D3.3.1 Tools for mining non-stationary data - v2 Clustering models for discovery of regional and demographic variation - v2

Daniel Preoţiuc-Pietro, University of Sheffield
Dr. Sina Samangooei, University of Southampton
Andrea Varga, University of Sheffield
Douwe Gelling, University of Sheffield
Dr. Trevor Cohn, University of Sheffield
Prof. Mahesan Niranjan, University of Southampton

Abstract.
FP7-ICT Strategic Targeted Research Project (STREP) ICT-2011-287863 TrendMiner
Deliverable D3.3.1 (WP3)

Keyword list: bilinear model, Latent Dirichlet Allocation, Dirichlet multinomial Regression, periodicities, clustering, Gaussian Processes, RBF, regional models, temporal models

Project	TrendMiner No. 287863
Delivery Date	April 30, 2014
Contractual Date	April 30, 2014
Nature	Other
Reviewed By	Paul Ringler, Thiery Declerk
Web links	`https://github.com/andreavarga/trendminer-sptempclustering`
Dissemination	PU

TrendMiner Consortium

This document is part of the TrendMiner research project (No. 287863), partially funded by the FP7-ICT Programme.

DFKI GmbH
Language Technology Lab
Stuhlsatzenhausweg 3
D-66123 Saarbrcken
Germany
Contact person: Thierry Declerck
E-mail: declerck@dfki.de

University of Southampton
Southampton SO17 1BJ
UK
Contact person: Mahensan Niranjan
E-mail: mn@ecs.soton.ac.uk

Internet Memory Research
45 ter rue de la Rvolution
F-93100 Montreuil
France
Contact person: France Lafarges
E-mail: contact@internetmemory.org

Eurokleis S.R.L.
Via Giorgio Baglivi, 3
Roma RM
00161 Italy
Contact person: Francesco Bellini
E-mail: info@eurokleis.com

Universidad Carlos III of Madrid
Av. de la Universidad, 30,
28911, Madrid
Spain
Contact person: Paloma Martinez
E-mail: pmf@inf.uc3m.es

Instytut Podstaw Informatyki Polskiej Akademii Nauk
ul. Ordona 21,
01-237 Warszawa,
Poland
Contact person: Maciej Ogrodnicuk
E-mail: maciej.ogrodniczuk@gmail.com

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Contact person: Kalina Bontcheva
E-mail: K.Bontcheva@dcs.shef.ac.uk

Ontotext AD
Polygraphia Office Center fl.4,
47A Tsarigradsko Shosse,
Sofia 1504, Bulgaria
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Sora Ogris and Hofinger GmbH
Bennogasse 8/2/16
AT-1080 Wien
Austria
Contact person: Christoph Hofinger
E-mail: ch@sora.at

Hardik Fintrade Pvt Ltd.
227, Shree Ram Cloth Market,
Opposite Manilal Mansion,
Revdi Bazar, Ahmedabad 380002
India
Contact person: Suresh Aswani
E-mail: m.aswani@hardikgroup.com

Department of Corpus Linguistics of the Hungarian Academy of Sciences
Benczùr u. 33,
068 Budapest VI.,
Hungary
Contact person: Tamás Váradi
E-mail: tavaradi@gmail.com

Daedalus Data, Decision and Language, S.A.
C/ Lòpez de Hoyos 15,
28006 Madrid,
Spain
Contact person: Josè Carlos Gonzales
E-mail: jgonzalez@daedalus.es

Executive Summary

This document presents advanced research and software development work for Task 3.2 on tools for mining non-stationary data and for Task 3.3 on clustering models integrating regional and demographic information for the aim of understanding streaming data.

First, for modelling non-stationary data, a research experiment is presented for categorising and forecasting word frequency patterns using Gaussian Processes, with an emphasis on word periodicities. A new soft clustering method based on topic models is introduced, which learns topics and their temporal profile jointly.

For using regional and demographic user information, the predictive model presented in previous work (Samangooei et al., 2013) is extended. This is used to identify differences in voting intention between different regions of the United Kingdom and different genders. For discovering specific regional clusters, the soft clustering technique is extended to learn the topics, their regional and temporal profile jointly.

Finally, the predictive and clustering models developed on social media data are applied to a news summary dataset where richer linguistic features are also used.

Contents

1 Introduction **2**
- 1.1 Relevance to TrendMiner . . . 3
  - 1.1.1 Relevance to project objectives . . . 3
  - 1.1.2 Relation to other workpackages . . . 3

2 Non-stationary models **4**
- 2.1 Temporal topic models . . . 4
  - 2.1.1 Methods . . . 5
  - 2.1.2 Data . . . 8
  - 2.1.3 Results . . . 8
- 2.2 Periodicities . . . 11
  - 2.2.1 Gaussian Processes . . . 11
  - 2.2.2 Data . . . 14
  - 2.2.3 Forecasting hashtag frequency . . . 15

3 Discovering regional and demographic variation **19**
- 3.1 Regional and demographic bilinear model . . . 19
  - 3.1.1 Bilinear model using user region and demographic features . . . 20
  - 3.1.2 Experiments . . . 23
- 3.2 Regional topic models . . . 26
  - 3.2.1 Methods . . . 26
  - 3.2.2 Results . . . 28

4 Application to news media **31**
- 4.1 Experimental setup . . . 31
- 4.2 Forecasting socioeconomic indicators . . . 32
  - 4.2.1 Experiments . . . 32
  - 4.2.2 Using rich text features . . . 35
- 4.3 Temporal and regional variation . . . 37

5 Conclusions **40**

A City list **42**

Chapter 1

Introduction

Large scale streaming user-generated text varies due on a number of factors, such as time, location and user properties. Better representation of this information can help achieve a deeper understanding of streaming data. A principled integration of the document timestamp information can afford a better understanding of what text data contains and how it evolved in time. By creating richer predictive models which explicitly integrate text metadata, namely spatial and user demographic information, we can learn how text from different sub-groups of users pertain to a real-world outcome (e.g. political indicators or economic signals).

In this document we introduce new methods, extensions to previously introduced models and further experiments for the general goal of discovering temporal, spatial and demographic variation in text. We present two methods which explicitly use time information to exploit different types of non-stationarity in social media data. We also introduce a generic metadata extension of the bilinear predictive models, showing how this approach can help integrate regional and demographic user information into the discovery of relevant terms and users for the prediction of a real valued real-world outcome. Finally, further experiments are presented for demonstrating the efficacy of our predictive and topic detection models to news media sources, going beyond their initial application to social media data sources.

This document contains the following work:

1. An extension of the Dirichlet-multinomial regression topic model (Mimno and McCallum, 2008) used to learn a soft clustering of words in a collection of documents jointly with their temporal profiles and regional metadata, imposing a smoothness constraint over these;

2. A method to model and classify temporal patterns of word frequencies using Gaussian Processes;

3. An extension of the bilinear predictive models introduced in D3.1.2 (Samangooei et al., 2013) which incorporates regional and demographic information;

4. Experiments with both spatio-temporal topic models and bilinear models on news media;
5. Use of rich features derived from WP2 in predictive experiments.

1.1 Relevance to TrendMiner

Text in large user-generated collections is inherently dependent on multiple factors, such as the time, and regional and demographic user information. The work and software presented in this section allow similar exploitations as those released in previous deliverables, but by taking into consideration these important factors.

1.1.1 Relevance to project objectives

The techniques and software developed as part of this deliverable present methods of integrating regional, temporal and demographic information for discovering underlying topics in large collections and for predicting future text or real-world outcomes. Experiments conducted on a novel news summaries dataset highlight the general applicability of the methods developed.

1.1.2 Relation to other workpackages

The ability to detect topics conditioned on temporal, regional and demographic information are useful for the summarisation methods developed in WP4 and the visualisations from WP5. The ability to explore important words and users based on particular user metadata are also helpful for these WPs. All of these techniques can aid the use case workpackages (WP6, WP7, WP10) in order to better contextualise text data and its relation to their problem domains. Deeper linguistic inputs obtained from WP2 have been studied as features in predictive models in order to improve both performance and interpretability.

Chapter 2

Non-stationary models

Social media text has shown to be indicative of real-world activity. Hence, social media text properties, such as frequency, are expected to change over time leading to data non-stationarity. In this section, we present models which exploit different types of non-stationarity in social media data. First, we present a model which learns topics jointly with their temporal dynamics, which can include multiple modes. This model is based on temporal smoothness wherein data authored at a specific time interval smoothly influences data written at neighbouring time intervals. Finally, we introduce a way of identifying and modelling complex temporal dynamics, such as periodicities, in social media text frequencies over time. Applications include identifying topics in tweets written in German and predicting the frequency of hashtags up to one month in the future.

2.1 Temporal topic models

In this section we introduce an unsupervised method for learning the *topics* in a collection of documents together with their temporal profiles. In contrast to the previous approach described in D3.2.1 (Preoţiuc-Pietro et al., 2013), the membership of a word to a topic is probabilistic (soft clustering), which allows relevant words to belong to multiple topics. Another distinctive propriety is that the model learns the entire temporal profile of a topic jointly with the word-topic membership. In order to achieve this we employ Dirichlet-multinomial regression (DMR) (Mimno and McCallum, 2008), which conditions the topic distribution within a document on document-level metadata. We extend the model to use Radial Basis Function (RBF) kernels for modelling the temporal smoothness bias between neighbouring time intervals. Experiment on tweets extracted from throughout Germany and Austria for an entire year show the benefit of conditioning on temporal features.

2.1.1 Methods

Topic models (Blei et al., 2003) are generative probabilistic models which learn soft clusters of words, called *topics*, that describe a document collection. Each document is represented by a mixture of these topics. Due to the dimensionality reduction they perform, topic models have been widely used as a method to summarise and browse document collections. Latent Dirichlet Allocation (LDA) (Blei et al., 2003), the most popular topic model, is an unsupervised methods which uses only document word co-occurrence information to group words into topics.

However, social media text is often characterised by highly temporal dependencies and further, by additional factors such as location as will be investigated in Section 3.2. For example, it is likely that during a large scale event, such as the European Football Championship in 2012 (#euro2012, #em12), a large portion of tweets discuss this topic. However, except for the month the competition took place, tweets on this topic are less frequent. On a more persistent topic, such as the Bundesliga (i.e. the German football championship), tweets are authored consistently throughout the year, with the exception of the summer and winter breaks.

This section presents a method to incorporate such temporal document metadata features into topic models. We aim to learn topics jointly with time in order to better analyse the topics and their temporal profile.

Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation models each document as a mixture of latent topics, where each topic consists of a probability distribution over a fixed set of words. The document-topic and word-topic probabilistic memberships are learnt jointly over a document collection.

The LDA model can be described by the following generative process:

1. For each topic t
   (a) Draw $\phi_t \sim \text{Dirichlet}(\beta)$
2. For each document d
   (a) Draw $\theta_d \sim \text{Dirichlet}(\alpha)$
   (b) For each word w_i
      i. Draw $z_i \sim \text{Multinomial}(\theta_d)$
      ii. Draw $w_i \sim \text{Multinomial}(\phi_z)$

where α is the Dirichlet prior for the document-topic distributions, θ_t refers to the document-topic distribution for topic t, w_i denotes the token at the position i in document d, β is the topic distribution Dirichlet prior.

The plate diagram of LDA is presented in Figure 2.1. T denotes the number of topics, D the number of documents, N_d the number of tokens in document d.

One of the key assumptions of LDA is that words in a document are *exchangeable* i.e. their order is irrelevant. This is equivalent to the bag-of-words assumption for word features inside a document. Exchangeability also holds for documents in the collection. The distribution of topics in documents is independent of any feature. However, conditioned on document metadata (e.g. time), the topics and their mixture are expected to be different. In the next section, we present Dirichlet-multinomial regression (DMR) which can make use of any metadata derived features, in our case, the timestamps of the documents. We further introduce a temporal smoothness bias in order to encode the intuition that similar timestamps have similar properties.

Dirichlet-multinomial regression (DMR)

In order to integrate document level observations we use Dirichlet-multinomial regression (Mimno and McCallum, 2008). This is an upstream topic models which can incorporate arbitrary types of features. The upstream models condition on the observations in order to generate the topic distribution of a document. Alternative downstream models aim to generate both the words and the metadata given the latent topic variable.

The DMR model can be described by the following generative process:

1. For each topic t
   (a) Draw $\boldsymbol{\lambda}_t \sim \mathcal{N}(\mu, s^2 I)$
   (b) Draw $\phi_t \sim \text{Dirichlet}(\beta)$
2. For each document d
   (a) For each topic t
      i. $\alpha_{d,t} = \exp(\boldsymbol{x}_d^T \boldsymbol{\lambda}_t)$
   (b) Draw $\theta_d \sim \text{Dirichlet}(\boldsymbol{\alpha}_d)$
   (c) For each word w_i
      i. Draw $z_i \sim \text{Multinomial}(\theta_d)$
      ii. Draw $w_i \sim \text{Multinomial}(\phi_z)$

In addition to LDA, we introduce the values μ, s^2 as the mean and variance of the Normal prior on metadata features. For each topic t we have a vector $\boldsymbol{\lambda_t}$ of length F (number of features). The observed metadata features for each document is presented in the form of a feature vector $\boldsymbol{x}_d$ with F elements. Details on how this vector is constructed are presented in the following sections. The plate diagram of DMR is presented in Figure 2.2.

