# Journalist-in-the-Loop: Continuous Learning as a Service for Rumour Analysis

**Twin Karmakharm** **Nikolaos Aletras** **Kalina Bontcheva**
Department of Computer Science
University of Sheffield, UK
[t.karmakharm, n.aletras, k.bontcheva]@sheffield.ac.uk

## Abstract

Automatically identifying rumours in social media and assessing their veracity is an important task with downstream applications in journalism. A significant challenge is how to keep rumour analysis tools up-to-date as new information becomes available for particular rumours that spread in a social network.

This paper presents a novel open-source web-based rumour analysis tool that can continuous learn from journalists. The system features a rumour annotation service that allows journalists to easily provide feedback for a given social media post through a web-based interface. The feedback allows the system to improve an underlying state-of-the-art neural network-based rumour classification model. The system can be easily integrated as a service into existing tools and platforms used by journalists using a REST API.

## 1 Introduction

Identifying rumours and assessing their veracity in social media is an important task with downstream applications in journalism (Zubiaga et al., 2018a). Such tools can be used to make journalists more productive and also have the potential to be valuable tools in informing the public about the veracity of rumours especially during political crises and pre-electoral periods (Tsakalidis et al., 2018).

Individual and collaborative manual approaches to rumour analysis do not scale due to the large volume and velocity of user generated content (Konstantinovskiy et al., 2018). On the other hand, automatic machine learning-based approaches are often falling short on accuracy, when presented with previously unseen rumours (Zubiaga et al., 2018a).

Current rumour analysis practices also tend to entail journalists making decisions using a disparate set of tools, such as Google reverse image search, Tweetdeck, or more experimental machine learning-based rumour or video analysis algorithms. Even the latest research projects in these areas, e.g. PHEME (Derczynski and Bontcheva, 2014), InVID (Teyssou et al., 2017), Hoaxy (Shao et al., 2016), envisage the computer algorithms as intelligence augmentation tools for the journalists, used to scale up their abilities to deal with a large volume, velocity, and variety of social content with uncertain veracity.

Even though large amounts of new human insights and evidence accumulate over time, journalists cannot use this to improve their tools since the machine learning models behind them are not updatable. This is a major limitation on the practical usefulness of these tools, since our latest research (Lukasik et al., 2015) on rumour stance and veracity classification, for example, demonstrated that giving the machine learning models as few as just ten human-labelled examples from a newly emerging rumour can improve the algorithm accuracy by at least 10%.

Therefore, not only journalists can benefit from using machine intelligence to improve their productivity, but also the underlying algorithms can get smarter, if only journalists could feed back the new evidence in a way that enables the tools to learn from such new data.

This paper presents a new open source web service for rumour analysis[1] that can improve over time by using feedback from journalists. The main user requirement is for journalists to invest minimal time and effort providing the additional manual feedback which in turn will yield productivity gains from the more accurate machine learning models. From a technological perspective, the assumption being tested is that continuous learning and human which computation techniques can

[1] https://tweetveracity.gate.ac.uk



*Proceedings of the 2019 EMNLP and the 9th IJCNLP (System Demonstrations)*, pages 115–120
Hong Kong, China, November 3 – 7, 2019. 

be used to underpin an easy to use misinformation analysis interface for journalists. Due to this being a prototype project, we focus initially on rumour analysis and only on Twitter.

To test this hypothesis, we have built a prototype web service that brings together state-of-the-art machine learning-based algorithms for rumour detection (Aker et al., 2019). Journalists are able to access the service via a web-based application interface from their desktop or mobile devices and monitor and verify emerging rumours in social media. The web application uses the machine learning algorithms, in order to quickly gather and present journalists with evidence around the narratives being monitored, e.g. how fast is the rumour spreading and which are the users who are most affected; who are the key proponents; is the rumour attracting polarised opinions and discussions; how likely is the rumour to be true. Our main aim is to support much more complex types of analysis, than just focusing on the veracity of an individual piece of information, e.g. has an image or a video been tampered with.

## 2 Related Work

In 2016 alone, the Duke Reporters Lab reported a 50% increase in global fact-checking by media and independent fact-checking organisations.[2] Journalists and news editors currently use largely manual processes for analysis, investigation, and verification of social media and other online content. A limited number of production quality tools are currently available to support them in the individual steps of this process, with much of the technology still in research and experimental phases.

