Classifying Tweet Level Judgements of Rumours in Social Media

Michal Lukasik,[1] **Trevor Cohn**[2] and **Kalina Bontcheva**[1]
[1]Computer Science
The University of Sheffield
{m.lukasik,k.bontcheva}@shef.ac.uk
[2]Computing and Information Systems
The University of Melbourne
t.cohn@unimelb.edu.au

Abstract

Social media is a rich source of rumours and corresponding community reactions. Rumours reflect different characteristics, some shared and some individual. We formulate the problem of classifying tweet level judgements of rumours as a supervised learning task. Both supervised and unsupervised domain adaptation are considered, in which tweets from a rumour are classified on the basis of other annotated rumours. We demonstrate how multi-task learning helps achieve good results on rumours from the 2011 England riots.

1 Introduction

There is an increasing need to interpret and act upon rumours spreading quickly through social media, especially in circumstances where their veracity is hard to establish. For instance, during an earthquake in Chile rumours spread through Twitter that a volcano had become active and that there was a tsunami warning in Valparaiso (Mendoza et al., 2010). Other examples, from the riots in England in 2011, were that rioters were going to attack Birmingham's children hospital and that animals had escaped from the zoo (Procter et al., 2013).

Social scientists (Procter et al., 2013) analysed manually a sample of tweets expressing different judgements towards rumours and categorised them manually in supporting, denying or questioning. The goal here is to carry out tweet-level judgement classification automatically, in order to assist in (near) real-time rumour monitoring by journalists and authorities (Procter et al., 2013). In addition, information about tweet-level judgements has been used as a first step for early rumour detection by (Zhao et al., 2015).

The focus here is on tweet-level judgement classification on unseen rumours, based on a training set of other already annotated rumours. Previous work on this problem either considered unrealistic settings ignoring temporal ordering and rumour identities (Qazvinian et al., 2011) or proposed regular expressions as a solution (Zhao et al., 2015). We expect posts expressing similar opinions to exhibit many similar characteristics across different rumours. Based on the assumption of a common underlying linguistic signal, we build a transfer learning system that labels newly emerging rumours for which we have little or no annotated data. Results demonstrate that Gaussian Process-based multi task learning allows for significantly improved performance.

text	position
Birmingham Children's hospital has been attacked. F***ing morons. #UKRiots	support
Girlfriend has just called her ward in Birmingham Children's Hospital & there's no sign of any trouble #Birminghamriots	deny
Birmingham children's hospital guarded by police? Really? Who would target a childrens hospital #disgusting #Birminghamriots	question

Table 1: Tweets on a rumour about hospital being attacked during 2011 England Riots.

The novel contributions of this paper are: 1. Formulating the problem of classifying judgements of rumours in both supervised and unsupervised domain adaptation settings. 2. Showing how a multi-task learning approach outperforms single-task methods.

2 Related work

In the context of rumour spread in social media, researchers have studied differences in informa-

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2590–2595,
Lisbon, Portugal, 17-21 September 2015.

tion flows between content of varying credibility. For instance, Procter et al. (2013) grouped source tweets and re-tweets into information flows (Lotan et al., 2011), then ranked these by flow size, as a proxy of significance. Information flows were then categorised manually. Along similar vein, Mendoza et al. (2010) found that users deal with true and false rumours differently: the former are affirmed more than 90% of the time, whereas the latter are challenged (questioned or denied) 50% of the time. Friggeri et al. (2014) analyzed a set rumours from the *Snopes.com* website that have been matched to Facebook public conversations. They concluded that false rumours are more likely to receive a comment with link to *Snopes.com* website. However, none of the above attempted to automatically classify rumours.

With respect to automatic methods for detecting misinformation and disinformation in social media, Ratkiewicz et al. (2011) detect political abuse (a kind of disinformation) spread through Twitter. The task is defined in purely information diffusion settings and is not necessarily related with the truthfulness of the piece of information. Castillo et al. (2013) proposed methods for identifying newsworthy information cascades on Twitter and then classifying these cascades as credible and not credible. The main difference from our task is that credibility classification is carried out over the entire information cascade, classified objects are not necessarily rumours and no explicit judgement classification was performed in their approach.