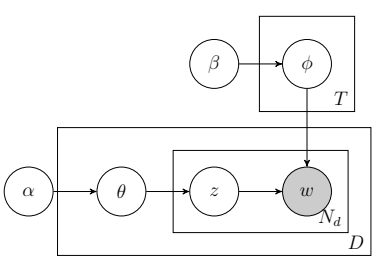


Figure 2.1: The Latent Dirichlet Allocation (LDA) topic model.

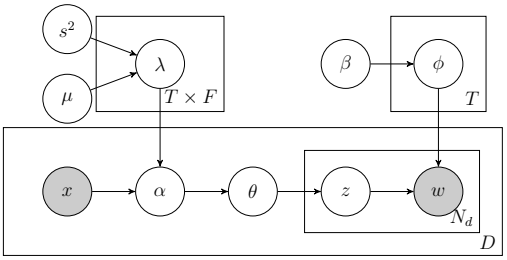


Figure 2.2: The Dirichlet-multinomial Regression (DMR) topic model.

The DMR model is trained using stochastic Expectation Maximisation (EM) sampling, which alternates between a sampling step (using Gibbs sampling) for topic assignments and a optimisation of $\boldsymbol{\lambda}$ (using L-BFGS – Limited-memory Broyden-Fletcher-Goldfarb-Shanno) (Liu and Nocedal, 1989) given the topic assignments.

In summary, DMR assumes that the documents are generated by a latent mixture of topics. The prior distribution over topics is a function of observed document features. Thus, documents having the same metadata values are more likely to share similar topics. Because the document timestamps are continuous, we experiment with different ways of representing the time of a document ($\boldsymbol{x}_d$):

Modelling time with indicator features (MId) This representation captures our natural intuition to group documents in a specific time interval (e.g. month) in which the documents have been authored. Thus, the documents which were authored in the same month are biased to share similar topics. This feature consists of a collection of monthly indicator features spanning over the time frame of the documents with a feature with value 1 on the month the document was authored.

Modelling time with RBF kernels (TimeRBF) By grouping documents based on their time interval we lose any sharing of information between neighbouring time intervals. For example, a document authored at the start of a given time interval shares the same metadata feature with a document authored at the end of the time interval. In contrast, it does not share the same value as the document written at the end of the previous interval, although temporally they are much closer. We should thus model the temporal continuity and add a temporal smoothing effect, where documents of a given timestamp influence documents with similar timestamps.

For this purpose we employ the radial basis function (RBF) kernels in order to transform the data. The RBF kernels are well known in NLP and they are usually employed for regression and classification tasks using non-linear methods (e.g. Support Vector Machines or Gaussian Processes). For a particular document published at time t, and an RBF

centred in t_c, the kernel captures the distance between the two time points:

$$\text{RBF}_{\text{time}}(t, t_c) = \exp\left(\frac{-(t - t_c)^2}{2\sigma^2}\right) \tag{2.1}$$

The RBF function allows a non-linear dependency representing a Gaussian distribution with *mean* t_c and *variance* σ^2. Considering the mean of each RBF kernel fixed, varying variance of this distribution would allow us to control the amount of influence between neighbouring time intervals.

Similarly to the monthly indicator features, we created a set of RBF kernels spanning the timeframe in which the documents were written. Given the kernels, each document's timestamp will be mapped using the kernels into a set of continuous values depending on the dataset.

2.1.2 Data

For the experiments, we have gathered a Twitter dataset ($\mathcal{D}_1$) consisting of 42,802,603 tweets. This was collected using the Twitter Search API from across 37 of the most populous cities in Austria and Germany (full list in Appendix 5). This mostly consists of tweets written in German. The data collection interval was 14/06/2012 to 11/06/2013. We have kept a vocabulary of 41,555 most frequent well-formed words and eliminating the most frequent 1000 words. Because of the limited context contained in each tweet, we have merged into the same document all tweets authored on the same day from a specific location. This is a common practice when using tweets.

2.1.3 Results

We now present experimental results for topic modelling on the $\mathcal{D}_1$ dataset with an emphasis of time modelling. Experiments were conducted using the open-source MALLET toolkit.[1]

Quantitative results

In order to quantitatively asses the performance of topic models, we measure perplexity of the trained topic models on held out data. Perplexity measures the ability of the model to predict the contents of documents. A lower perplexity indicates that a model is able to better predict the content of a document, given its metadata (i.e. time in this experiments)

[1] `http://mallet.cs.umass.edu/`

and. The perplexity on a set of documents $\mathcal{D} = \{D_i\}_{i=1}^n$ is defined as follows:

$$\text{perplexity}(\mathcal{D}) = \exp\left(-\frac{\sum_{i=1}^n \log p(D_i|\phi)}{\sum_{i=1}^n N_i}\right) \tag{2.2}$$

where N_i is the number of tokens in document D_i. The likelihood $p(D_i|\phi)$ is intractable and generally a sampling method is used to approximate it (Wallach et al., 2009).

We followed the same experimental setup as in (Mimno and McCallum, 2008): we run the sampler for 1000 iterations, set the burn-in to 250, chose 100 as the total number of topics ($T = 100$). We evaluate on a held out set of 10% of the data and average results across 3 different runs. All the experiments are run with $T = 100$, a number recommended based on topic coherence by Stevens et al. (2012). The results comparing the methods are presented in Table 2.1.

Method	**Perplexity**
LDA	12,777.19
DMR MId	12,497.33
DMR TimeRBF ($\sigma = 20$)	12,412.40

Table 2.1: Perplexity on held out data for Tweet dataset $\mathcal{D}_1$. Lower is better.

Foremost, we notice that incorporating time features into the topic model consistently and significantly improves perplexity of the model. Secondly, we find that adding the temporal smoothing constraint using the RBF kernel (TimeRBF), we obtain further performance improvements over MId. The DMR TimeRBF performance presented in Table 2.1 shows the best results obtained by tuning the σ parameter. The plot showing the perplexity for various values of σ is presented in Figure 2.3. We use as centers for the RBF functions the middle of each month. Experiments have shown no significant changes when the RBF centres were different.

Qualitative results

In this section we present qualitative results of the topic models. We selected six topics from the best performing model (DMR TimeRBF $\sigma = 20$) and we present them by showing their top 10 most representative words and their temporal profile in Figure 2.4. The temporal profile is computed by passing the learnt magnitudes through each kernel, thus obtaining a mixture of RBFs with a smooth, possibly multi-modal shape.

Figure 2.4 shows different temporal patterns for topics. Topic #68 presents a topic which shows no important temporal changes throughout the year long period we study. Topics #0 and #73 show topics which see periods of slightly lower prevalence alternated with periods with a higher prevalence. For example, Topic #0 is mainly about football

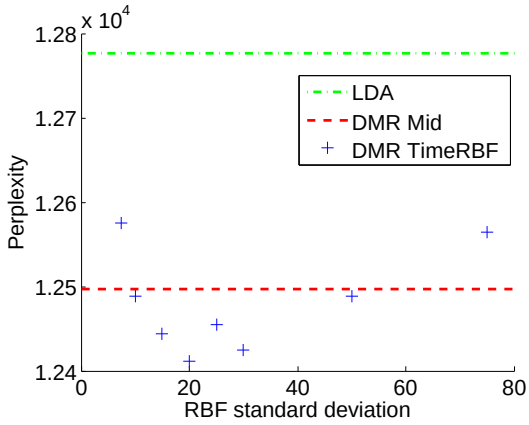


Figure 2.3: Results for different values of RBF standard deviation values compared to the baseline models. Lower perplexity is better.

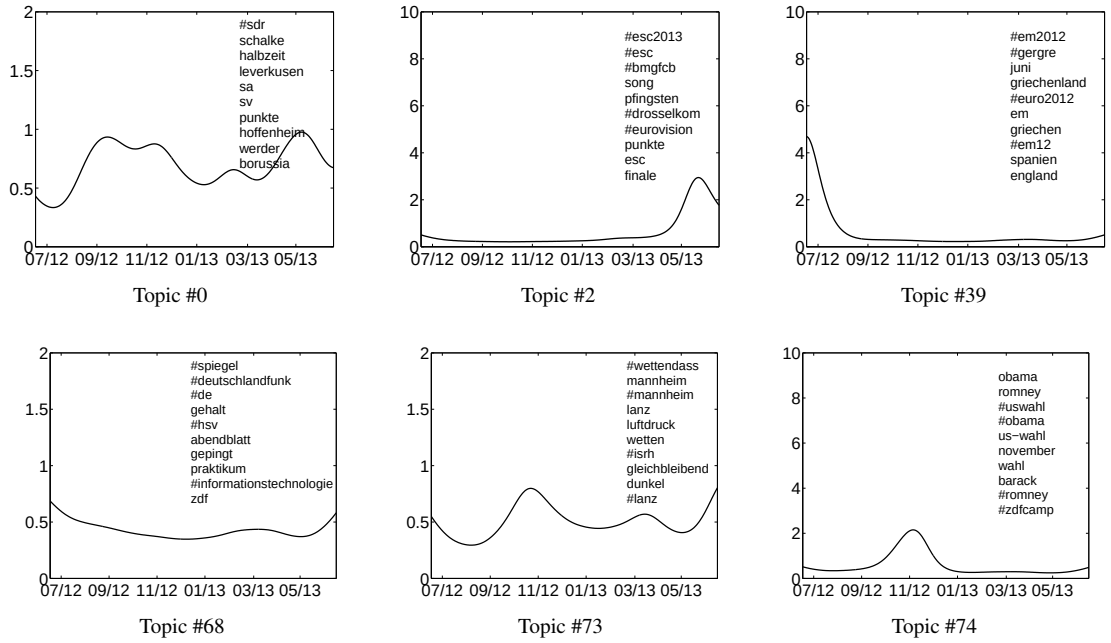


Figure 2.4: Sample topics (represented by top 10 words) and their temporal profile learnt using the DMR Time RBF model.

teams, having the lowest magnitude in the month of August 2012 when the football competitions are on summer break. We also distinguish topics which have a burst of importance around a date (Topics #2, #39, #74). These are topics about events that are very time sensitive and about which people talk about only for a restricted period of time. For example, Topic #2 is about the Eurovision Song Contest which took place between 14-18 May 2013, while Topic #39 is about the European Football Championship (EURO 2012) which took place throughout June 2012. Topic #74 is about the US presidential elections.

Here we notice that the topic has slightly higher prevalence in the months running up to the election (6 November 2013) than in the months after the election.

2.2 Periodicities

For modelling non-stationarity we have incorporated into the previous novel models the intuition that data proprieties are smoothly influenced by data from neighbouring time intervals. While this is a reasonable assumption when modelling temporality, these methods cannot capture more complex and long term temporal dependencies such as periodic rise and fall. In this section we present a research experiment which models and classifies word time series automatically using Gaussian Processes (GP) for the first time in the NLP literature. The experimental task is supervised regression, where we use known word frequencies for a given interval and aim to predict future word frequencies in a future time interval. Most of this section has been published in (Preoţiuc-Pietro and Cohn, 2013).

2.2.1 Gaussian Processes

GPs are a probabilistic machine learning framework incorporating kernels and Bayesian non-parametrics which is widely considered as state-of-the-art for regression. Consider a time series regression task where we only have one feature, the value x_t at time t. Our training data consists of n pairs $\mathcal{D} = \{(t, x_t)\}$. The model will need to predict values x_t for values of t not in the dataset.