In more detail, existing projects and tools are mostly focused on images/video forensics and verification (e.g. InVID (Teyssou et al., 2017), REVEAL[3]), crowdsourced verification (e.g. CheckDesk[4], Veri.ly[5]), fact-checking claims made by politicians (e.g. Politifact[6], FactCheck.org[7], FullFact[8]), citizen journalism (e.g. Citizen Desk), repositories of checked facts/rumours/websites (e.g. Emergent (Ferreira and Vlachos, 2016), FactCheck [7] , Decodex[9]), or pre-trained machine learning models and tools, which however cannot be adapted by the journalists to new data (e.g. PHEME (Derczynski and Bontcheva, 2014), REVEAL[3]).

There are also existing tools for visualising and analysing online rumours which are related to the user interface of our system:

- RumorLens (Resnick et al., 2014) is a prototype aimed at citizens and journalists, to help detect rumours early, then classify posts as spreading or correcting the given rumour, and also visualising its spread. A human-in-the-loop learning showed good results on the tweet retrieval task. This motivated us to propose extending this approach to other rumour and misinformation analysis tasks.

- TwitterTrails (Metaxas et al., 2015) is an interactive, web-based tool that visualises the origin and propagation characteristics of a rumour and its refutation, on Twitter. Another visualisation-based framework for studying rumour propagation is RumourFlow (Dang et al., 2016).

- Hoaxy (Shao et al., 2016) is a recent open-source tool focused on visualising and searching over claims and fact checks. Such sophisticated visualisations are out of scope of our system, but relevant open-source visualisation tools, e.g. from Hoaxy, could be integrated in the future.

- CrossCheck[10] was a collaborative rumour checking project led by First Draft and Google News Lab, during the French elections. Its output was a useful dataset of false or unverified rumours.

- Meedan's Check[4] is an open-source breaking news verification platform, which however does not support continuously updated machine learning methods.

- ClaimBuster (Hassan et al., 2017) is a tool which gathered volunteer and expert-based claim annotations to train machine learning methods for claim classification (factual vs

[2] https://reporterslab.org/global-fact-checking-up-50-percent
[3] https://revealproject.eu/
[4] https://meedan.com/en/check/
[5] https://veri.ly
[6] https://www.politifact.com/
[7] https://www.factcheck.org/
[8] https://fullfact.org/
[9] https://www.lemonde.fr/verification/
[10] https://crosscheck.firstdraftnews.org/france-en/

non-factual). In contrast, we propose a service where rumour annotation is carried out as a side effect of the journalist workflow, as well as having a wider range of machine learning methods, for different rumour analysis tasks.

However, to the best of our knowledge, the above tools do not consider using feedback provided by journalists to continuously update their underlying rumour classification models.

## 3 Journalist-in-the-Loop System Overview

The system is an integration of state-of-the-art machine learning algorithms for detecting emerging rumours; analysing the online narratives around them for stance and temporal evolution; and automatic veracity classification. The starting point are our open source algorithms from the PHEME project (Derczynski and Bontcheva, 2014) adapted to fit the learning paradigm, so the models evolve as more journalist-labelled data comes in.

The core of the rumour analysis service consists of three main parts: (1) a rumour classification system; (2) a rumour annotation service; and (3) a database which stores the training data required by the classification system and allows the system to be updated continuously using the newly annotated rumours. The diagram in Fig. 1 provides an overview of the system.

## 4 Rumour Classification System

The rumour classification system contains and manages the model that is used for classifying new unseen rumours. The system is built on top of the PyTorch[11] deep learning framework and consists of three components.

First, the data processing component is used to transform text into a representation suitable for the Rumour Classification Model where its inputs and outputs are described in Section 4.1. If there are many annotations provided by the user for the same piece of rumour, the system chooses the most frequent one. The processed text is then transformed into a set of matrices that can be directly used in the model training process.

Second, the model training component trains and validates the model using the available annotated rumours dataset prepared by the data processing component. The dataset is randomised and split in to training (80%) and testing (20%) sets. In this case, there is no validation dataset as there is currently no further tuning of hyperparameters, hence more data is dedicated for training the model instead. Once the training is complete, the model's parameters are then saved to be used in the next stage.

Finally, the prediction component offers an interface for making predictions on new unseen rumours which is used directly by the Rumour Annotation Service as described in Section 5. The component uses the stored model's parameters from the last stage.

