[32] A. Olteanu, S. Vieweg, and C. Castillo, "What to expect when the unexpected happens: Social media communications across crises," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 2015, pp. 994–1009.

[33] A. Zubiaga, "A longitudinal assessment of the persistence of twitter datasets," *JASIST*, vol. 69, pp. 974–984, 2018.

[34] W. Xu, A. Ritter, C. Callison-Burch, W. B. Dolan, and Y. Ji, "Extracting lexically divergent paraphrases from Twitter," *Transactions of the Association for Computational Linguistics*, 2014. [Online]. Available: http://www.cis.upenn.edu/~xwe/files/tacl2014-extracting-paraphrases-from-twitter.pdf

[35] T. Mitra and E. Gilbert, "Credbank: A large-scale social media corpus with associated credibility annotations," in *Ninth International AAAI Conference on Web and Social Media*, 2015.

[36] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 177–186.

[37] G. Ifrim, B. Shi, and I. Brigadir, "Event detection in twitter using aggressive filtering and hierarchical tweet clustering," in *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*. ACM, 2014.

[38] X. Lu, B. Zheng, A. Velivelli, and C. Zhai, "Enhancing text categorization with semantic-enriched representation and training data augmentation," *Journal of the American Medical Informatics Association*, vol. 13, no. 5, pp. 526–535, 2006.

[39] J. Shin, L. Jian, K. Driscoll, and F. Bar, "The diffusion of misinformation on social media: Temporal pattern, message, and source," *Computers in Human Behavior*, vol. 83, pp. 278 – 287, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0747563218300669

[40] R. Procter, F. Vis, and A. Voss, "Reading the riots on twitter: methodological innovation for the analysis of big data," *International journal of social research methodology*, vol. 16, no. 3, pp. 197–214, 2013.

[41] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[42] C. S. Perone, R. Silveira, and T. S. Paula, "Evaluation of sentence embeddings in downstream and linguistic probing tasks," *arXiv preprint arXiv:1806.06259*, 2018.

TABLE VI
NUMBER OF RUMOR AND NON-RUMOR SOURCE TWEETS AND REPLIES IN THE AUGMENTED DATA.

	Augmented data				**After balancing and merging**			
	Rumor		**Non-rumor**		**Rumor**		**Non-rumor**	
Event	**source**	**threads**	**source**	**threads**	**source**	**threads**	**source**	**threads**
germanwings	272	710	373	1,642	502 (238)	2,913 (2,256)	604 (231)	3,406 (1,764)
sydneysiege	1,289	4,432	3,955	14,935	1,766 (522)	12,216 (8,155)	3,248 (699)	27,173 (14,621)
ottawashooting	625	1,817	3,607	14,939	1,047 (470)	7,349 (5,966)	1,648 (420)	16,158 (5,428)
ferguson	375	4,133	2,992	22,810	638 (284)	10,096 (6,196)	1,609 (859)	35,057 (16,837)
charliehebdo	802	2,020	4,437	26,425	1,225 (458)	8,599 (6,888)	3,213 (1,621)	49,888 (29,302)
bostonbombings	429	3,483	3,231	44,198	429 (N/A)	3,483 (N/A)	858 (N/A)	37,692 (N/A)
Total	3,792	16,595	18,595	124,949	5,607 (1,772)	44,656 (29,461)	11,180 (3,830)	169,374 (67,952)