GP regression assumes a latent function f that is drawn from a GP prior $f(t) \sim \mathcal{GP}(m, k(t, t'))$ where m is the mean and k a kernel. The prediction value is obtained by the function evaluated at the corresponding data point, $x_t = f(t) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is white-noise. The GP is defined by the mean m, here 0, and the covariance kernel function, $k(t, t')$. The kernel specifies the covariance between pairs of outputs:

$$k(t, t') = \text{cov}(x_t, x'_t) \tag{2.3}$$

The posterior at a test point t_* is given by:

$$p(x_*|t_*, \mathcal{D}) = \int_f p(x_*|t_*, f) \cdot p(f|\mathcal{D}) \tag{2.4}$$

where x_* and t_* are the test value and time. The posterior $p(f|\mathcal{D})$ shows our belief over possible functions after observing the training set $\mathcal{D}$. The predictive posterior can be solved analytically with solution:

$$x_* \sim \mathcal{N}(k_*^T (K + \sigma_n^2 I)^{-1} \mathbf{t}, k(t_*, t_*) - k_*^T (K + \sigma_n^2 I)^{-1} k_*) \tag{2.5}$$

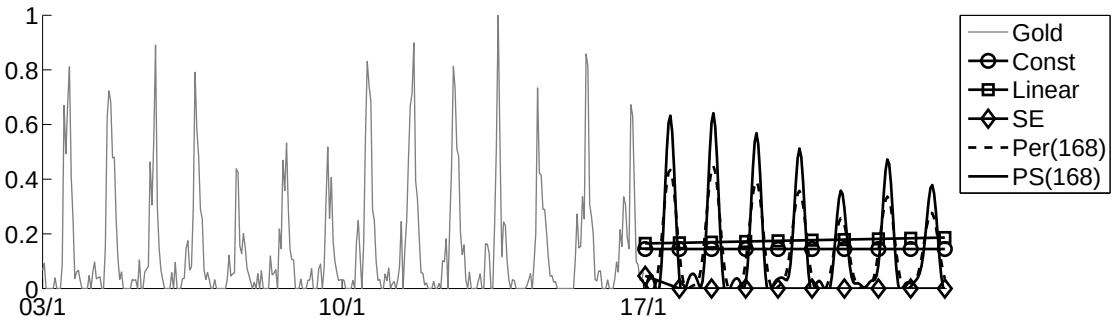


Figure 2.5: Extrapolation for #goodmorning over 3 weeks with GPs using different kernels.

where $k_* = [k(t_*, t_1)...k(t_*, t_n)]^T$ are the kernel evaluations between the test point and all the training points, $K = \{k(t_i, t_j)\}_{j=1..n}^{i=1..n}$ is the Gram matrix over the training points and $\mathbf{t}$ is the vector of training points. The posterior of x_* includes the mean response as well as its variance, thus expressing the uncertainty of the prediction. In this section, we will consider the forecast as the expected value.

In our experiments we consider an extrapolation setup where the range of the prediction is outside the training input bounds. Given the covariance is defined over an infinite set of pairs, we need to assume a simple form of the covariance by defining covariance values using a kernel function. The covariance implies a distribution over functions and encodes the properties of the functions (e.g. smoothness, periodicity, stationarity). Intuitively, if the desired function is smooth, closer points should have high covariance compared to points that are further apart. If a periodic behaviour is desired, points at period p distance should have the highest covariance. In extrapolation, the covariance kernel plays a major role in the prediction, incorporating the types of patterns the model aims to model. To illustrate this, in Figure 2.5, we show the time series for #goodmorning over 2 weeks and plot the regression for the future week learnt by using different kernels.

We will use multiple kernels, each most suitable for a specific category of temporal patterns in our data. This includes a new kernel inspired by observed word occurrence patterns. The kernels are:

Constant (Const): The constant kernel is $k_C(t, t') = c$ and it describes a constant relationship between outputs. Its mean prediction will always be the value c learnt in training. Its assumption is that the signal is modelled only by Gaussian noise centred around this value. This describes the data best when we have a noisy signal around a stationary mean value.

Squared exponential (SE): The SE kernel or the Radial Basis Function (RBF):

$$k_{SE}(t, t') = s^2 \cdot \exp\left(-\frac{(t - t')^2}{2l^2}\right) \tag{2.6}$$

This gives a smooth transition between neighbouring points and best describes time

series with a smooth shape e.g. a uni-modal burst with a steady decrease. However, the predictive variance increases exponentially with distance. Predictions well into the future will have no covariance. Its two parameters s and l are the characteristic length-scales along the two axes.

Periodic (PER): The periodic kernel represents a SE kernel in polar coordinates and describes a sinusoidal relationship between outputs:

$$k_{PER}(t,t') = s^2 \cdot \exp \cdot \left(-\frac{2\sin^2(2\pi(t-t')/p)}{l^2} \right) \tag{2.7}$$

The kernel is good at modelling periodically patterns that oscillate smoothly between low and high frequency. s and l are characteristic length-scales as in the SE kernel and p is the period i.e. distance between consecutive peaks.

Periodic spikes (PS): We introduce this kernel in order to model the following periodic behaviour: abrupt periods of high values, usually with a peak, followed by periods of very low occurrence:

$$k_{PS}(t,t') = \cos\left(\sin\left(\frac{2\pi\cdot(t-t')}{p}\right)\right) \cdot \exp\left(\frac{s\cos(2\pi\cdot(t-t'))}{p} - s\right) \tag{2.8}$$

This is inspired by observations on studying word frequencies for which low values can be observed for short or long time intervals, followed by abrupt periodic rise in usage. For example, words associated with a weekly TV series will only have non-zero frequency during and around its air time. Some other words will only be used at close to nighttime and seldom during the rest of the day.

The kernel is parametrised by its period p and a shape parameter s. The period indicates the time interval between the peaks of the function, while the shape parameter controls the *width* of the spike. The behaviour of the kernel is illustrated in Figure 2.9. We constrain $s \geq 1$.

The flexibility of the GP framework allows us to combine kernels (e.g. $SE \cdot PS$ or $PS + Lin$) in order to identify a combination of trends (Duvenaud et al., 2013). Experiments on a subset of data showed no major benefits of combining kernels, but the computational time and model complexity increased drastically due to the extra hyperparameters.

There is the case when multiple covariance kernels can describe our data. For choosing the right kernel only using the training data we employ Bayesian model selection which makes a trade-off between the fit of the training data and model complexity. More details on this method are presented in (Preoţiuc-Pietro and Cohn, 2013). For further presentation to GPs we refer the interested reader to (Rasmussen and Williams, 2005).

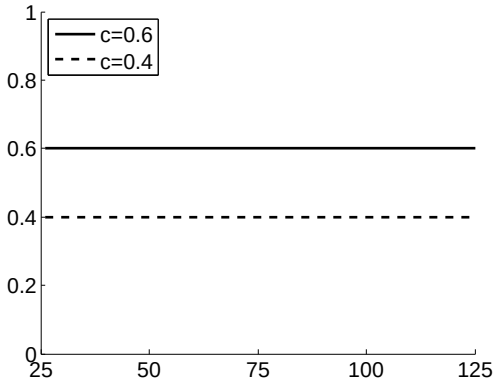


Figure 2.6: Constant kernel.

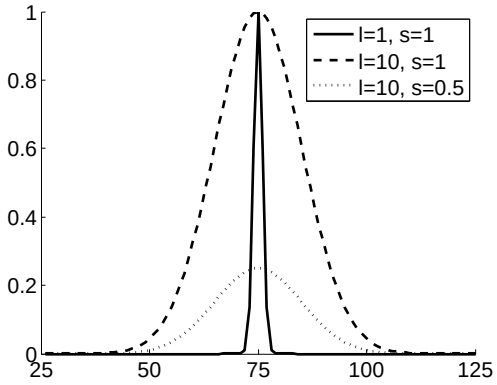


Figure 2.7: Squared exponential kernel.

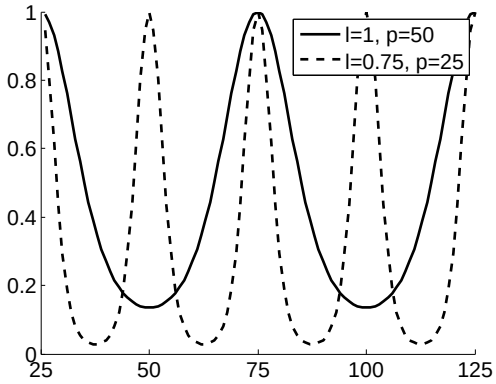


Figure 2.8: Periodic kernel.

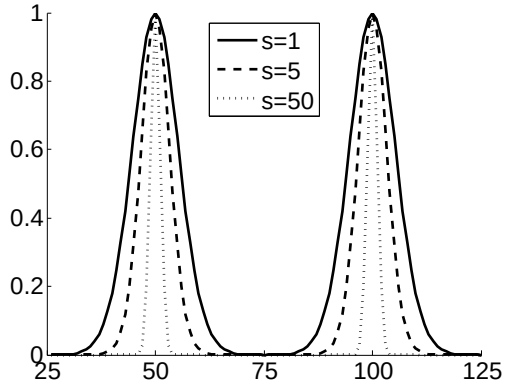


Figure 2.9: PS kernel with p=50.

2.2.2 Data

For our experiments we used data collected from Twitter using the public Gardenhose stream (10% representative sample of the entire Twitter stream). The data collection interval was 1 January – 28 February 2011. For simplicity in the classification task, we filtered the stream to include only tweets that have exactly one hashtag. These represent approximately 7.8% of our stream.

As text processing steps, we have tokenised all the tweets and filtered them to be written in English using the Trendminer pipeline (Preoţiuc-Pietro et al., 2012). We also remove duplicate tweets (retweets and tweets that had the same first 6 content tokens) because they likely represent duplicate content, automated messages or spam which would bias the dataset, as also stated by Tsur and Rappoport (2012). In our experiments we use the first month of data as training and the second month as testing. Note the challenging nature of this testing configuration where predictions must be made for up to 28 days into the future. We keep a total 1176 of hashtags which appear at least 500 times in both splits of the data. The vocabulary consists of all the tokens that occur more than 100 times in the dataset and start with an alphabetic letter. After processing, our dataset consists of

6,416,591 tweets with each having on average 9.55 tokens.

2.2.3 Forecasting hashtag frequency

We treat our task of forecasting the volume of a Twitter hashtag as a regression problem. Because the total number of tweets varies depending on the day and hour of day, we chose to model the proportion of tweets with the given tag in that hour. Given a time series of these values as the training set for a hashtag, we aim to predict the values in the testing set, extrapolating to the subsequent month.

Hashtags represent free-form text labels that authors add to a tweet in order to enable other users to search them to participate in a conversation. Some users use hashtags as regular words that are integral to the tweet text, some hashtags are general and refer to the same thing or emotion (#news, #usa, #fail), others are Twitter games or memes (#2010dissapointments, #musicmonday). Other hashtags refer to events which might be short lived (#worldcup2022), long lived (#25jan) or periodic (#raw, #americanidol). We chose to model hashtags because they group similar tweets (like topics), reflect real world events (some of which are periodic) and present direct means of evaluation. Note that this approach could be applied to many other temporal problems in NLP or other domains. We treat each regression problem independently, learning for each hashtag its specific model and set of parameters.

Hashtag	**Lag+**	**Const**		**SE**		**PER**		**PS**	
	NRMSE	NLML	NRMSE	NLML	NRMSE	NLML	NRMSE	NLML	NRMSE
#fyi	0.1578	**-322**	**0.1404**	-320	0.1898	-321	0.1405	-293	0.1456
#confessionhour	0.0404	-85	0.0107	**-186**	**0.0012**	-90	0.0327	-88	0.0440
#fail	0.1431	-376	0.1473	-395	0.4695	**-444**	**0.1387**	-424	0.1390
#breakfast	0.1363	-293	0.1508	-333	0.1773	-293	0.1514	**-367**	**0.1276**
#raw	0.0464	-1208	0.0863	-1208	0.0863	-1323	0.0668	**-1412**	**0.0454**

Table 2.2: NRMSE shows the best performance for forecasting and NLML (only using training data) shows the best model for all the regressions in Figure 2.10. Lower is better.

Methods

We choose multiple baselines for our prediction task in order to compare the effectiveness of our approach. These are:

Mean value (M): We use as prediction the mean of the values in the training set. Note that this is the same as using a GP model with a constant kernel (+ noise) with a mean equal to the training set mean.