### 4.1 Rumour Classification Model

The model used in the Rumour Classification System is based on the state-of-the-art rumour veracity classification algorithm of Aker et al. (2019). It is a recurrent network which classifiers Twitter rumours into three categories, true, false or unverified. A diagram of the model can be seen in Fig. 2.

The model takes three inputs. First input is the source tweets, the text is cleaned and tokenized before being fed into the network. The second input are the recurring terms that occur frequently and which are associated to each veracity category (e.g. the word 'live' has strong association with true rumours). In this case, only the recurring terms that are associated with false rumours are used as they are shown to give the best result. The third input is the stance related to the tweet obtained from the comments associated with the tweet. The stance are proportions of associated tweets which either support the original tweet, denies it or are neutral (e.g. further questions or simply making comments without supporting or denying).

The tokenized source tweets and recurring terms are embedded using the pre-trained Google news word2vec model (Mikolov et al., 2013). The embedded source tweets and recurring terms are then fed into an attention layer that uses the recurring terms to weight the importance of the source tweet words. The output of the attention layer is fed into a Long-Short Term Memory (LSTM) layer (Hochreiter and Schmidhuber, 1997) that generates a single 10-feature vector. The LSTMs output feature vector is concatenated with the stance input and fed in to a dense layer

[11] https://pytorch.org

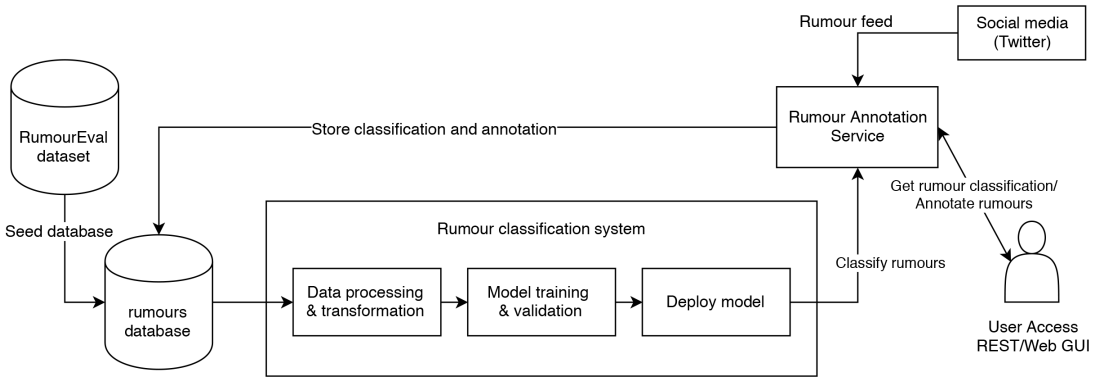


Figure 1: Data flow diagram of the rumour classification service.

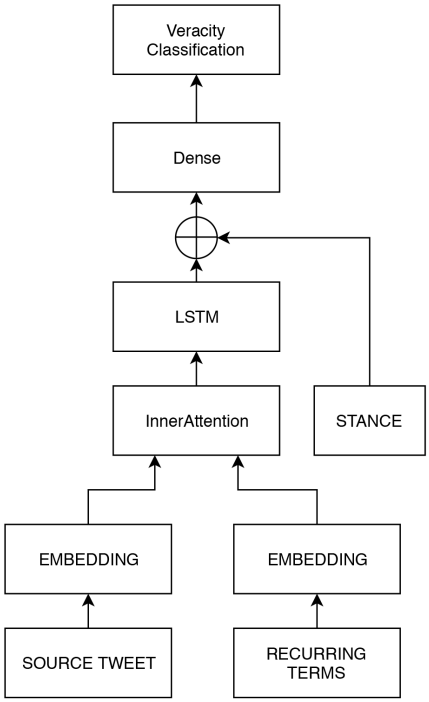


Figure 2: Network diagram of the rumour veracity model.

for a three-category classification output.