Early rumour identification is the focus of Zhao et al. (2015), where regular expressions are used for finding questioning and denying tweets as a key pre-requisite step for rumour detection. Unfortunately, when we applied these regular expressions on our dataset, they yielded only 16% recall for questioning and 14% recall for denying tweets. Consequently, this motivated us to seek a better approach to tweet-level classification.

The work most relevant to ours is due to Qazvinian et al. (2011). Their method first carries out rumour retrieval, whereby tweets are classified into rumour related and non-rumour related. Next, rumour-related tweets are classified into supporting and not-supporting. The classifier is trained by ignoring rumour identities, i.e., pooling together tweets from all rumours, and ignoring the temporal dependencies between tweets. In contrast, we formulate the rumour classification problem as transfer learning, where unseen rumours (or rumours with few initial tweets observed) are classified using already known rumours – a much harder and more practical setting. Moreover, unlike Qazvinian et al. (2011), we consider the multi-class classification problem and do not collaps questioning and denying tweets into a single class, since they differ significantly.

Rumour	Supporting	Denying	Questioning
army bank	62	42	73
hospital	796	487	132
London Eye	177	295	160
McDonald's	177	0	13
Miss Selfridge's	3150	0	7
police beat girl	783	4	95
zoo	616	129	99

Table 2: Counts of tweets with supporting, denying or questioning labels in each rumour collection.

3 Data

We evaluate our work on several rumours circulating on Twitter during the England riots in 2011 (see Table 2). The dataset was analysed and annotated manually as supporting, questioning, or denying a rumour, by a team of social scientists studying the role of social media during the riots (Procter et al., 2013). The original dataset also included commenting tweets, but these have been removed from our experiments due to their small number (they constituted only 5% of the corpus).

As can be seen from the dataset overview in Table 2, different rumours exhibit varying proportions of supporting, denying and questioning tweets, which was also observed in other studies of rumours (Mendoza et al., 2010; Qazvinian et al., 2011). These variations in majority classes across rumours underscores the modeling challenge in tweet-level classification of rumour attitudes.

With respect to veracity, one rumour has been confirmed as true (Miss Selfridge's being on fire), one is unsubstantiated (police beat girl), and the remaining five are known to be false. Note, however, that the focus here is not on classifying truthfulness, but instead on identifying the attitude expressed in each tweet towards the rumour.

4 Problem formulation

Let R be a set of rumours, each of which consists of tweets discussing it, $\forall_{r \in R}\, T_r = \{t^r_1, \cdots, t^r_{r_n}\}$. $T = \cup_{r \in R} T_r$ is the complete set of tweets from all rumours. Each tweet is classified as supporting, denying or questioning with respect to its rumour: $y(t) \in \{0, 1, 2\}$, where 0 denotes supporting, 1 means denying and 2 denotes questioning.

First, we consider the Leave One Out (LOO) setting, which means that for each rumour $r \in R$, we construct the test set equal to T_r and the training set equal to $T \setminus T_r$. Therefore this is a very challenging and realistic scenario, where the test set contains an entirely unseen rumour, from those in the training set.

The second setting is Leave Part Out (LPO). In this formulation, a very small number of initial tweets from the target rumour is added to the training set $\{t^r_1, \cdots, t^r_{r_k}\}$. This scenario becomes applicable typically soon after a rumour breaks out and journalists have started monitoring and analysing the related tweet stream. The experiments section investigates how the number of initial training tweets influences classification performance on a fixed test set, namely: $\{t^r_{r_l}, \cdots, t^r_{r_n}\}$, $l > k$.

The tweet-level classification problem here assumes that tweets from the training set are already labelled with the rumour discussed and the attitude expressed towards that. This information can be acquired either via manual annotation as part of expert analysis, as is the case with our dataset, or automatically, e.g. using pattern-based rumour detection (Zhao et al., 2015). Afterwards, our method can be used to classify the attitudes expressed in each new tweet from outside the training set.