Lag model with GP determined period (Lag+): The prediction is the mean value in the training set of the values at lag Δ where Δ is the period rounded to the closest integer as determined by our GP model. This is somewhat similar to an autoregressive

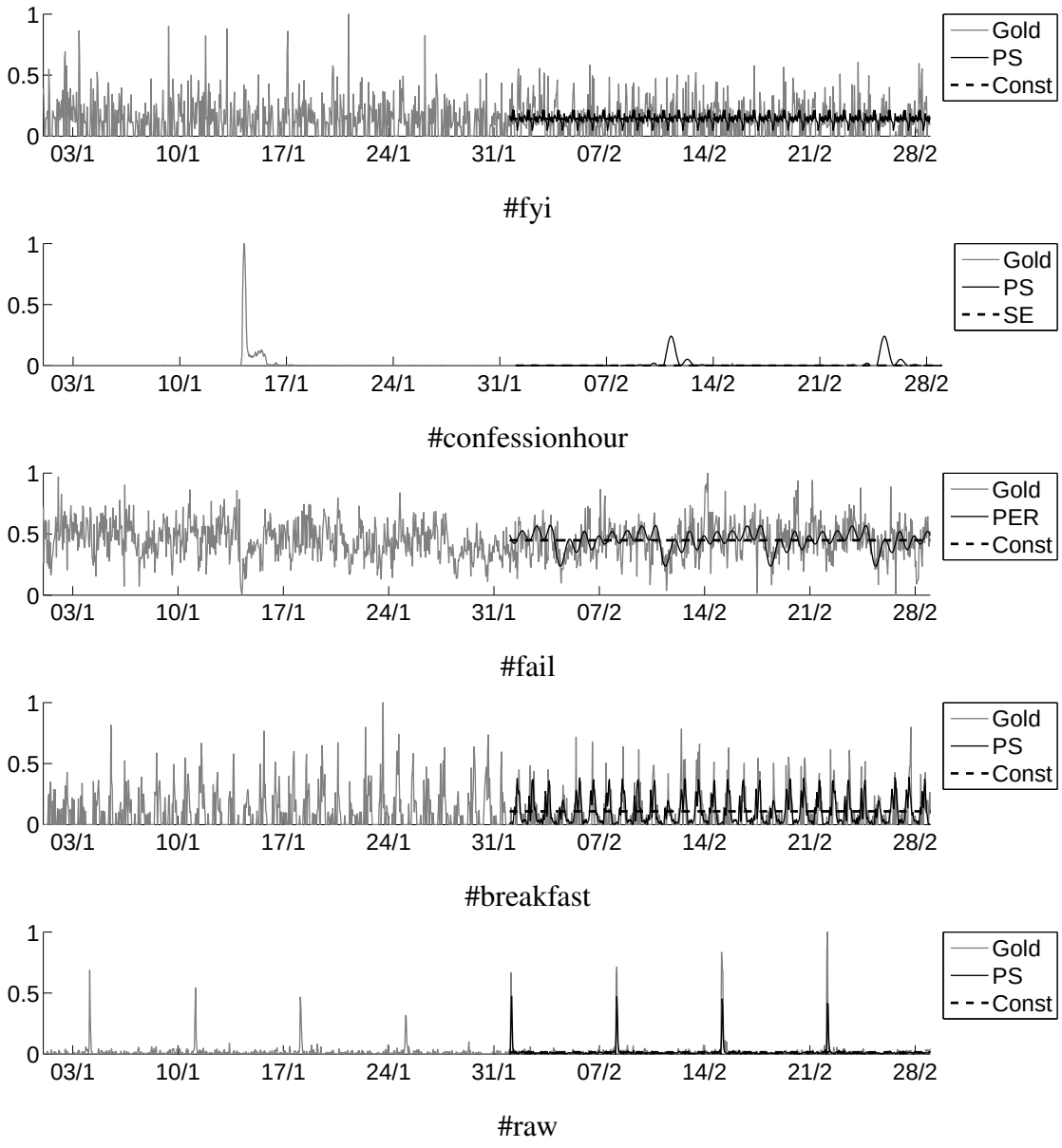


Figure 2.10: Sample regressions and their fit using different methods.

(AR) model with all the coefficients except Δ set to 0. We highlight that given the period Δ this is a very strong model as it gives a mean estimate at each point. Comparing to this model we can see if the GP model can recover the underlying function that described the periodic variation and filter out the noise in the observations. Correctly identifying the period is very challenging as we discuss below.

GP regression: Gaussian Process regression using only the SE kernel (**GP-SE**), the periodic kernel (**GP-PER**), the PS kernel (**GP-PS**). The method that chooses between kernels using model selection as mentioned in Section 2.2.1 is denoted as **GP+**. We will

also compare to GP regression the linear kernel **(GP-Lin)**, but we will not use this as a candidate for model selection due the poor results shown below.

Results

We start by qualitatively analysing a few sample regressions that are representative of each category of time series under study. These are shown in Figure 2.10. For clarity, we only plotted a few kernels on each figure. The full evaluation statistics in NRMSE and the Bayesian evidence are show in Table 2.2.

For the hashtag #fyi there is no clear pattern. For this reason the model that uses the constant kernel performs best, being the simplest one that can describe the data, although the others give similar results in terms of NRMSE on the held out testing set. While functions learnt using this kernel never clearly outperform others on NRMSE on held out data, this is very useful for interpretation of the time series, separating noisy time series from those that have an underlying periodic behaviour.

The #confessionhour example illustrates a behaviour best suited for modelling using the SE kernel. We notice a sudden burst in volume which decays over the next 2 days. This is actually the behaviour typical of 'internet memes' (this hashtag tags tweets of people posting things they would never tell anyone) as presented in (Yang and Leskovec, 2011). These cannot be modelled with a constant kernel or a periodic one as shown by the results on held out data and the time series plot. The periodic kernels will fail in trying to match the large burst with others in the training data and will attribute to noise the lack of a similar peak, thus discovering wrong periods and making bad predictions. In this example, forecasts will be very close to 0 under the SE kernel, which is what we would desire from the model.

The periodic kernel best models hashtags that exhibit an oscillating pattern. For example, this best fits words that are used frequently during the day and less so during the night, like #fail. Here, the period is chosen to be one week (168) rather than one day (24) because of the weekly effect superimposed on the daily one. Our model recovers that there is a daily pattern with people tweeting about their or others' failures during the day. On weekends however, and especially on Friday evenings, people have better things to do.

The PS kernel models best hashtags that have a large and short lived burst in usage. We show this by two examples. First, we choose #breakfast which has a daily and weekly pattern. As we would expect, a big rise in usage occurs during the early hours of the day, with very few occurrences at other times. Our model discovers a weekly pattern as well. This is used mainly for modelling the difference between weekends and weekdays. On weekends, the breakfast tag is more evenly spread during the hours of the morning, because people do not have to wake up for work and can have breakfast at a more flexible time than during the week. In the second example, we present a hashtag that is associated to a weekly event: #raw is used to discuss a wrestling show that airs every week for 2

Const	SE	PER	PS
#funny	#2011	#brb	#ff
#lego	#backintheday	#coffee	#followfriday
#likeaboss	#confessionhour	#facebook	#goodnight
#money	#februarywish	#facepalm	#jobs
#nbd	#haiti	#funny	#news
#nf	#makeachange	#love	#nowplaying
#notetoself	#questionsidontlike	#rock	#tgif
#priorities	#savelibraries	#running	#twitterafterdark
#social	#snow	#xbox	#twitteroff
#true	#snowday	#youtube	#ww
49	**268**	**493**	**366**

Table 2.3: Sample hashtags for each category. The last line shows the total number of hashtags of each type.

Lag+	GP-Lin	GP-SE	GP-PER	GP-PS	GP+
7.29%	-3.99%	-34.5%	0.22%	7.37%	**9.22%**

Table 2.4: Average relative gain over mean (M) prediction for forecasting on the entire month using the different models.

hours on Monday evenings in the U.S.. With the exception of these 2 hours and the hour building up to it, the hashtag is rarely used. This behaviour is modelled very well using our kernel, with a very high value for the shape parameter ($s = 200$) compared to the previous example ($s = 11$) which captures the abrupt trend in usage. In all cases, our GP model chosen by the evidence performs better than the Lag+ model, which is a very strong method if presented with the correct period. This further demonstrates the power of the Gaussian Process framework to deal with noise in the training data and to find the underlying function of the time variation of words. In Table 2.3 we present sample tags identified as being part of the 4 hashtag categories, and the total number of hashtags in each.

As a means of quantitative evaluation we compute the relative NRMSE compared to the Mean (M) method for forecasting. We choose this, because we consider that NRMSE is not comparable between regression tasks due to the presence of large peaks in many time series, which distort the NRMSE values. The results are presented in Table 2.4 and show that our Gaussian Process model using model selection is best. Remarkably, it consistently outperforms the Lag+ model, which shows the effectiveness of the GP models to incorporate uncertainty. The GP-PS model does very well on its own. Although chosen in the model selection phase in only a third of the tasks, it performs consistently well across tasks because of its ability to model well all the periodic hashtags, be they smooth or abrupt.

We have also applied this modelling approach to word frequencies in a downstream task, document classification, and shown that it can lead to improvements in accuracy. Experiments are presented in (Preoţiuc-Pietro and Cohn, 2013).

Chapter 3

Discovering regional and demographic variation

The text in streaming user-generated content is influenced by different user-level properties, such as their location and demographics, in addition to the authoring time. Social media allows us the novel opportunity to analyse text usage in context of different user types. Recent work has shown, for example, that language use is different based on user geolocation (Eisenstein et al., 2010) or that sentiment is expressed differently depending on the user's gender (Volkova et al., 2013). In this section, we present models which exploit different types of regional and demographic information in social media data. First, we extend our bilinear predictive framework to include arbitrary user partitioning e.g. based on regions or demographics. Further, we extend the topic models introduced in Section 2.1 to include regional information. Applications include predicting and characterising voting intention in the UK and spatial analysis of tweets written in German.

3.1 Regional and demographic bilinear model

In this section we describe a novel extension of our bilinear model introduced in D3.1.2 (Samangooei et al., 2013) and in (Lampos et al., 2013) which allows for the incorporation of regional and demographic user information. The original bilinear predictive model is formulated as:

$$y = \boldsymbol{u}^{\mathrm{T}} X \boldsymbol{w} + \beta \tag{3.1}$$

where X is a $m \times p$ matrix of user-word frequencies and $\boldsymbol{u}$ and $\boldsymbol{w}$ are learnt parameters representing the predictive weight for users and words respectively. Aiming to learn a sparse set of users and words, our biconvex learning scheme incorporates an elastic-net regulariser (Zou and Hastie, 2005) applied to both $\boldsymbol{u}$ and $\boldsymbol{w}$. This formulation of the model is called the Bilinear Elastic-Net (**BEN**). This model was extended by learning of $\boldsymbol{u}$ and

$\boldsymbol{w}$ across multiple tasks, formulated as:

$$\boldsymbol{y} = \boldsymbol{U}^{\mathrm{T}} X \boldsymbol{W} + \boldsymbol{\beta} \tag{3.2}$$

where X holds user-word frequencies, but $\boldsymbol{U}$ and $\boldsymbol{W}$ are $m \times \tau$ and $p \times \tau$ matrices respectively and $\boldsymbol{y} \in \mathbb{R}^{\tau}$, where τ is the number of tasks for learning. Facilitated by the ℓ_1, ℓ_2 regulariser, a multi-task extension of the group LASSO regulariser (Argyriou et al., 2007), used in our biconvex learning scheme, this formulation selects a sparse set of words and users biasing similar weights for users and words across all tasks. This method is named the Bilinear Group ℓ_1, ℓ_2, or **BGL**.

The general concept of transitioning from BEN to BGL was to bias the tasks to share similar users and words (through the ℓ_1, ℓ_2 regulariser) on the basis that similar words and users are relevant to all tasks, but also allowing for variation. Based on this, we consider adding a new factor into the model, namely metadata about a user. Practically, we consider user features which allow the partitioning of users into disjoint groups (e.g. based on region, age group, gender, social grade) and posit that our model should learn different sets of weights for each user group, however biasing the models to share similar words.

In the rest of this section we describe this extension, detailing the learning scheme as well as what extra information structure they can discover. We apply this for creating regional and demographic (i.e. gender) models, which allow us to explore peculiarities of their correlations to political voting intention. Though we achieve improved predictive ability when incorporating these extra factors, we primarily emphasise the ability to select word and user relevance to each region and demographic group.

3.1.1 Bilinear model using user region and demographic features

In this section we outline the extended bilinear model and associated learning procedure for user and word weights across tasks, while also considering extra user metadata (i.e. regions or demographics). In the following, the model is described in terms of regions, but all the ideas are equally true for any other type of user metadata which can partition the users (e.g. gender, age groups, social grade). Let $\mathcal{Q} \in \mathbb{R}^{n \times \varrho \times m \times p}$ be a tensor which captures our training inputs, where n, ϱ, m and p denote the number of training examples, regions, users and words respectively; $\mathcal{Q}$ can simply be interpreted as $n \times \varrho$ versions of X (denoted by $\mathcal{Q}_{ir}$ in the remainder of the script), a different one for each day and region, put together in a tensor. Each element $\mathcal{Q}_{irjk}$ holds the frequency of term k for user j in the region r during the day i in our dataset. If a user j in region r has posted $c_{irj\cdot}$ tweets during day i, and $c_{irjk} \leq c_{irj\cdot}$ of them contain a term k, then the frequency of the term k for day i, by user k in a region r is defined as $\mathcal{Q}_{irjk} = \frac{c_{irjk}}{c_{irj\cdot}}$. Recall that this extension only allows a user to belong to a single group defined by the metadata, i.e. a single region or a single demographic group. In $\mathcal{Q}$ this phenomena is encoded by having for a user j in region r only non-zero values in the r-th region slice of $\mathcal{Q}$.

