Estimating collective judgement of rumours in social media

Michal Lukasik
University of Sheffield
m.lukasik@shef.ac.uk

Trevor Cohn
University of Melbourne
t.cohn@unimelb.edu.au

Kalina Bontcheva
University of Sheffield
k.bontcheva@shef.ac.uk

Abstract

Social media is a rich source of rumours and corresponding community reactions. Determining the extent of community belief in a rumour is of value for marketing, politics and journalism. Moreover, collective judgements have been shown to correlate with ground truth. We formulate the problem of estimating the collective judgement of a rumour in social media as a supervised learning using annotated examples. Rumours reflect different characteristics, some shared and some individual. We consider both supervised and unsupervised domain adaptation, in which rumour predictions are made on the basis of other annotated rumours. Evaluation on a set of several thousands tweets on seven rumours shows that we can successfully predict individual and collective judgements.

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], which turned out to be false. Other examples, from the riots in England in 2011, are that rioters were going to attack Birmingham’s children hospital and that animals had escaped from the zoo [Procter et al. 2013].

Social media provide a sample of community’s reactions to rumours. Some examples for the rumour about a hospital being attacked during the England riots are shown in Table 1. We can see how different views are given, such as support, rejection or questioning.

We introduce the problem of estimating collective trust in a new rumour. The settings we consider are when one has access to a set of rumours that have already been analysed. 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 new rumours for which we have either no or little annotated data. We show that standard supervised learning models are consistently outperformed by multi task learning.

Collective 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 aggregate community reactions, Castillo et al. [2013] has shown 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.

Annotating all tweets for a target rumour is too
costly, especially when we deal with many stories, each consisting of thousands (or more) of tweets. However, it may be useful to do some annotation. We consider second setting in which part of a target rumour is annotated and seen during training. We build a sophisticated solution based on Gaussian Processes for classification.

The novel contributions of this paper are:
1. Formulating the problem of classifying collective judgement of rumours in both supervised and unsupervised domain adaptation settings.
2. Showing how a multi-task learning approach is successful in both settings.
3. Formulating a Gaussian Process Classification model for multi-task learning, leading to the best results in the supervised setting.

2 Related work

In the context of rumour spread in social media, researchers have studied differences in information 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 and used the groupings to explore how a given rumour spread through Twitter. 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.

The work most relevant to ours is 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 classes. The classifier is trained by ignoring rumour identities, i.e., pooling together tweets from all their rumours, and ignoring the temporal dependencies between tweets. In contrast, we consider the rumour classification problem as a form of transfer learning, and seek to classify unseen rumours from other known rumours, a much harder and more practical setting. We also consider classifying a rumour when some annotated tweets can be used. However, we restrict ourselves to the setting in which only a first half of the tweet stream is used, a difficult scenario due to the temporal evolution of community behaviour.

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 by social scientists studying the role of social media during the riots [Procter et al., 2013]. Tweets have been manually categorised as supporting, questioning, or rejecting the rumour. The origin-counts of tweets with support or deny labels in each rumour collection, as well as their ratio and verified truthfulness (× = false, ? = unknown, ✓ = true).

| Rumour      | Support | Deny | Ratio | Truth |
|-------------|---------|------|-------|-------|
| army bank   | 62      | 115  | 0.54  | ×     |
| hospital    | 796     | 619  | 1.29  | ×     |
| London Eye  | 177     | 455  | 0.39  | ×     |
| McDonald’s  | 177     | 13   | 13.62 | ×     |
| Miss Selfridge’s | 3150 | 7    | 450.00| ✓     |
| police beat girl | 783  | 99   | 7.91  | ?     |
| zoo         | 616     | 228  | 2.70  | ×     |

Table 2: Counts of tweets with support or deny labels in each rumour collection, as well as their ratio and verified truthfulness (× = false, ? = unknown, ✓ = true).