5 Gaussian Processes for Classification

Gaussian Processes are a Bayesian non-parametric machine learning framework that has been shown to work well for a range of NLP problems, often beating other state-of-the-art methods (Cohn and Specia, 2013; Lampos et al., 2014; Beck et al., 2014; Preotiuc-Pietro et al., 2015). We use Gaussian Processes as this probabilistic kernelised framework avoids the need for expensive cross-validation for hyperparameter selection.[1]

[1]There exist frequentist kernel methods, like SVMs, which additionally require extensive heldout parameter tuning.

The central concept of Gaussian Process Classification (GPC; (Rasmussen and Williams, 2005)) is a latent function f over inputs $\mathbf{x}$: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where m is the mean function, assumed to be 0 and k is the kernel function, specifying the degree to which the outputs covary as a function of the inputs. We use a linear kernel, $k(\mathbf{x}, \mathbf{x}') = \sigma^2 \mathbf{x}^\top \mathbf{x}'$. The latent function is then mapped by the probit function $\Phi(f)$ into the range $[0, 1]$, such that the resulting value can be interpreted as $p(y = 1|\mathbf{x})$.

The GPC posterior is calculated as

$$p(f^*|X, \mathbf{y}, \mathbf{x}_*) = \int p(f^*|X, \mathbf{x}_*, \mathbf{f}) \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{y}|X)} d\mathbf{f},$$

where $p(\mathbf{y}|\mathbf{f}) = \prod_{j=1}^{n} \Phi(f_j)^{y_j}(1 - \Phi(f_j))^{1-y_j}$ is the Bernoulli likelihood of class y. After calculating the above posterior from the training data, this is used in prediction, i.e.,

$$p(y_* = 1|X, \mathbf{y}, \mathbf{x}_*) = \int \Phi(f_*)\, p(f_*|X, \mathbf{y}, \mathbf{x}_*)\, df_*\,.$$

The above integrals are intractable and approximation techniques are required to solve them. There exist various methods to deal with calculating the posterior; here we use Expectation Propagation (EP; (Minka and Lafferty, 2002)). In EP, the posterior is approximated by a fully factorised distribution, where each component is assumed to be an unnormalised Gaussian.

In order to conduct multi-class classification, we perform a one-vs-all classification for each label and then assign the one with the highest likelihood, amongst the three (supporting, denying, questioning). We choose this method due to interpretability of results, similar to recent work on occupational class classification (Preotiuc-Pietro et al., 2015).

Intrinsic Coregionalization Model In the LPO setting initial labelled tweets from the target rumour are observed as well. In this case, we propose to weight the importance of tweets from the reference rumours depending on how similar their characteristics are to the tweets from the target rumour available for training. To handle this with GPC, we use a multiple output model based on the Intrinsic Coregionalisation Model (ICM; (Álvarez et al., 2012)). It has already been applied successfully to NLP regression problems (Beck et al., 2014) and it can also be applied to classification

ones. ICM parametrizes the kernel by a matrix which represents the extent of covariance between pairs of tasks. The complete kernel takes form of

$$k((\mathbf{x}, d), (\mathbf{x}', d')) = k_{data}(\mathbf{x}, \mathbf{x}')B_{d,d'} ,$$

where B is a square coregionalization matrix, d and d' denote the tasks of the two inputs and k_{data} is a kernel for comparing inputs $\mathbf{x}$ and $\mathbf{x}'$ (here, linear). We parametrize the coregionalization matrix $B = \boldsymbol{\kappa} I + \boldsymbol{v}\boldsymbol{v}^T$, where $\boldsymbol{v}$ specifies the correlation between tasks and the vector κ controls extent of task independence.

Hyperparameter selection We tune hyperparameters $\mathbf{v}$, $\boldsymbol{\kappa}$ and σ^2 by maximizing evidence of the model $p(\mathbf{y}|X)$, thus having no need for a validation set.

Methods We consider GPs in three different settings, varying in what data the model is trained on and what kernel it uses. The first setting (denoted GP) considers only target rumour data for training. The second (GPPooled) additionally considers tweets from reference rumours (i.e. other than the target rumour). The third setting is GPICM, where an ICM kernel is used to weight influence from tweets from reference rumours.