To learn user and term weights extended to support user metadata, a global optimisation function similar to BGL can be formulated as:

$$\begin{aligned}\{W^*, U^*, \boldsymbol{\beta}^*\} = \underset{W,U,\boldsymbol{\beta}}{\operatorname{argmin}} & \sum_{t=1}^{\tau}\sum_{r=1}^{\varrho}\sum_{i=1}^{n}\left(\boldsymbol{u}_{tr}^{\mathrm{T}}\mathcal{Q}_{ir}\boldsymbol{w}_{tr} + \beta_{tr} - y_{tir}\right)^2 \\ & + \lambda_1 \sum_{r=1}^{\varrho}\sum_{j=1}^{p}\|U_{rj}\|_2 \\ & + \lambda_2 \sum_{g\in G}\|W_g\|_\infty \end{aligned} \tag{3.3}$$

where $\mathcal{Q}_{ir}$ is defined as previously described, $W = [\boldsymbol{w}_{1,1} \ldots \boldsymbol{w}_{\tau,1} \ldots \boldsymbol{w}_{\tau,\varrho}]$ is the term weight matrix (each $\boldsymbol{w}_{t,r}$ is the t-th task in the r-th region), and equivalently $U = [\boldsymbol{u}_{1,1} \ldots \boldsymbol{u}_{\tau,1} \ldots \boldsymbol{u}_{\tau,\varrho}]$ is the user weight matrix. Unlike both BGL and BEN, the regulariser functions added to this global optimisation function are different for U as compared to W. U_{rj} denotes a single row across all tasks of the user weight tensor U for a user j in a region r. To regularise U, the weights of users j in different regions r are regularised separately from one another using multiple, disjoint ℓ_1, ℓ_2 regularisers as in BGL. This encourages a similar activation of users as in BGL (Lampos et al., 2013), but only between users in the same region. We employ this strategy because users are necessarily associated with a single region, and so there cannot be any shared structure between users. Therefore, the regulariser does not need to allow for a given user's activation affecting another user's activation if those two users are from separate regions.

Contrary to the regularisation of U, the terms weights in W aren't necessarily disjoint across regions. Ideally, terms should affect one another's activation across regions as BGL allows them to do so across tasks, and yet few terms should be selected overall. A group graph regulariser (Mairal et al., 2010) was used to achieve this term selection. G represents the set of groups, each holding a set of indices into the W matrix. We construct G such that:

$$\begin{aligned} G = & \{\{(t,r,k) : \forall t \in [1 \ldots \tau]\} : \forall r \in [1 \ldots \varrho] \ \forall k \in [1 \ldots m]\} \\ & \cup \\ & \{\{(t,r,k) : \forall r \in [1 \ldots \varrho]\} : \forall t \in [1 \ldots \tau] \ \forall k \in [1 \ldots m]\} \end{aligned} \tag{3.4}$$

Here, for a given term k, each cell of a given task t across all regions r form one set of groups $g \in G$, and each cell of a given region r across all tasks t form another set of groups $g \in G$. This results in $|G| = (m \times \varrho + m \times \tau)$ groups. Note that for each of these groups we are computing the ℓ_1, ℓ_∞-norm for the elements of W in their respective group. As a result, we expect to encourage the activation of a sparse set of groups (i.e. terms k) but with non-zero weights for each cell of W in that group (i.e. across the tasks

τ or the regions ϱ). Consequently we are achieving the filtering and weighting effect of BGL, but withing each user group as defined by their metadata. We refer to this method as the Bilinear Group Graph Regulariser (**BGGR**) and is depicted below BGL in Figure 3.1.

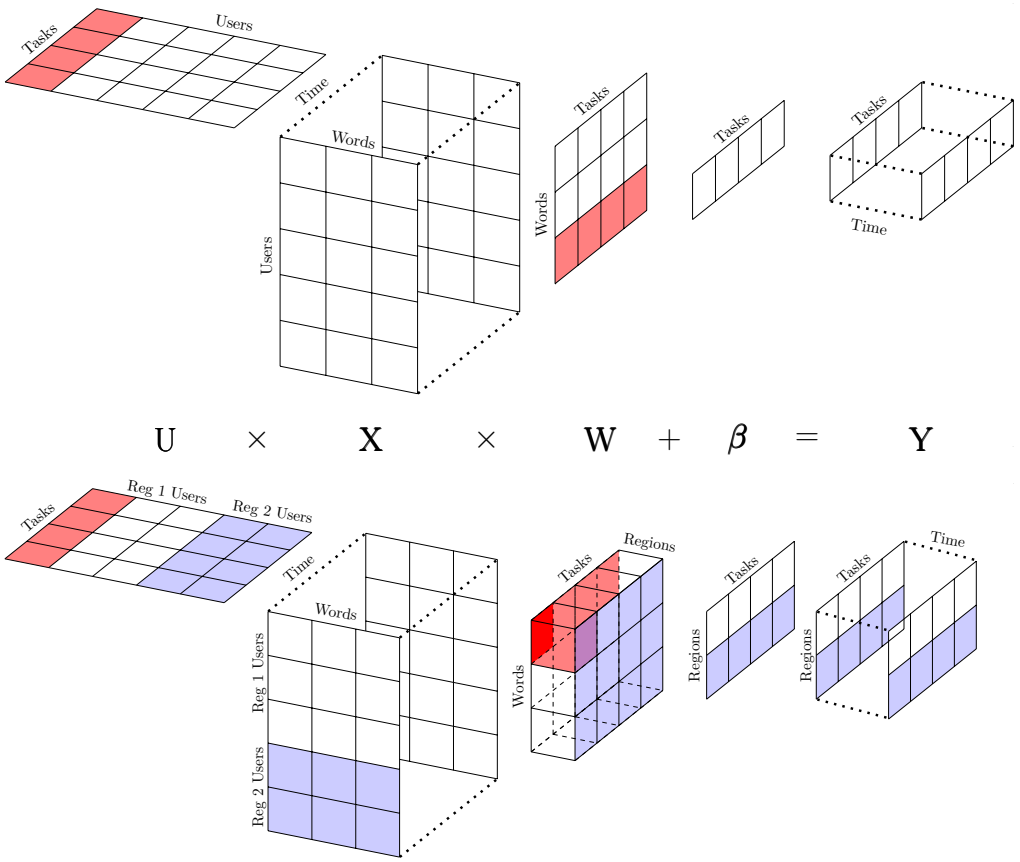


Figure 3.1: Graphical presentation of BGL (top) and BGGR (bottom). Red shows groups over which regularisation is effective, blue shows sub-tensors which are multiplied.

Broadly speaking, Equation 3.3 can be broken into a set of convex tasks as with BGL and BEN. The process of holding W fixed while optimising U and the reverse when optimising W is maintained. Key changes are that in the U step we actually perform ϱ separate optimisations and in the W step careful matrix arrangement must be made in order to correctly match W with the groups in G. More details of these technical details can be found in the released version of the bilinear library.[1]

[1] `https://github.com/sinjax/trendminer-python/tree/master/bivariate`

3.1.2 Experiments

We evaluate our new model using a dataset of tweets from the UK in order to predict UK voting intention. We formulate two sets of experiments based on the user metadata we use, namely gender and region information. We measure the predictive performance of these models by compare it to the ground truth polling data as well as established baselines. We also qualitatively analyse the key terms selected with high weights across different regions.

Dataset

The dataset ($\mathcal{D}_2$) used in the experiments is the same as that used in the previous experiments with the BEN and BGL models in the UK political use case (Samangooei et al., 2013). To form the input data $\mathcal{Q}$, we collected around 60 million tweets produced by approximately 42k UK Twitter users from 30/04/2010 to 13/02/2012. We obtained user metadata information for users by parsing their location field. This was done with the method presented to construct the gold standard dataset in (Rout et al., 2013). Note that the users were selected initially so that they can be mapped to a valid UK location. For identifying the user gender we have used an unpublished text classification method provided by Svitlana Volkova.[2] This information was used to create the two data matrices $\mathcal{Q}_{\text{gender}}$ and $\mathcal{Q}_{\text{region}}$ in the format described in Section 3.1.1.

As with BEN and BGL, the ground truth for training and evaluating our predictive model is formed by voting intention polls provided by YouGov.[3] More specifically, we use voting intentions for the three major parties in the UK, namely Conservatives (CON), Labour (LAB) and Liberal Democrats (LBD). We matched the time span of the Twitter input data, as well as collecting the voting intentions split by the 2 genders and 5 geographic regions for which user metadata was gathered. We collected a time series of 240 voting intention polls, each separated by five geographic regions (South of England – **S**, London – **L**, Midlands and Wales – **M**, North of England – **N**, Scotland – **Sc**) and two genders (Male – **M**, Female – **F**).

Results

To evaluate the capability BGGR to select important terms and users with respect to each party's voting intention, we test the predictive accuracy of our model in a forecasting setup. We again emulate a real-life scenario of voting intention prediction from tweets, using historic training data. The model is trained on a fixed size sliding window of past polls and matched tweet data. Once a BGGR model is trained, predictions are made using tweet data alone. The difference between these predictions and the ground truth voting

[2] http://www.cs.jhu.edu/~svitlana/

[3] http://labs.yougov.co.uk

intention is measured. This counts as a single fold of the experiment. In the next fold, the training sliding window is moved to subsume the previous fold's test window. This means for each set of tests, the training window is the same length and holds data from a fixed period of time before the test elements.

In the experiments we present in this section, the training window size was fixed at 185 points and the test window size was 5. We test on 10 different folds, resulting in a total of 50 predictions. Due to the relative expense of the graph solver, the parameters for BGGR, namely λ_1 and λ_2 were not optimised for these experiments as they were for BGL or BEN. Instead, parameters were chosen such that a sparsity level of around 99% was achieved for both terms and users selected. In the future, we will explore automatic tuning of these two parameters.

Table 3.1 shows the average Root Mean Square Errors (RMSE) across the 10 folds. Table 3.1a presents the BGGR model where region information was used and in Table 3.1b we show similar results for the gender grouping. On top of the results for our BGGR model, we present the results for $\mathbf{B}_\mu$ and $\mathbf{B}_{\text{last}}$ for comparison. $\mathbf{B}_\mu$ calculates an average poll across all the training responses and uses it as the answer for each test item, while $\mathbf{B}_{\text{last}}$ uses the previous day's results to predict the current results. In these results, a Mean Square Error (MSE) is calculated across the folds such that $\text{MSE} = \frac{1}{n}\sum_{i\in\text{folds}}(Y_i^{(\text{predicted})} - Y_i^{(\text{correct})})^2$ and a $RMSE = \sqrt{MSE}$. $\boldsymbol{\mu}$ denotes the mean RMSE across all groups for a political party, and $\boldsymbol{\mu_{\text{all}}}$ is the mean RMSE across all parties and groups.

	CON						LAB						LBD						$\boldsymbol{\mu}$
	S	L	M	N	Sc	μ	S	L	M	N	Sc	μ	S	L	M	N	Sc	μ	μ_{all}
$\mathbf{B}_\mu$	3.9	4.6	3.9	4.4	4.0	4.2	3.1	4.8	3.8	3.7	5.4	4.2	1.9	2.5	2.0	1.6	2.2	2.0	3.4
$\mathbf{B}_{\text{last}}$	3.4	5.2	4.7	4.6	5.6	4.7	3.6	6.1	5.1	4.2	7.8	5.4	2.1	3.6	3.2	2.3	2.7	2.8	4.3
BGGR	3.4	4.5	4.1	3.9	3.9	**3.9**	2.7	4.8	3.7	3.5	5.0	**3.9**	1.9	2.5	2.0	1.6	2.2	**2.0**	**3.3**