1See also http://www.theguardian.com/uk/interactive/2011/dec/07/london-riots-twitter for a detailed analysis of these rumours in the media.
inal 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). Following the approach of Qazvinian et al. (2011), rejecting and questioning tweets for each rumour are joined into one class, which is referred to as the deny class.

Table 2 summarizes information about the corpora. Due to varying degrees of community belief, different rumours exhibit various supporting vs denying ratios, which was also observed in other studies of rumour corpora (Mendoza et al. 2010; Qazvinian et al. 2011). This inconsistency in what label prevails underscores the modelling challenge in automatic classification of rumours, especially without explicit supervision. Moreover, Zubiaga et al. (2015) in their study of the Ferguson riots found that rumours were highly heterogenous, with varying length and on several different topics.

From the rumours, one has been confirmed to be true (Miss Selfridge’s being on fire), one is unsubstantiated (police beat girl), and the remaining five are known to be false. This aligns with the findings of previous work, which have shown that true rumours are rare (Kate et al. 2014; Qazvinian et al. 2011). Note that we are not primarily interested in truthfulness of rumours, but rather in the judgements expressed by the online community.

Nevertheless, there is a strong relationship between community support and rumour veracity (compare ‘ratio’ and ‘truth’ in Table 2). The highest ratio of supporting vs denying tweets is for the Miss Selfridge’s rumour, which is the only one proven to be true. The unsubstantiated rumour about police beating a girl has the third highest ratio. It may come as a surprise that McDonald’s rumour has a high ratio, even though it has been proved to be false. This might relate to it being a very catchy story (rioters breaking into McDonald’s and cooking their own food), which many people would like to believe.

Even though this is the biggest rumour dataset used for classifier evaluation, all rumours revolve around a single theme being London riots. Nevertheless, there is a very small vocabulary overlap between the rumours (12-25%). This is later confirmed by Table 5 reporting most indicative words for rumour classification, where most words are unrelated to London riot events.

4 Problem formulation

Let \( R \) be a set of rumours, each of which consists of tweets describing it, \( \forall r \in R \quad T_r = \{ t_1^r, \ldots, t_n^r \}. \) \( T = \cup_{r \in R} T_r \) is the complete set of tweets from all rumours. Each tweet is classified as being supporting or denying with respect to its rumour, \( y(t) \in \{ 0, 1 \} \), where 1 denotes support and 0 means rejection.

We consider two settings for our problem. The first is Leave One Out (LOO), which means that for each rumour \( r \in R \), we construct the test set equal to \( T_r \) and the training set equal to \( T - T_r \). Therefore this is a challenging problem where the test set comes from a different source than the training set and so requires careful model not to overfit or learn negative transfer.

The second setting is Leave Part Out (LPO). In this formulation, we add the first half of a target rumour for training \( \{ t_1^r, \ldots, t_{r/2}^r \} \), leaving the rest for testing \( \{ t_{r/2+1}^r, \ldots, t_n^r \} \). This setting may be used when initial tweets are accessible for annotation when the rumour breaks out.

5 Methods

Baselines Our baseline classifier is logistic regression (LR) with \( \ell_1 \) regularisation and bag-of-words features (BOW). In addition, we also experimented with a Support Vector Machines classifier with a linear (SVMLIN) or RBF (SVMRBF) kernel, both using BOW features. We also compare ourselves against the method of Qazvinian et al. (2011), who first extracts features using Naive Bayes (NB) and then reweights the NB component log-probabilities via logistic regression (NBLR) or a linear SVM (NBSVMLIN).