6 Features

We conducted a series of preprocessing steps in order to address data sparsity. All words were lowercased; stopwords removed; all emoticons were replaced with words[2]; and stemming was performed. In addition, multiple occurrences of a character were replaced with a double occurrence (Agarwal et al., 2011), to correct for misspellings and lengthenings, e.g., *looool*. All punctuation was also removed, except for ., *!* and *?*, which we hypothesize to be important for expressing emotion. Lastly, usernames were removed as they tend to be rumour-specific, i.e., very few users comment on more than one rumour.

After preprocessing the text data, we use either the resulting bag of words (BOW) feature representation or replace all words with their Brown cluster ids (Brown), using 1000 clusters acquired from a large scale Twitter corpus (Owoputi et al., 2013). In all cases, simple re-tweets are removed from the training set to prevent bias (Llewellyn et al., 2014).

[2]We used the dictionary from: `http://bit.ly/1rX1Hdk` and extended it with: :o, : |, =/, :s, :S, :p.

method	acc
Majority	0.68
GPPooled Brown	0.72
GPPooled BOW	0.69

Table 3: Accuracy taken across all rumours in the LOO setting.

7 Experiments and Discussion

Table 3 shows the mean accuracy in the LOO scenario following the GPPooled method, which pools all reference rumours together ignoring their task identities. ICM can not use correlations to target rumour in this case and so can not be used. The majority baseline simply assigns the most frequent class from the training set.

We can observe that methods perform on a level similar to majority vote, outperforming it only slightly. This indicates how difficult the LOO task is, when no annotated target rumour tweets are available.

Figure 1 shows accuracy for a range of methods as the number of tweets about the target rumour used for training increases. Most notably, performance increases from 70% to around 80%, after only 10 annotated tweets from the target rumour become available, as compared to the results on unseen rumours from Table 3. However, as the amount of target rumour increases, performance does not increase further, which suggests that even only 10 human-annotated tweets are enough to achieve significant performance benefits. Note also how the use of reference rumours is very important, as methods using only the target rumour obtain accuracy similar to the Majority vote classifier (GP Brown and GP BOW).

The top performing methods are GPCIM and GPPooled, where use of Brown clusters consistently improves results for both methods over BOW, irrespective of the number of tweets about the target rumour annotated for training. Moreover, GPICM is better than GPPooled both with Brown and BOW features and GPCIM with Brown is ultimately the best performing of all.

In order to analyse the importance of Brown clusters, Automatic Relevance Determination (ARD) is used (Rasmussen and Williams, 2005) for the best performing GPICM Brown in the LPO scenario. Only the case where the first 10 tweets are used for training is considered, since it already

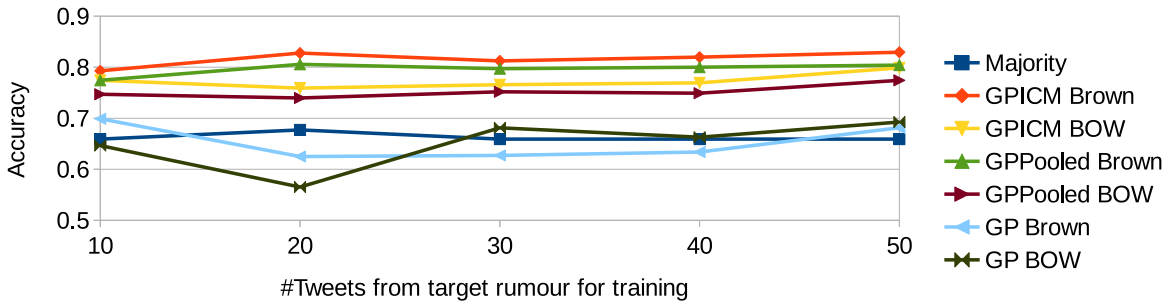

Figure 1: Accuracy measures for different methods versus the size of the target rumour used for training in the LPO setting. The test set is fixed to all but the first 50 tweets of the target rumour.

supporting	denying	questioning
?	fake	?
10001101	11111000001	10001101
!	not	!
10001100	001000	10001100
not	?	hope
001000	10001101	01000111110
fake	!	true
11111000001	10001100	111110010110
true	bullshit	searching
111110010110	11110101011111	01111000010

Table 4: Top 5 Brown clusters, each shown with a representative word. For further details please see the cluster definitions at `http://www.ark.cs.cmu.edu/TweetNLP/cluster_viewer.html`.

performs very well. Using ARD, we learn a separate length-scale for each feature, thus establishing their importance. The weights learnt for different clusters are averaged over the 7 rumours and the top 5 Brown clusters for each label are shown in Table 4. We can see that clusters around the words *fake* and *bullshit* turn out to be important for the denying class, and *true* for both supporting and questioning classes. This reinforces our hypothesis that common linguistic cues can be found across multiple rumours. Note how punctuation proves important as well, since clusters *?* and *!* are also very prominent.