### 4.2 Data and Initial Results

The initial data source used for seeding the rumour veracity classification model is the RumourEval2017 dataset (Derczynski et al., 2017) which is derived from the PHEME dataset (Zubiaga et al., 2018b). The RumourEval2017 dataset contains 325 Twitter conversation threads discussing rumours with respect to eight different man-made events like Germanwings Air Crash, Charlie Hebdo, Ottowa Shootings, etc. Each thread in the dataset is annotated as true, false or unverified. Also each reply to a source tweet is annotated with one of the labels: supporting, denying, questioning and commenting. It is split into training, development and test set as in the RumourEval2017 challenge with 272, 25 and 28 rumours respectively. The dataset has a majority class baseline of 0.429.

Evaluation is performed by comparing against several state-of-the-art approaches – NileTMRG (Enayet and El-Beltagy, 2017), Branch LSTM (Zubiaga et al., 2018b), Multi-task Learning (Kochkina et al., 2018), vanilla LSTM (without inner-attention), LSTM with soft attention (Bahdanau et al., 2014) and our Inner-Attention algorithm. The algorithms achieved F-1 scores of 0.539, 0.558, 0.491, 0.528, 0.496, 0.616 and accuracy of 0.571, 0.571, 0.5, 0.537, 0.5, and 0.607 respectively. We expect the results should improve as more data is added to the training set through the Rumour Annotation Service. Further analysis and discussion of the algorithm and its evaluation can be found in Aker et al. (2019).

## 5 Rumour Annotation Service

The Rumour Annotation Service part provides the functionality needed for annotating rumours and storing them in a database. It has a role of an interface for interacting with users. It retrieves the social media posts, classifies them using the Rumour Classification System and sends this information back to the user. Since journalists' time is precious, we focus on scenarios where they are asked to label no more than ten to fifteen examples per rumour. Journalists will optionally be able to provide further examples if this fit their workflow.

The service can be accessed through a web Graphical User Interface (GUI) front-end (Fig. 3) that can be used standalone or as REST API which makes it possible to be easily incorporated into existing journalist platforms and tools. As long as

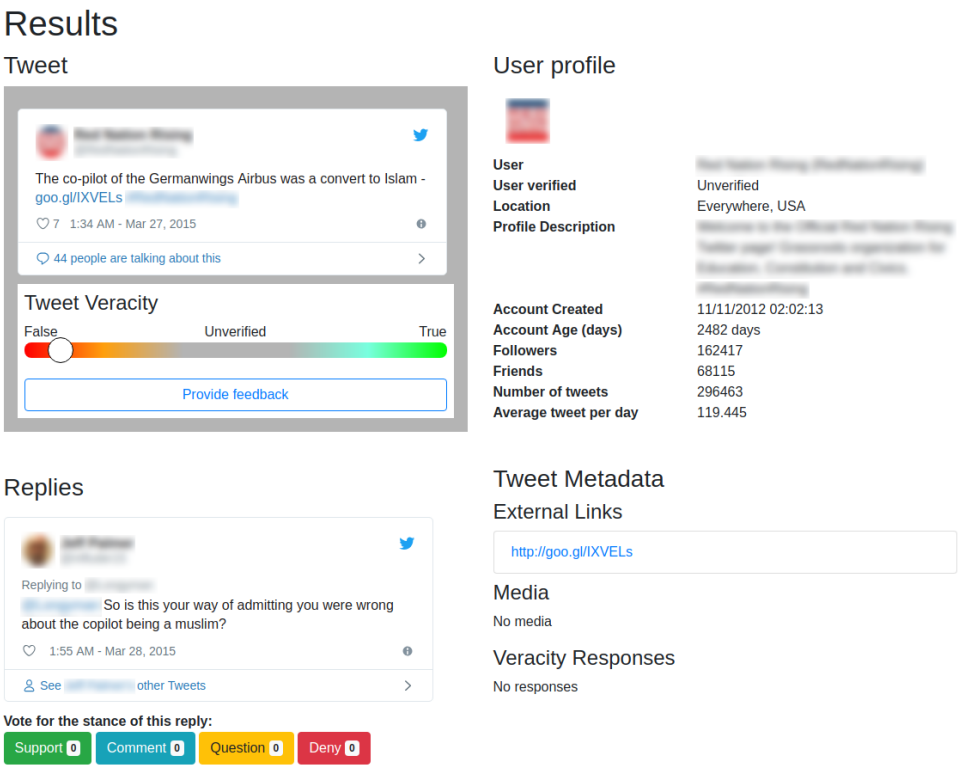


Figure 3: A screenshot of the web-based interface. The source tweet is shown on the top left. The veracity classification is displayed below the tweet on a single axis scale that ranges between False (red), Unverified (grey) and True (green). Metadata about the tweet is shown on the right. Replies to the tweet are shown on the bottom left where journalists can annotate the stance of each reply.

a journalist chooses to annotate a rumour, it will be instantly stored into the database, allowing the classification system to get updated regularly by leveraging the newly annotated rumours.