Multi task learning via mean regularization

The dataset consists of different rumours, which can exhibit different characteristics. Therefore we use multi task learning, where we treat each rumour as a separate task. We experimented with several different methods, finding that mean regularization (MR) was very effective (Evgeniou and Pontil 2004). In this setting we jointly learn several per-task models with a regularizer that drives task specific weight vectors towards the mean weights over all tasks. The
The training objective is defined as
\[
\sum_{i=1}^{R} \sum_{j=1}^{n_i} \log \left( 1 + \exp \left( -y_{i,j} \left( w_1^T x_{i,j} \right) \right) \right) + \lambda_1 \frac{1}{R} \sum_{s=1}^{R} \left\| w_s \right\|_F^2 + \lambda_2 \| W \|_1 , \tag{1}
\]
where \( R \) is the number of rumours, \( n_i \) is number of tweets per \( i \)-th rumour, \( x_{i,j} \) are the feature vector of the \( j \)-th tweet from the \( i \)-th rumour, \( y_{i,j} \) is the gold response, and \( w_t \) is the weight vector for rumour \( t \). The first regularization component biases feature vectors to be close to the global mean and the second regularizer induces sparsity in the weights.

Using the method above, we learn specific weight vectors for each rumour present in the training set. In the LPO setting, at prediction time we simply use the weights for the target rumour. In the LOO setting this is more difficult since we have no feature vector for the target rumour, as it is excluded from training. In this case, we use the averaged weight vector, which we expect to contain the core underlying signal useful for general rumour classification.

**Gaussian Process 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). It is most widely used for regression, however as we describe below, it can also be used for classification.

We use Gaussian Processes because they explicitly handle uncertainty which we hypothesize to better handle complex rumour settings, in particular label imbalance. Moreover, as will be stated later, the multitask setting of Gaussian Processes we use is a generalization of MR.

The central concept of Gaussian Process Classification (GPC; Rasmussen and Williams, 2005) is a latent function \( f \) over inputs \( x: f(x) \sim GP(m(x), k(x, x')) \), 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 Radial Basis Function (RBF) kernel,
\[
k(x, x') = \sigma^2 \exp \left( -\sum_{i=1}^{d} \frac{(x_i - x'_i)^2}{l^2} \right),
\]
linear kernel, \( k(x, x') = \sigma^2 x^T x' \). 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 \( p(y = 1|x) \).

The GPC posterior is calculated as
\[
p(f^*|X, y, x_*) = \int p(f^*|X, x_*, f)p(y|f)p(f)df ,
\]
where \( p(y|f) = \prod_{j=1}^{n} \Phi(f_j)^{y_j}(1 - \Phi(f_j))^{1-y_j} \) is the Bernoulli likelihood of class \( y \). After calculating the above posterior from the training data, this is used in prediction, i.e.,
\[
p(y_* = 1|X, y, x_*) = \int \Phi(f_*) p(f_*|X, y, x_*) df_* .
\]

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. The parameters of the approximation are iteratively refined to minimise Kullback-Leibler divergence between the true posterior and the approximation. As convergence is not guaranteed in general, we conduct 11 iterations and select the approximation parameters from the iteration which maximises model evidence, \( p(y|X) \). The EP approximation yields a Gaussian posterior, leading to an analytical solution to the second integral. For more details on Gaussian Processes we refer the reader to Rasmussen and Williams (2005).

**Intrinsic Coregionalization Model** Our multitask experimental setting requires joint inference over several different classification tasks, one for each rumour. To handle this with GPC we use a multiple output model based on the Intrinsic Coregionalisation Model (ICM) (Álvarez et al., 2012), which has previously been successfully applied to NLP regression problems (Beck et al., 2014). This parametrizes the kernel by a matrix which represents the extent of covariance between pairs of tasks. The complete kernel takes form of
\[
k((x, d), (x', d')) = k_{data}(x, x') B_{d,d'} ,
\]
where B is a square coregionalization matrix, d and \(d'\) denote the tasks of the two inputs and \(k_{\text{data}}\) is a kernel for comparing inputs x and x′ (here, linear or RBF).

We parametrize the coregionalization matrix \(B = WW' + \kappa I\), where W is a rectangular matrix with a user specified number of columns (relating to the rank of B) and the vector \(\kappa\) controls extent of task independence. This way we can control the number of hyperparameters and expressiveness of the model. In fact the mean regularisation technique described above turns out to be a special case of ICM (Evgeniou et al., 2005). Needless to say, the former method is simpler, and in this paper we empirically compare both approaches.