8 Conclusions

This paper investigated the problem of classifying judgements expressed in tweets about rumours. First, we considered a setting where no training data from target rumour is available (LOO). Without access to annotated examples of the target rumour the learning problem becomes very difficult. We showed that in the supervised domain adaptation setting (LPO) even annotating a small number of tweets helps to achieve better results. Moreover, we demonstrated the benefits of a multi task learning approach, as well as that Brown cluster features are more useful for the task than simple bag of words.

Judgement estimation is undoubtedly of great value e.g. for marketing, politics and journalism, helping to target widely believed topics. Although the focus here is on classifying community reactions, Castillo et al. (2013) showed that community reaction is correlated with actual rumour veracity. Consequently our classification methods may prove useful in the broader and more challenging task of annotating veracity.

An interesting direction for future work would be adding non-textual features. For example, the rumour diffusion pattern (Lukasik et al., 2015) may be a useful cue for judgement classification.

Acknowledgments

Work partially supported by the European Union under grant agreement No. 611233 PHEME. The work was implemented using the GPy toolkit (GPy authors, 2015).

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. 2012. Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.*, 4(3):195–266.

Daniel Beck, Trevor Cohn, and Lucia Specia. 2014. Joint emotion analysis via multi-task Gaussian processes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 1798–1803.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2013. Predicting information credibility in time-sensitive social media. *Internet Research*, 23(5):560–588.

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task Gaussian processes: An application to machine translation quality estimation. In *51st Annual Meeting of the Association for Computational Linguistics*, ACL '13, pages 32–42.

Adrien Friggeri, Lada Adamic, Dean Eckles, and Justin Cheng. 2014. Rumor cascades. In *International AAAI Conference on Weblogs and Social Media*.

The GPy authors. 2015. GPy: A Gaussian process framework in Python. `http://github.com/SheffieldML/GPy`.

Vasileios Lampos, Nikolaos Aletras, Daniel Preotiuc-Pietro, and Trevor Cohn. 2014. Predicting and characterising user impact on twitter. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, EACL'14, pages 405–413.

Clare Llewellyn, Claire Grover, Jon Oberlander, and Ewan Klein. 2014. Re-using an argument corpus to aid in the curation of social media collections. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC'14, pages 462–468.

Gilad Lotan, Erhardt Graeff, Mike Ananny, Devin Gaffney, Ian Pearce, and danah boyd. 2011. The Arab spring— the revolutions were tweeted: Information flows during the 2011 Tunisian and Egyptian revolutions. *International Journal of Communication*, 5(0).

Michal Lukasik, Trevor Cohn, and Kalina Bontcheva. 2015. Point process modelling of rumour dynamics in social media. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, pages 518–523.

Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. 2010. Twitter under crisis: Can we trust what we RT? In *1st Workshop on Social Media Analytics*, SOMA'10, pages 71–79.

Thomas Minka and John Lafferty. 2002. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, pages 352–359.

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL*, pages 380–390.

Daniel Preotiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. 2015. An analysis of the user occupational class through twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, pages 1754–1764.

Rob Procter, Jeremy Crump, Susanne Karstedt, Alex Voss, and Marta Cantijoch. 2013. Reading the riots: What were the police doing on twitter? *Policing and society*, 23(4):413–436.

Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1589–1599.

Carl Edward Rasmussen and Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonalves, Alessandro Flammini, and Filippo Menczer. 2011. Detecting and tracking political abuse in social media. In *5th International AAAI Conference on Weblogs and Social Media*, ICWSM'11.

Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Early detection of rumors in social media from enquiry posts. In *International World Wide Web Conference Committee (IW3C2)*.