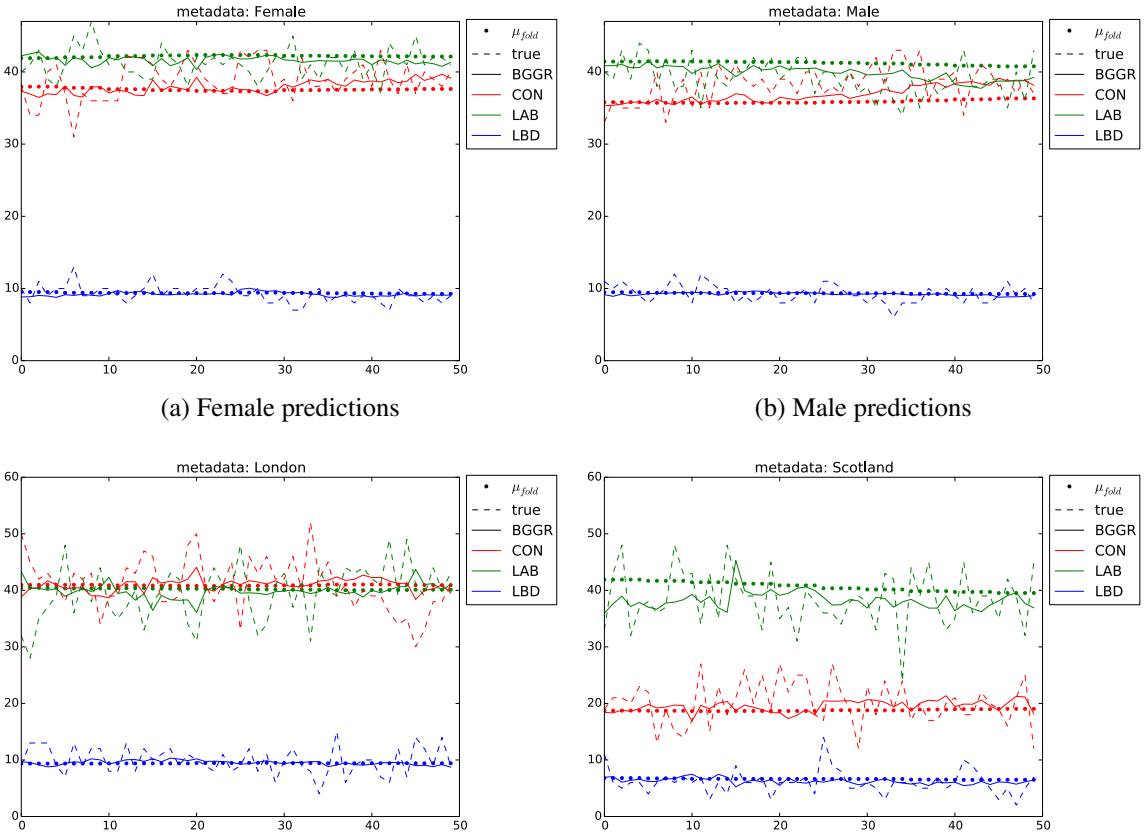
(a) Region Metadata

	CON			LAB			LBD			$\boldsymbol{\mu}$
	F	M	μ	F	M	μ	F	M	μ	μ_{all}
$\mathbf{B}_\mu$	3.0	3.4	3.2	2.2	3.1	2.7	1.2	1.2	**1.2**	2.4
$\mathbf{B}_{\text{last}}$	3.0	3.1	3.0	2.7	3.4	3.1	1.5	1.5	1.5	2.5
BGGR	2.8	2.7	**2.8**	2.2	2.5	**2.3**	1.3	1.3	1.3	**2.1**

(b) Gender Metadata

Table 3.1: UK voting intention — RMSE for political parties for given regions/genders, averages within parties across regions/genders and averages across parties across regions/genders.

Figure 3.2 shows the predicted, mean and actual voting intentions over the 10 folds (50 data points). We show both Female (Figure 3.2a) and Male (Figure 3.2b) results along



(a) Female predictions

(b) Male predictions

(c) London predictions

(d) Scotland predictions

Figure 3.2: Predictions over 10 folds of the BGGR model.

(a) Female

(b) London

Figure 3.3: Highest magnitude positive terms selected in fold 3 by the BRRG model.

with the results for London (Figure 3.2c) and Scotland (Figure 3.2d).

From Table 3.1 we can see that in almost all cases BGGR achieved a better result than both of the powerful baselines $\mathbf{B}_{\text{last}}$ and $\mathbf{B}_{\mu}$. A notable situation where this is not the case is the results for the Liberal Democratic party between Males and Females. However, upon closer inspection of Figures 3.2a and 3.2b it is clear that the overall change of Liberal Democrats is much less than that of Labour or Conservatives, meaning the baselines of averages results from the past are very difficult to beat. Also of note is the predictive ability of our model in the prediction of London poll results. We can see from distinct events near test points 20 and 40 that our predictive model showed an ability to follow the overall trend of both the Conservative and Labour signals. From this period of interesting events we present the highest weighted terms selected in test 15 to 20 (i.e. fold 3) for London as a word cloud in Figure 3.3b and the terms selected in the same period for Female in 3.3a. The size of the words in these diagrams represent their overall magnitude. In these term clouds we can see that though some words such as 'christmas' are selected in both contexts, it is clear that in the London metadata, more geographically significant terms such as 'london' are selected.

Though the magnitudes of the predictions made by BGGR were not on par with the ground truth, we believe this is likely an effect of not optimising the parameters of the BGGR regulariser; further optimisation of these is expected to improve results. Also, from the word clouds we can see that the model is selecting very generic words such as 'good' and 'happy' which, taken on their own, are not very informative about the underlying effects. An exploration into the use of deeper linguistic features such as bigrams or named entities as terms as opposed to unigrams, is expected to yield more semantically relevant terms.

3.2 Regional topic models

This section presents an extension of the temporal topic model introduced in Section 2.1 for incorporating geographic features in order to extract regional topics. For this purpose we introduce a set of novel features derived from multiple geographic indicators and evaluate their usefulness on the same Twitter ($\mathcal{D}_1$) dataset used in our temporal experiments from Section 2.1.

3.2.1 Methods

As highlighted by the experiments in Section 2.1, the temporal dependencies of social media text play an important role in the topics discussed. In addition to this, there are other factors which influence the prevalent topics. One such factor we study is the geographic information which augments social media data. For instance, during city-specific events, such as those related to a local football team (#werder, #schalke) or local politics,

messages related to this event are likely only popular in the respective city and maybe a few more surrounding cities. For other larger scale events, e.g. country-wide events, such as such as the national football league (#bundesliga) or national politics (#merkel), the tweets are more likely to span a larger geographic region, such as the an entire country (Germany). However, global events, such as the Olympics (#london2012), would be probably discussed as much without regard to the location of the authors.

We use the same LDA and DMR models as in Section 2.1.1. In order to capture these geographic dependencies among topics, we propose a set of novel features (x_d) for DMR in a similar way to Section 2.1.1:

Modelling geographic information with city indicator features (CtyId) This representation captures our natural intuition to consider the geographic location, in particular the city in which the documents have been collected. Documents authored within the same city are more likely to share the same topics than documents written in different cities. This could be for example the case of local events, politics or sports teams. Considering the list of unique cities in which the documents have been posted, the feature consists of a Boolean indicator for each city with a value of 1 for the city where the data was collected.

Modelling geographic information with Country Indicator features (CouId) This feature set aims to capture the temporal dependencies among topics at a a broader country level. The main intuition here is that documents written in the same country are more likely to share the same topics than document written in different countries. This could be for example, the case of country specific TV shows or national politics. Considering the list of unique countries (i.e. Austria and Germany) in which the documents have been posted, thus this feature consists of a Boolean indicator for each country with a value of 1 for the country where the data was collected.

Modelling geographic information with Geographical smoothing (GeoRBF) By grouping documents based on their location together we lose any sharing of information between neighbouring locations. For example, considering each city in isolation could allow us to learn the topics specific to a city given that a sufficient amount of tweets are available for that location. However, by considering the geographic boundaries between cities (e.g. cities within a specific distance to each other), could allow to further detect coherent topics for these geographic regions. For this reason, we aim to model geographic continuity and add a geographic smoothing effect, where documents within a geographic region influence each other, similarly to the TimeRBF method describe in Section 2.1.1.

To achieve this, we employ the RBFs kernels in order to transform the data. Each particular document is collected from a location with geo-coordinates $g = (g_{lat}, g_{lon})$, where g_{lat} is the latitude and g_{lon} is the longitude. An RBF kernel centred in a location

$g^c = (g^c_{lat}, g^c_{lon})$ captures the distance between the two locations:

$$\text{RBF}_{\text{geo}}(g, g^c) = \exp\left(\frac{-(d_{eu}(g - g^c))^2}{2\sigma^2_{geo}}\right) \tag{3.5}$$

where d_{eu} is the Euclidean distance between the two locations, computed as follows: $d_{eu}(g - g^c) = \sqrt{(g_{lat} - g^c_{lat})^2 + (g_{lon} - g^c_{lon})^2}$.

The RBF function allows for a non-linear dependency between two locations as a function of by the *Euclidean distance* d_{eu} to the kernel centre and the *shape* parameter σ^2_{geo} which gives the amount of decay in influence across space. Considering the centre of each RBF kernel is fixed, varying the shape of this distribution would allow us to control the amount of influence between neighbouring geographic locations.

Similarly to the city indicator features, we created a set of RBF kernels corresponding to each city in which documents were collected. Given the kernels, each document's location will be mapped using the kernels into a set of continuous values.

3.2.2 Results

We now present experimental results for regional topic modelling on the $\mathcal{D}_1$ dataset with an emphasis of regional modelling.

Quantitative results

The main aim of these experiments is to evaluate the impact of incorporating geographic information into the DMR topic model, using perplexity as an error metric. The previous experiments presented in Section 2.1 have shown that the DMR temporal model with RBF kernels with $\sigma = 20$ (DMR TimeRBF σ=20) outperformed both the baseline LDA and the DMR with monthly indicators (DMR MId) model. For this reason, this subsection employs as the initial topic model the DMR TimeRBF σ=20 model, and evaluates its performance using geographic features in addition.

In order to allow direct comparison between the models, we used the same experimental setup as for the temporal DMR models: we run the sampler for 1000 iterations, set the burn-in to 250, the total number of topics to 100 ($T = 100$) and averaging the results across 3 runs. For the RBF Geo models, we experimented with different values for $\sigma_{geo} = \{0.5, 1, 1.5, 2, 2.5\}$. The results comparing the methods are presented in Table 3.2.

We observe that incorporating geographic information into topic models consistently and significantly improves upon the baseline LDA and the DMR TimeRBF ($\sigma = 20$) model. Looking at the indicator features, we notice that both country and city indicator features improve results and the combination of these two features achieved the best overall results. Concerning the geographic smoothing features, we observe that smaller σ_{geo} values lead to lower perplexity values. This indicates that regional topic patterns seem to

Features	Method	Perplexity
–	LDA	12,777.19
Temporal	DMR TimeRBF($\sigma = 20$)	12,412.40
Temporal & Regional	DMR TimeRBF($\sigma = 20$)+CouId	12,390.57
	DMR TimeRBF($\sigma = 20$)+CouId+CtyId	**12,167.11**
	DMR TimeRBF($\sigma = 20$)+CouId+GeoRBF($\sigma = 0.5$)	12,233.74
	DMR TimeRBF($\sigma = 20$)+CouId+GeoRBF($\sigma = 1$)	*12,190.91*
	DMR TimeRBF($\sigma = 20$)+CouId+GeoRBF($\sigma = 1.5$)	12,257.32
	DMR TimeRBF($\sigma = 20$)+CouId+GeoRBF($\sigma = 2$)	12,320.52
	DMR TimeRBF($\sigma = 20$)+CouId+GeoRBF($\sigma = 2.5$)	12,364.71

Table 3.2: Perplexity on held out data for $\mathcal{D}_1$ dataset. Lower is better.

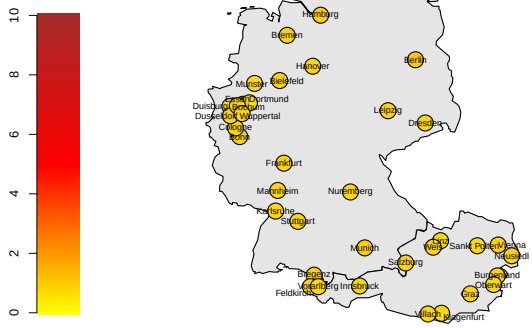


Figure 3.4: Spatial distribution for Topic #64.

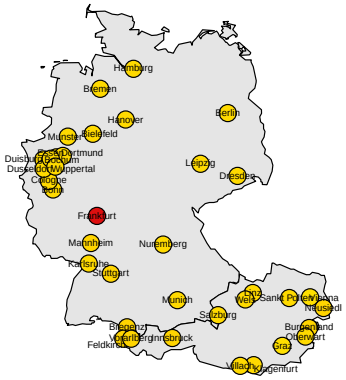


Figure 3.5: Spatial distribution for Topic #69.

Topic	Top Terms
Topic #64	obama romney #uswahl #obama us-wahl november wahl #romney barack #zdfcamp
Topic #69	#frankfurt frankfurter #eintracht main #stellenangebot #hiring hessen #wirtschaft #job #finanze

Table 3.3: The top terms for Topics #64 and #69 from Figures 3.4 and 3.5.

appear for cities which are closer to each other rather than cities which are a more distant. The best results were obtained for $\sigma_{geo} = 1$. These results, are however worse than the ones which use the indicator features. We can thus conclude that for geographical features, the city level information is the most relevant and sharing of information between regions does not aid our models in this setting. However, this is more likely to change when the data is drawn more uniformly over a region, and not only assigned to a city.

Qualitative results

We now present qualitative results of the regional topic models. We present two selected topics from the best performing model (DMR TimeRBF σ=20+ CouId + CtyID) as list of top words and their spatial distribution across Germany and Austria in Figures 3.5 and 3.4.

Topic #64, presenting the US presidential election is discussed almost equally in all cities used for building our dataset. However, Topic #69 shows to have a different spatial distribution from Figure 3.4. The tweets authored in the city of 'Frankfurt' in Germany use this topic much more frequently than others. If we examine the top words of the topic, we find relevant city keywords such as '#frankfurt', 'frankfurter', 'main' (the river the city is situated on), 'hessen' (the region containing the city), '#eintracht' (the local football team) as well as words related to jobs in finance ('#job', '#finanzen'), a domain the city is at the center of in Germany.