### Hyperparameter selection

In case of the frequentist methods (LR, SVM, NBLR, NBSVM, Mean Regularization) we select the hyperparameter values for each rumour using the training set differently in unsupervised and supervised domain adaptation settings. For the LOO setting, hyperparameter selection is conducted via nested LOO on the training rumours, where the choice is on hyperparameters that yield the best average accuracy. In LPO setting, 40% of the target rumour is held out for validation set, on which each hyperparameters value set is evaluated. The considered values for each hyperparameter are \{0.0001, 0.001, 0.01, 0.1, 1, 10, 100\}.

As for the GPC model, we tune hyperparameters \(W, \kappa, \sigma^2, l\) by maximizing evidence of the model \(p(y|X)\), thus having no need for a validation set.

### 6 Features

In this section we discuss the features used by all models and baselines for our task. All are text based, therefore we first outline the preprocessing steps. Afterwards, we describe a method we used for dimensionality reduction.

#### 6.1 Text Preprocessing

We conducted a series of preprocessing steps in order to deal with data sparsity. We lowercased the words, removed stopwords, replaced emoticons with words\(^3\) and performed stemming. We replaced multiple occurrences of a character with a double occurrence, to correct for misspellings and lengthenings, e.g., looooool (Agarwal et al., 2011). Afterwards, we removed usernames and hashtags based on their rumour specificity, i.e., very few users comment on more than one rumour, and hash tags often signify events specific to a single rumour. We also removed simple re-tweets from the training set, which would otherwise unduly bias results (Llewellyn et al., 2014).

#### 6.2 Spectral Features

Apart from using BOW features, we employ spectral clustering for dimensionality reduction (Ng et al., 2001; Shi and Malik, 2000). This method has been demonstrated to be useful in modelling Twitter data for user impact regression (Lampos et al., 2014). Spectral clustering conducts graph partitioning on a word-word similarity matrix. For similarity we Normalised Pointwise Mutual Information (NPMI), which measures how similar are the contexts in which the words occur. The cluster centrality of a word is calculated by finding the average NPMI of this word to all other words from the cluster.

We run spectral clustering on tweets from all rumours with 100 clusters. In the first column of Table 3 we show words from cluster 1. It is apparent that central words about the ‘police beat girl’ rumour were assigned to it, which makes it of little use for other rumours. However, clusters 8, 9 and 11 contain general terms that are not indicative of a particular rumour, and stand to be more useful for transfer learning. The clusters are used to construct 100 additional features for each tweet, where each fea-

---

\(^2\)In nested LOO for each held-out target rumour, we perform an inner loop where we hold out another rumour. Hyperparameters are selected based on the average over the inner loop, which is then used to classify the target in the outer loop.

\(^3\)We used the dictionary from: [http://bit.ly/1rX1iQd](http://bit.ly/1rX1iQd) and extended it with: :o, :, |, =, /, :, :, :, :p.
| Method            | Accuracy |
|-------------------|----------|
| Mean regul (MR)   | 0.7863   |
| Oracle            | 0.7827   |
| SVMLIN            | 0.7327   |
| NBLR              | 0.6771   |
| NBSVMLIN          | 0.6771   |
| LR                | 0.6667   |

Table 4: Accuracy metrics taken across all rumours from LOO experiment. Note the Oracle has access to target rumour statistics.

7 Experiments and Discussion

In this section we report results from experiments on estimating collective judgement of rumours. We first show how different methods perform for unsupervised domain adaptation setting (LOO). Afterwards, we move to supervised formulation (LPO) and demonstrate how more sophisticated methods based on Gaussian Process Classification yield the best results.