The journalist can currently make two types of annotations. Firstly, annotations on the veracity of the rumour itself. Whether it is true, false, or unverified, and are encouraged to provide evidence for making this claim. Secondly, they can annotate on the stance of the responses to the rumour. The stance of the response can either be to support the claim, deny the claim, or to offer further questions or comments, which we currently regard as neutral stance. When creating a dataset for re-training with user-provided annotations, each tweet, for both rumour veracity and stance classification, uses the class with the majority vote. Each tweet must also have 50% or more votes in the majority category to be used.

## 6 Conclusion and Future Work

This paper presented an open source web service for analysis of rumours on social media. The system uses a state-of-the-art deep neural network model (Aker et al., 2019) to classify Twitter rumours and allows journalists to provide annotated feedback to the system allowing the predictions to be improved as it is used.

In the future, we intend to also integrate a rumour stance classifier. User credibility will be taken into account to affect the influence of their annotations by analysing the accuracy and quality of provided evidence. In this demo, we focused only on the Twitter platform but we plan to offer support for other social media platforms, such as Reddit and 4chan. Finally, as new rumour classification models are developed, they could easily be integrated into the system to provide an ensemble of predictions.

## Acknowledgements

This research was supported by a Google DNI Prototype project and the WeVerify project (partially funded by the European Commission under contract number 825297).

## References


Ahmet Aker, Alfred Sliwa, Fahim Dalvi, and Kalina Bontcheva. 2019. Rumour verification through recurring information and an inner-attention mechanism. *Online Social Networks and Media*, 13:100045.

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

Anh Dang, Abidalrahman Moh'd, Evangelos Milios, and Rosane Minghim. 2016. What is in a rumour: Combined visual analysis of rumour flow and user activity. In *Proceedings of the 33rd Computer Graphics International*, CGI '16, pages 17–20, New York, NY, USA. ACM.

L Derczynski and Kalina Bontcheva. 2014. Pheme: Veracity in digital social networks. *CEUR Workshop Proceedings*, 1181:19–22.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Omar Enayet and Samhaa R El-Beltagy. 2017. Niletmrg at semeval-2017 task 8: Determining rumour and veracity support for rumours on twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 470–474.

William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1163–1168.

Naeemul Hassan, Anil Nayak, Vikas Sable, Chengkai Li, Mark Tremayne, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, and Aaditya Kulkarni. 2017. Claimbuster: the first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10:1945–1948.

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

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

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2018. Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection.

Michal Lukasik, Trevor Cohn, and Kalina Bontcheva. 2015. Estimating collective judgement of rumours in social media. *ArXiv*, abs/1506.00468.

Panagiotis Takas Metaxas, Samantha Finn, and Eni Mustafaraj. 2015. Using twittertrails.com to investigate rumor propagation. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, CSCW'15 Companion, pages 69–72, New York, NY, USA. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Paul Resnick, Samuel Carton, Souneil Park, Yuncheng Shen, and Nicole Zeffer. 2014. Rumorlens: A system for analyzing the impact of rumors and corrections in social media. In *Proc. Computational Journalism Conference*, pages 10121–0701.

Chengcheng Shao, Giovanni Luca Ciampaglia, Alessandro Flammini, and Filippo Menczer. 2016. Hoaxy: A platform for tracking online misinformation. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 745–750, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Denis Teyssou, Jean-Michel Leung, Evlampios Apostolidis, Konstantinos Apostolidis, Symeon Papadopoulos, Markos Zampoglou, Olga Papadopoulou, and Vasileios Mezaris. 2017. The invid plug-in: Web video verification on the browser. In *Proceedings of the First International Workshop on Multimedia Verification*, MuVer '17, pages 23–30, New York, NY, USA. ACM.

Adam Tsakalidis, Nikolaos Aletras, Alexandra I Cristea, and Maria Liakata. 2018. Nowcasting the stance of social media users in a sudden vote: The case of the Greek Referendum. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM, pages 367–376.

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.