Chapter 4

Application to news media

In recent years there has been a shift in paradigms in posting of online content towards user-generated content, such as social media. However, traditional news outlets continue to be a central reference point (Nah and Chung, 2012) as they still have the advantage of being professionally authored, alleviating the noisy nature of citizen journalism formats. In this chapter we present a dataset of news summaries spanning eight years of European Union (EU) related news. We experiment on this new dataset with prediction and clustering methods we have previously developed for social media, testing their efficiency and robustness. The dataset presented in Section 4.1 and the experiments from Section 4.2.1 are published in (Lampos et al., 2014).

4.1 Experimental setup

For these experiments we gathered a news summaries dataset, consisting of summaries of news items covering European Union issues, as published by the Open Europe Think Tank.[1] The press summaries are daily collections of news items about the EU or its member countries with a focus on politics; the news outlets used to compile each summary are listed below the summary's text. The summaries are published every weekday, with the major news being covered in a couple of paragraphs, and other less prevalent issues being mentioned in one paragraph to as little as one sentence. The press summaries were first published on 1 February 2006, and were collected up to 15 November 2013, creating a dataset with the temporal resolution of 1913 days (or 94 months).

The text content of the summaries was cleaned up and tokenised using the NLTK tokeniser (Bird et al., 2009). News outlets with fewer than 5 mentions were removed, resulting in a total of 435 sources. Some summaries are listed with no source, which have all been attributed to the same user 'No Source'. Each summary contains on average 14 news items, with an average of 3 news sources per item; where multiple sources were

[1] `http://www.openeurope.org.uk/Page/PressSummary/en/`

present, the summary was assigned to all the referenced news outlets. After removing stopwords, we ended up with the most frequent $8,413$ unigrams and $19,045$ bigrams; their daily occurrences were normalised using the total number of news items for that day. We will refer to this dataset as the **EUSummaries** ($\mathcal{D}_3$) dataset.

The daily press summaries are similar to Twitter data in two main aspects: each summary has an associated time, in our case with the granularity of single days and each summary has actors associated, in this case the press outlets that reported on the issue. The press summaries can thus be treated in the same manner as tweets that may have been simultaneously tweeted by a number of different users.

4.2 Forecasting socioeconomic indicators

In this section we present experiments in analysing socioeconomic patterns in news articles. Our analysis show how Machine Learning methods can be used to gain insights into the interplay between text in news articles, the news outlets and socioeconomic indicators. The experiments are performed using the news summaries ($\mathcal{D}_3$) dataset with the intention to study two basic socioeconomic factors: EU's unemployment rate and Economic Sentiment Index (ESI) (European Commision, 1997). To determine connections between the news, the outlets and the indicators of interest, we use formulate the task as bilinear text-based regression as previously introduced in (Lampos et al., 2013) and D3.1.2 (Samangooei et al., 2013).

For the purposes of our supervised analysis, we use the response variables of ESI and unemployment rate across the EU. The monthly time series of these socioeconomic indicators were retrieved from Eurostat,[2] EU's statistical office (see the red lines in Fig. 4.1 and 4.2 respectively). ESI is a composite indicator often seen as an early predictor for future economic developments (Gelper and Croux, 2010). It consists of five confidence indicators with different weights: industrial (40%), services (30%), consumer (20%), construction (5%) and retail trade (5%). The unemployment rate is a seasonally adjusted ratio of the non employed persons over the entire EU labour force.[3] High unemployment usually coincides with periods of economic recession.

4.2.1 Experiments

We apply both **LEN** (Linear Elastic Net) (Zou and Hastie, 2005) and its bilinear counterpart **BEN** (Bilinear Elastic Net) (Lampos et al., 2013). Both models are applied to the news summaries ($\mathcal{D}_3$) dataset with the aim to predict EU's ESI and rate of unemployment. The predictive capability of the derived models, assessed by their respective

[2] http://epp.eurostat.ec.europa.eu

[3] http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Unemployment_statistics

	ESI	Unemployment
LEN	9.253 (9.89%)	0.9275 (8.75%)
BEN	**8.209** (8.77%)	**0.9047** (8.52%)

Table 4.1: 10-fold validation average RMSEs (and error rates) for LEN and BEN on ESI and unemployment rates prediction.

inference performance, is used as a metric for judging the degree of relevance between the learnt model parameters – word and outlet weights – and the response variable. A strong predictive performance increases confidence on the soundness of those parameters.

To match input with the monthly temporal resolution of the response variables, we compute the mean monthly term frequencies for each outlet. Evaluation is performed via a 10-fold validation, where each fold's training set is based on a moving window of $p = 64$ contiguous months, and the test set consists of the following $q = 3$ months; formally, the training and test sets for fold i are based on months $\{q(i-1)+1, ..., q(i-1)+p\}$ and $\{q(i-1)+p+1, ..., q(i-1)+p+q\}$ respectively. In this way, we emulate a scenario where we always train on past and predict future points.

Performance results for LEN and BEN are presented in Table 4.1; we show the average Root Mean Squared Error (RMSE) as well as an error rate (RMSE over $\mu(y)$) across folds to allow for a better interpretation. BEN outperforms LEN in both tasks, with a clearer improvement when predicting ESI. Predictions for all folds are depicted in Figures 4.1 and 4.2 together with the actual values. Note that reformulating the problem into a multi-task learning scenario where ESI and unemployment are modelled jointly, similarly to (Lampos et al., 2013), did not improve accuracy.

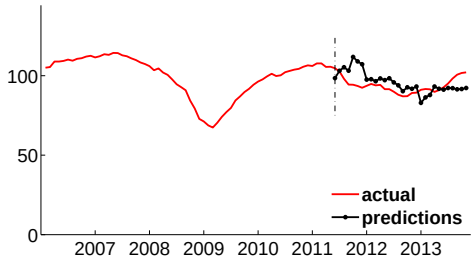


Figure 4.1: Time series of ESI together with BEN predictions (smoothed using a 3-point moving average).

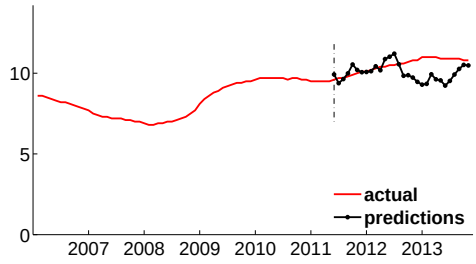
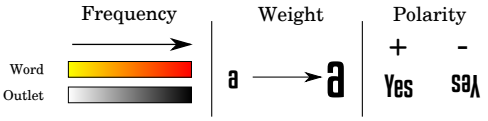


Figure 4.2: Time series of unemployment together with BEN predictions (smoothed using a 3-point moving average).

The relatively small average error rates ($< 8.8\%$) make meaningful a further analysis of the model's outputs. We present the most recent results, depicting the models derived in the 10th fold. Following Schwartz et al. (2013), we use a word cloud visualisation, where the font size is proportional to the derived weights by applying BEN, flipped terms

Figure 4.3: ESI. Word cloud for words and outlets visualising the outputs of BEN.

Figure 4.4: Unemployment. Word cloud for words and outlets visualising the outputs of BEN.



denote negative weights and colours are determined by the frequency of use in the corpus (Figure 4.3 and 4.4). Word clouds depict the top-60 positively and negatively weighted n-grams (120 in total) together with the top-30 outlets; bigrams are separated by '_'.

Our visualisations (Figures 4.3 and 4.4) present various interesting insights into the news and socioeconomic features being explored, serving as a demonstration of the potential power of the proposed modelling. Firstly, we notice that in the word cloud, the size of a feature (BEN's weight) is not tightly connected with its colour (frequency in the corpus). Also, the word clouds suggest that mostly different terms and outlets are selected for the two indicators. For example, '*sky.it*' is predominant for ESI but not for unemployment,

while the opposite is true for '*hedgefundsreview.com*'. Some of the words selected for ESI reflect economical issues, such as '*stimulus*' and '*spending*', whereas key politicians like '*david_cameron*' and '*berlusconi*', are major participants in the word cloud for unemployment. In addition, the visualisations show a strong negative relationship between unemployment and the terms '*food*', '*russia*' and '*agriculture*', but no such relationship with respect to ESI. The disparity of these selections is evidence for our framework's capability to highlight features of lesser or greater importance to a given socioeconomic time series.

4.2.2 Using rich text features

For these experiments, the summaries ($\mathcal{D}_3$) dataset was annotated using GATE (Cunningham et al., 2002) by adding annotations for Part-of-Speech (POS) tags and Named Entities (NE), as well as annotations for named entities found in DBpedia [4] using the methods from D2.2.2 (Aswani et al., 2013). In particular, the following features are added to the previously mentioned features:

- Unigram counts of every word with the POS tag it has been annotated with, for example: 'Europe/NNP' (**POS**);
- Counts of identified named entities with the type of named entity it is annotated as (either Person, Location or Organisation), for example: 'Person:Jean_Claude_Juncker' (**Entities**);
- Counts of various annotations derived from the DBpedia ontology of entities tagged by the system. These properties are fully described below (**Annotations**).

There are several features that are inferred from the DBpedia annotations, in addition to the DBpedia instance name. These are:

- the offices any person might hold
- all political parties a person might be or have been affiliated with
- a persons birthplace
- the country an organisation or location is located in
- the leader of any location

These annotations allow a mention of 'Angela Merkel' to have a similar effect as a mention of 'The Chancellor of Germany', and similarly for organisations and their countries, for example 'France' and 'the French Parliament' as for other properties.

[4] `http://dbpedia.org`

We have built new representations for the different types of feature categories. News outlets with fewer than 10 mentions were removed, resulting in a total of 296 sources. For all features we have used the same frequency cutoff (i.e. 12). After removing stopwords, the number of features in each category is: $8,912$ Unigrams, $33,206$ Bigrams, $10,277$ POS, $1,013$ Entities and $3,392$ Annotations. Their daily occurrences were normalised using the total number of news items for that day. We also have experimented with fixed regulariser parameters, which explains the small differences in results compared to the previous section.

Experimental results with the different types of features are presented in Table 4.2.

Features	**ESI**	**Unemployment**
Unigrams	8.21	1.27
Bigrams	9.66	1.61
Unigrams + Bigrams	8.91	1.47
POS	**7.87**	1.14
Entities	9.59	1.45
POS + Entities	8.09	**1.12**
Entities + Annotations	12.67	1.62
POS + Entities + Annotations	10.50	1.31
Unigrams + Entities + Annotations	10.92	1.31
Unigrams + Bigrams + Entities + Annotations	10.81	1.53

Table 4.2: Held out RMSE across 10 folds using BEN model with different types of features. Lower is better.

By observing results with both outcomes, we can derive some patterns of improvement. Firstly, unigram features obtain better results than bigram features. By adding the later set of features to the unigram features, the predictive performance is decreased. However, POS features which include unigrams annotated with their respective part-of-speech add further improvement to the unigram features. This is expected, as POS features disambiguate between different uses of the same word, which are conflated by only using unigrams. We notice that entities have a relatively good predictive performance (better than Bigrams alone) and in one case they add to the performance when joined with the POS features. This is despite that the Entities class has the lowest number of features. Using the Entities is also beneficial for interpretability, as they usually represent known persons, locations or organisations. The Annotation features however are not suitable for use, as all the methods that included them saw a drop in predictive performance.

4.3 Temporal and regional variation

The news summaries dataset ($\mathcal{D}_3$) spans a temporal length of eight years and discusses EU wide events which span a number of different member states. Spatial information about the news can be discovered via the news outlets. In this section we present the results obtained for both temporal and regional topic modelling using the $\mathcal{D}_3$ dataset. Similarly to the experiments presented on the $\mathcal{D}_1$ Twitter dataset, we used the implementation of the topic models from the MALLET toolkit,[5] and experimented with various temporal and geographic feature indicators for the DMR model as introduced in Sections 2.1 and 3.2.

For clarity, we provide a short summary of the experimental setup, including the description of the temporal and regional features. First, in order to capture the geographical information of a document, we use the country information of each news provider. We obtained this information, by parsing the domain name of each news provider's website and extracted the top-level domain. Most top-level domains represent country codes (e.g. telegraph.co.uk), while a few (e.g. ft.com) are generic. Further, another difference compared to the $\mathcal{D}_1$ Twitter dataset, is that in these experiments we considered as an individual documents each news item published by a specific news outlet at a given time. In contrast to the previous experiments, we did not aggregate each news on a daily basis to compile the documents for the topic models.

To summarise, we considered as temporal information the time (TimeRBF) the timestamp of the news item and as geographic information the top-level domain of the news provider (CountryId). We have also added the news outlet which authored the news (OutletId) as a feature, as this is likely to contain important information as well.