7.1 Unsupervised Transfer Learning (LOO)

In Table 4 we show the mean accuracy from different methods across all rumours. We can see that Mean Regularizer has the highest mean and is the only method to achieve performance of Oracle (the most common class from target rumour). We also show results for each rumour in Figure 1. Note that even though NBLR is better in some cases, MR yields good results across all rumours. MR turns out to be the most successful method, exceeding the accuracy of the baselines LR and NBLR in 5 out of 7 cases and outperforming the SVM for 6 out of 7 rumours. Note also that our techniques show significant improvements over the NBLR and NBSVMLIN methods of Qazvinian et al. (2011).

In the LOO setting it is very difficult to select good hyperparameter values. Figure 2 illustrates how the actual accuracy differs from the estimated accuracy (computed using nested LOO) for different hyperparameter values, when using mean regularization. Note that the effect of hyperparameter $\lambda_2$ is reversed: the larger values are deemed to be very good in the nested LOO settings, whereas in reality they do not perform very well. This shows how different the rumours are, since the training rumours on average show different characteristics than army bank rumour on its own. It turns out that selecting large values for $\lambda_2$ hyperparameter to enforce high features sparsity yields better results, i.e., many words from the training set are not useful.

In Table 5 we show the most influential words together with their weights, according to the absolute value of the mean weight. We can see that words such as "fail", "irony" and "verify" are quite high, reinforcing our presumption from the introduction that common cues can be found across multiple rumours. However, the rumour specific words have not been
Table 5: Top 11 weights from mean regularization after averaging over all LOO iterations, according to the absolute value, together with rumour-specific weight values.

| feature   | army | hospital | LDN | MCD | miss | police | zoo | AVG |
|-----------|------|----------|-----|-----|------|--------|-----|-----|
| normal    | -1.91| -2.35    | -1.67| -2.14| -1.20| -2.13  | -2.35| -1.97|
| fail      | -0.88| -0.94    | -0.40| -0.93| -0.65| -1.02  | -0.60| -0.77|
| trouble   | -0.41| -0.59    | -0.38| -0.34| -0.23| -0.42  | 0.00 | -0.34|
| n         | -0.58| -0.16    | -0.20| -0.38| -0.06| -0.60  | -0.16| -0.31|
| whether   | 0.02 | 0.28     | 0.18 | 0.31 | 0.24 | 0.67   | 0.28 | 0.28 |
| app       | 0.26 | 0.03     | 0.33 | 0.23 | 0.09 | 0.36   | 0.06 | 0.20 |
| whsh      | 0.19 | 0.18     | 0.24 | 0.23 | 0.19 | 0.25   | 0.05 | 0.19 |
| verify    | 0.22 | 0.23     | 0.21 | 0.25 | 0.00 | 0.36   | 0.00 | 0.18 |
| iron      | 0.05 | -0.04    | -0.13| -0.22| -0.16| -0.36  | -0.23| -0.17|
| hooligan  | -0.19| -0.05    | -0.25| -0.20| -0.09| -0.35  | -0.04| -0.16|

Table 6: Accuracy metrics taken across all rumours from LPO setting.

| method                                      | accuracy |
|---------------------------------------------|----------|
| GPC RBF ICM BOW+spectral                   | 0.8617   |
| GPC RBF ICM BOW                            | 0.8562   |
| GPC LIN ICM BOW+spectral                   | 0.8546   |
| GPC LIN ICM BOW                            | 0.8588   |
| MR                                          | 0.8512   |
| MR LOO                                      | 0.7565   |
| LR                                          | 0.8373   |
| SVMLIN                                      | 0.8310   |
| SVMBRBF BOW                                 | 0.8068   |
| Oracle                                      | 0.7758   |
| NBLR                                        | 0.6798   |
| NBSVMLIN                                    | 0.5908   |

avoided, such as word hooligan below. This is due to the fact that all rumours come from the London Riots and so the word has been deemed to be a good indicator for a generalized rumour. Also, the ‘=’ sign has been assigned a significant weight, which seems to be evidence of overfitting, since this character occurs in just a few tweets in our dataset. As for the letter ‘n’, in our dataset it is shorthand for and or else resulting from tweet truncation.\footnote{Re-tweeting with added new content often results in the truncation of the original text.}

Next we use the outputs from the best method, Mean Regularization, to rank the rumours. We used the following metric: \[ \sum_{t: d(t)>0} d(t) \] where \( d(t) \) is the linear score for a tweet \( t \) together with a sign denoting the assigned label. We calculated the metric for each rumour based on the selected model using other rumours. In Figure 3 we show the true and predicted ranking of rumours according to the metric. The top four rumours are the same in both cases, with Miss Selfridge’s at the top. The bottom three stories change order: army bank, London Eye and hospital, which is not a big mistake, since these datasets have their gold ratios in the narrow range 0.39-1.29.

We also experimented with GPC but observed very poor performance. This may arise from optimisation issues relating to non-convexity of model training, or difficulties in formulating the predictive probability for handling an unseen task, however we plan to investigate further to posit a definite conclusion.

7.2 Supervised Transfer Learning (LPO)

We now move to the supervised domain adaptation setting, where we use first half of a target rumour to better train a classifier and to select hyperparameters. This may seem to be a significant amount of data, however as shall be seen later this problem is still difficult. Note that all other rumours are used for training, like in the previous LOO experiments.

The experimental setup has two steps. First, the hyperparameters are selected, optimizing the performance on the last 20% of the accessible data, after training on the first 30% of the target rumour (and the remaining rumours). Secondly, the best hyperparameters are chosen and a classifier is retrained on all accessible tweets on the target rumour and the remaining rumours.

In Table 6 we report results for a range of methods. We notice, that Gaussian Process Classification model with coregionalization, RBF kernel and spectral features used is the best method. On the other
hand, SVM with RBF kernel did not improve at all after incorporating additional spectral features (not reported in Table 6). The reason might be due to use of additive kernel joining BOW and spectral features in the GPC model, whereas in case of SVM the features were joined together and then treated as a single vector by a classifier. It also seems, that RBF kernel is better at exploiting spectral features, as can be seen from results on GPC with linear kernel.

Since spectral features were part of the best performing model, we now examine which of these clusters were most useful. To do so we employ automatic relevance determination (ARD), whereby each feature is assigned a separate learned length-scale inside the covariance kernel (see §5). We trained an ARD variant of the best GPC (RBF, ICM, BOW+Spectral) over all rumours. Two of the clusters were found to have much smaller lengthscales than the others, denoting their importance. These consisted of words camera, catch, news, youth, which are related to news reporting; and brum, hood, protect and toward, which broadly relate to the England riots, the central theme across our rumour corpus.

In Figure 4 we show results for three selected methods and the Oracle. We can see that the minimum accuracy across the rumours is the highest for GPC, confirming their robustness in LPO setting.

Overall the MR technique benefits from the additional supervision in the LPO setting (compare MR-LOO and MR-LPO in Figure 4), however note that performance drops in the case of the army bank and police beat girl rumours. This can be explained by temporal effects. Namely, in the case of army bank, the first 50% of tweets comprise of 46 positives and 43 negatives, whereas the last 50% comprise of 16 positives and 72 negatives. Therefore, the classifier wrongly learns from the first half that positives are the majority, leading to poor predictions on the test partition where this is not the case. This way, rumour non-stationarity is a challenge that makes for interesting research.

8 Conclusions

In this paper we analyzed the problem of estimating collective judgements of rumours in social media in both unsupervised (LOO) and supervised (LPO) domain adaptation settings. We found that in case of LOO, employing multi task learning is necessary to achieve the Oracle prediction which returns most frequent label from the test set. Inspection of learnt weight vectors revealed that general terms used to indicate user opinions about rumours can be automatically learned from data. However, without access to annotated examples of the target rumour the learning problem becomes very difficult. Allowing some supervised setting (LPO) helped to address this problem, however minor differences remained due to non-stationarity of community opinions.