Quantitative results

Features	Method	Perplexity
–	LDA	4,597.08
Temporal	DMR MInd	4,575.30
	DMR TimeRBF ($\sigma = 30$)	4,545.19
	DMR TimeRBF ($\sigma = 50$)	**4,262.56**
	DMR TimeRBF ($\sigma = 75$)	4,530.62
Temporal & Regional	DMR TimeRBF ($\sigma = 50$)+OutletId	4,086.23
	DMR TimeRBF ($\sigma = 50$)+CouId	4,231.10
	DMR TimeRBF ($\sigma = 50$)+CouId+OutletId	**4,036.56**

Table 4.3: Perplexity on held out data for the $\mathcal{D}_3$ dataset. Lower is better.

For quantitatively assessing the performance of the topic models we employed per-

[5] http://mallet.cs.umass.edu/

plexity 2.2. We followed the same experimental setup as in (Mimno and McCallum, 2008): we run the sampler for 1000 iterations, set the burn-in to 250, chose 100 as the total number of topics ($T = 100$).

For the temporal models (TimeRBF), we experimented with different values for $\sigma = \{30, 50, 75\}$ for varying the influence of neighbouring time intervals. As RBF centres we chose the middle of each month, which was shown to achieve best performance. Further, we divide the dataset into train and test datasets using a 90%-10% split. We evaluated each graphical model on the held out data and averaged the results over 3 independent runs. The results comparing the methods are presented in Table 4.3.

Looking at the results we can notice that incorporating temporal features into the topic model is beneficial with both indicator features (MInd) and RBF kernels (TimeRBF) consistently improving upon the baseline LDA model. The best results, were obtained for the DMR TimeRBF model with $\sigma = 50$, which further shows the importance of adding a temporal smoothing constraints for modelling time. These results are also in line with our observations obtained for the $\mathcal{D}_1$ Twitter dataset.

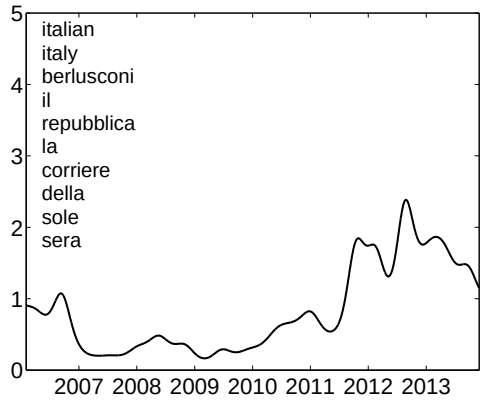


Figure 4.5: Temporal distribution and top words for Topic #70.

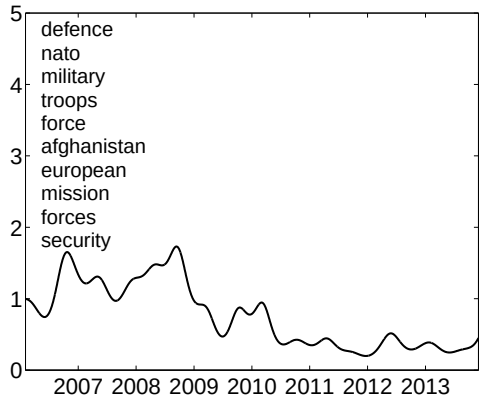


Figure 4.6: Temporal distribution and top words for Topic #91.

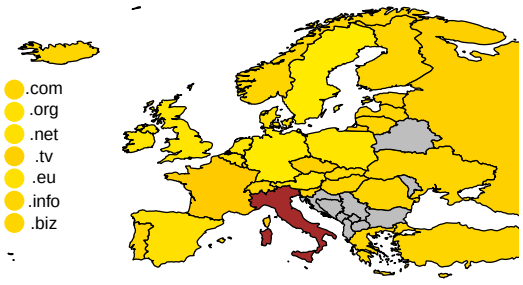


Figure 4.7: Spatial distribution for Topic #70.

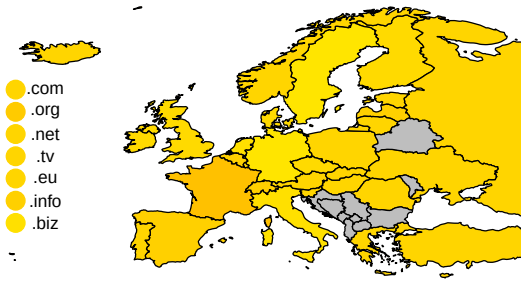


Figure 4.8: Spatial distribution for Topic #91.

Inspecting the results obtained by adding the regional features, we can observe an additional improvement over the temporal models. This indicates that capturing the ge-

ographic source as well as the news outlet information to the topics is beneficial. While both CouId and OutletId performed better than the temporal features alone, the best results were obtained by the combination of these features. These results also match the results obtained for the $\mathcal{D}_1$ Twitter dataset.

Qualitative results

We now qualitatively examine two selected topics learnt using the best performing method (DMR TimeRBF σ=50 + CouId + OutletId) which incorporates both temporal and spatial information. We present the top words of the topics, their temporal profile over eight years (Figures 4.5 and 4.6) and their spatial distribution (Figures 4.7 and 4.8).

We first notice that the temporal importance of both topics is very oscillating across the eight years of the dataset. Topic #70 consists on words related to the Italian politician Silvio Berlusconi and about Italian news in general, with the most popular news sources present in the top words ('Corriere della Sera', 'La Republica'). This topic has a strong spatial coherence, with Italy having the most text on this topic. Topic #91 is about the military intervention in Afghanistan, which decreases in news coverage after 2008. From a spatial point of view, we notice that the topic is equally mentioned in all countries in Europe.

Chapter 5

Conclusions

This deliverable has presented various methods which allow to better incorporate the temporal, spatial and user information in modelling of text streams. With regards to temporal variation, we have presented a method to categorise and forecast word frequency time series from social media. Further, we have introduced a soft clustering method of identifying topics jointly with their temporal and spatial distributions. In order to incorporate spatial and demographic user information in a predictive model, we have extended our previously introduced set of bilinear models. We have also shown that both clustering and predictive methods can be used off-the-shelf on data from a different source (i.e. news summaries). On this dataset, we have also experimented with richer linguistic features, showing how these can aid performance and interpretability.

The methods introduced in this deliverable have clear practical applicability to TrendMiner's use cases. The spatio-temporal topic model has shown to produce better quality topics than by ignoring these factors. Both temporal and regional information can be displayed to the end user alongside cluster membership in order to better understand topic evolution and their spatial distribution. A concrete application would be to present 'trending' topics as given by their prevalence at the current time compared to previous intervals or topics which are of broad interest, rather than regional ones.

An extension of the bilinear model which incorporated regional and demographic user information was introduced. This extension was shown to be capable of selecting regionally or demographically meaningful features. These specific features can be displayed in parallel, for example using word clouds representing weights for different regions or genders. Also, relevant users for each region can be highlighted, thus identifying 'regional representatives' for public opinion.

Experiments on news summaries showed that similar patterns of improvements for our models are achieved even when confronted with data from a different type of source. This establishes our goal of providing tools which are independent of language, external resources and specific data sources. However, experiments using richer linguistic features, such as part-of-speech tags and entities have shown that extra linguistic processing can

aid performance and interpretability, where available. For better integration and use in further applications, all our methods have been released as open-source code.

Appendix A

City list

Id	City	Country	Latitude	Longitude
1	Bregenz	Austria	47.516	9.766
2	Burgenland	Austria	47.5	16.333
3	Feldkirch	Austria	47.25	9.6333
4	Graz	Austria	47.083	15.366
5	Innsbruck	Austria	47.266	11.4
6	Klagenfurt	Austria	46.633	14.333
7	Linz	Austria	48.316	14.3
8	Neusiedl	Austria	47.966	16.85
9	Oberwart	Austria	47.3	16.2
10	Salzburg	Austria	47.8	13.05
11	Sankt Polten	Austria	48.2	15.616
12	Vienna	Austria	48.216	16.366
13	Villach	Austria	46.616	13.85
14	Vorarlberg	Austria	47.25	9.916
15	Wels	Austria	48.166	14.033
16	Berlin	Germany	52.516	13.4
17	Bielefeld	Germany	52.033	8.533
18	Bochum	Germany	51.483	7.216
19	Bonn	Germany	50.733	7.1
20	Bremen	Germany	53.083	8.8

Table A.1: Cities for collection, their contry and geo-coordinates

Id	City	Country	Latitude	Longitude
21	Cologne	Germany	50.933	6.95
22	Dortmund	Germany	51.516	7.45
23	Dresden	Germany	51.05	13.75
24	Duisburg	Germany	51.433	6.75
25	Dusseldorf	Germany	51.216	6.766
26	Essen	Germany	51.45	7.016
27	Frankfurt	Germany	50.116	8.683
28	Hamburg	Germany	53.55	10.0
29	Hanover	Germany	52.366	9.716
30	Karlsruhe	Germany	49.004	8.385
31	Leipzig	Germany	51.333	12.416
32	Mannheim	Germany	49.483	8.464
33	Munich	Germany	48.15	11.583
34	Munster	Germany	51.966	7.633
35	Nuremberg	Germany	49.447	11.068
36	Stuttgart	Germany	48.766	9.183
37	Wuppertal	Germany	51.266	7.183

Table A.2: Cities for collection, their contry and geo-coordinates

Bibliography

Argyriou, A., Evgeniou, T., and Pontil, M. (2007). Multi-Task feature learning. In *Advances in Neural Information Processing Systems*, NIPS.

Aswani, N., Gorrell, G., Bontcheva, K., Petrak, J., Declerck, T., and Krieger, H.-U. (2013). Multilingual, Ontology-Based IE from Stream Media - v2. *Public deliverable, Trendminer Project*.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A framework and graphical development environment for Robust NLP Tools and Applications. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, ACL.

Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., and Ghahramani, Z. (2013). Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the International Conference on Machine Learning*, ICML.

Eisenstein, J., O'Connor, B., Smith, N. A., and Xing, E. P. (2010). A latent variable model for geographic lexical variation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 1277–1287.

European Commision (1997). *The joint harmonised EU programme of business and consumer surveys*. European economy: Reports and studies.

Gelper, S. and Croux, C. (2010). On the construction of the European Economic Sentiment Indicator. *Oxford Bulletin of Economics and Statistics*, 72(1):47–62.

Lampos, V., Preoţiuc-Pietro, D., and Cohn, T. (2013). A user-centric model of voting intention from Social Media. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics*, ACL, pages 993–1003.

Lampos, V., Preoţiuc-Pietro, D., Samangooei, S., Gelling, D., and Cohn, T. (2014). Extracting socioeconomic patterns from the news: Modelling text and outlet importance jointly. In *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics, Workshop on Language Technologies and Computational Social Science*, ACL.

Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Journal Mathematical Programming: Series A and B*, 45(3):503–528.

Mairal, J., Jenatton, R., Obozinski, G., and Bach, F. (2010). Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems*, NIPS.

Mimno, D. and McCallum, A. (2008). Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, UAI.

Nah, S. and Chung, D. S. (2012). When citizens meet both professional and citizen journalists: Social trust, media credibility, and perceived journalistic roles among online community news readers. *Journalism*, 13(6):714–730.

Preoţiuc-Pietro, D. and Cohn, T. (2013). A temporal model of text periodicities using Gaussian Processes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP.

Preoţiuc-Pietro, D., Samangooei, S., Cohn, T., Gibbins, N., and Niranjan, M. (2012). Trendminer: An architecture for real time analysis of social media text. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media, Workshop on Real-Time Analysis and Mining of Social Streams*, ICWSM.

Preoţiuc-Pietro, D., Samangooei, S., Lampos, V., Cohn, T., Gibbins, N., and Niranjan, M. (2013). Clustering models for discovery of regional and demographic variation. *Public deliverable, Trendminer Project*.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. MIT Press.

Rout, D., Preoţiuc-Pietro, D., Kalina, B., and Cohn, T. (2013). Where's @wally: A classification approach to Geolocating users based on their social ties.

Samangooei, S., Lampos, V., Cohn, T., Gibbins, N., and Niranjan, M. (2013). Regression models of trends. Tools for mining non-stationary data: functional prototype. *Public deliverable, Trendminer Project*.

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, M. E. P., and Ungar, L. H. (2013). Personality, Gender, and Age in the Language of Social Media: The Open-Vocabulary Approach. *PLoS ONE*, 8(9).

Stevens, K., Kegelmeyer, P., Andrzejewski, D., and Buttler, D. (2012). Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP, pages 952–961.

Tsur, O. and Rappoport, A. (2012). What's in a hashtag? content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, WSDM.

Volkova, S., Wilson, t., and Yarowsky, D. (2013). Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP.

Wallach, H., Murray, I., Salakhutdinov, R., and Mimno, D. (2009). Evaluation methods for topic models. In *Proceedings of the International Conference on Machine Learning*.

Yang, J. and Leskovec, J. (2011). Patterns of temporal variation in online media. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, WSDM.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *JRSS: Series B*, 67(2):301–320.