In the supervised setting Gaussian Process Classification yielded the best results, beating competitive baselines. Inspection of the learned coregionalization matrix revealed that the correlation between our outlier rumour (Miss Selfridge’s) and the other rumours was near zero, while learned correlations between the other rumours was quite high. Therefore the model managed to learn to exploit similarities and differences between the rumours, enabling transfer learning only when justified from the data.

Apart from working on rumour non-stationarity, in future work we plan to explore how non-textual features contribute to rumour modelling. For example, rumours exhibit diffusion patterns over social network graphs, which may be a useful cue for judgement estimation.

---

5 The letter ‘c’ is a result of tokenisation errors, mainly relating to a media site.

6 An informal name for Birmingham.
References

A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Pas-sonneau. Sentiment analysis of twitter data. In Pro-ceedings of the Workshop on Languages in Social Me-dia, LSM ’11, pages 30–38, Stroudsburg, PA, USA, 2011.

M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. Found. Trends Mach. Learn., 4(3):195–266, Mar. 2012.

D. Beck, T. Cohn, and L. Specia. Joint emotion analysis via multi-task gaussian processes. In Conference on Empirical Methods in Natural Language Processing, EMNLP ’14, pages 1798–1803, Doha, Qatar, 2014.

C. Castillo, M. Mendoza, and B. Poblete. Predicting information credibility in time-sensitive social media. Internet Research, 23(5):560–588, 2013.

T. Cohn and L. Specia. 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, Sofia, Bulgaria, 2013.

T. Evgeniou and M. Pontil. Regularized multi–task learning. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’04, pages 109–117, 2004.

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. J. Mach. Learn. Res., 6:615–637, Dec. 2005.

A. Friggeri, L. Adamic, D. Eckles, and J. Cheng. Rumor cascades. In International AAAI Conference on Weblogs and Social Media, 2014.

S. Kate, M. Jim, O. Mania, A. Peg, and M. Robert. Rumes False Flags and Digital Vigilantes: Misinformation on Twitter after the 2013 Boston Marathon Bombing. In iConference 2014 Proceedings, pages 654–662, 2014.

V. Lampos, N. Aletras, D. Preotiuc-Pietro, and T. Cohn. 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, Gothenburg, Sweden, 2014.

C. Llewellyn, C. Grover, J. Oberlander, and E. Klein. Reusing 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, Reykjavik, Iceland, May 2014.

G. Lotan, E. Graeff, M. Ananny, D. Gaffney, I. Pearce, and danah boyd. The Arab spring— the revolutions were tweeted: Information flows during the 2011 Tunisian and Egyptian revolutions. International Journal of Communication, 5(0), 2011.

M. Mendoza, B. Poblete, and C. Castillo. Twitter under crisis: Can we trust what we RT? In 1st Workshop on Social Media Analytics, SOMA’10, pages 71–79, Washington, DC, USA, 2010.

T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, UAI’02, pages 352–359, San Francisco, CA, USA, 2002.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems, pages 849–856. MIT Press, 2001.

R. Procter, J. Crump, S. Karstedt, A. Voss, and M. Canti-joch. Reading the riots: What were the police doing on twitter? Policing and society, 23(4):413–436, 2013.

V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP ’11, pages 1589–1599, Edinburgh, United Kingdom, 2011.

C. E. Rasmussen and C. K. I. Williams. Gaussian Pro-cesses for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, 2005.

J. Ratkiewicz, M. Conover, M. Meiss, B. Gon alves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media. In 5th Interna-tional AAAI Conference on Weblogs and Social Media, ICWSM’11, Barcelona, Spain, 2011.

J. Shi and J. Malik. Normalized cuts and image segmenta-tion. IEEE Trans. Pattern Anal. Mach. Intell., 22(8): 888–905, Aug. 2000.

A. Zubiaga, M. Liakata, R. Procter, K. Bontcheva, and P. Tolmie. Towards detecting rumours in social media. In AAAI Workshop on AI for Cities, 2015.