Exchange Rate Prediction from Twitter’s Trending Topics

Fulya Ozcan
University of California, Irvine

September 30, 2016

Abstract

This paper predicts intra-day foreign exchange rates by making use of the trending topics from Twitter, using a sentiment based topic clustering algorithm. Twitter trending topics data provide a good source of high frequency information, which would improve the short-term or intra-day exchange rate predictions. This project uses an online dataset, where trending topics in the world are fetched from Twitter every ten minutes since July 2013. First, using a sentiment lexicon, the trending topics are assigned a sentiment (negative, positive, or uncertain), and then using a continuous Dirichlet process mixture model, the trending topics are clustered regardless of whether they are explicitly related to the currency under consideration. This unique approach enables to capture the general sentiment among users, which implicitly affects the currencies. Finally, the exchange rates are estimated using a linear model which includes the topic based sentiment series and the lagged values of the currencies, and a VAR model on the topic based sentiment time series. The main variables of interest are Euro/USD, GBP/USD, Swiss Franc/USD and Japanese Yen/USD exchange rates. The linear model with the sentiments from the topics and the lagged values of the currencies is found to perform better than the benchmark AR(1) model. Incorporating sentiments from tweets also resulted in a better prediction of currency values after unexpected events.

Keywords: Natural Language Processing, Dirichlet Mixture Models, Exchange Rate Prediction, Trending Topics, Twitter.
1 Introduction

The advancements in the data collecting and accumulating technologies have made tremendous amount of data available in many fields, varying from finance to political science to healthcare. This also enabled different types of data that were not available before to be used interdisciplinarily, such as personal data. One of the most groundbreaking strand of literature that arose with the availability of this personal big data is the use of social media data in social sciences and business.

Making use of the data available from social media helps incorporate the beliefs, expectations, information levels and behaviors of agents into the models. This has been vastly explored in finance, and lately in economics. Using social media data with machine learning algorithms is found to increase the predictive power of models in these disciplines. This paper utilizes text data from social media in exchange rate prediction models. In particular, the aim is to predict foreign exchange rates making use of the trending topics from Twitter, using natural language processing.

Text data has been very popular in finance, particularly for modeling the movements in the stock market, using data from various means, varying from news headlines to Google Trends. Applications of techniques to economics, on the other hand, have been comparatively limited. Especially the studies that focus on foreign exchange markets mostly use news headlines or news data to capture the market reaction to daily events. Incorporating this information into models enables medium and short-term exchange rate predictions; whereas by using only macroeconomic variables as features, only long-term prediction is available. Although the studies that incorporate news data into exchange rate prediction show improvements over models that only make use of macroeconomic variables, they miss out on events that occur with higher frequency.
Long term exchange rates are affected by macroeconomic fundamentals such as trade balance, purchasing power parity, inflation, whereas short term fluctuations in exchange rates cannot be easily predicted since there are no high frequency macroeconomic variables that can be used in prediction. Short term variations in exchange rates might be caused by some unobservable fundamentals that act like a noise such as risk premium of a currency. However Wang (2008) show that they do not have a role in predicting long term variations. One improvement over the models with fundamentals was when Meese and Rogoff (1983) found that a random walk model was as useful as predicting the exchange rates as fundamentals, which suggests that for an accurate prediction relying only on fundamentals is not sufficient.

In addition to these monetary models, there are two more theoretical models that have been used to predict the exchange rates: overshooting and portfolio balance models. The overshooting model by Dornbusch (1976) states that exchange rates overshoot their long run values temporarily when the fundamentals change. Portfolio balance model treats currencies like financial assets and hence the exchange rates are determined by the demand for and the supply of the foreign and domestic currencies (Branson et al., 1977). However, these models were also shown to be inaccurate and insufficient empirically (Backus, 1984). The failures of the theoretical models call for more technical analyses in order to predict especially short term movements in the currency values.

There are also many empirical studies that make use of econometric models such as GARCH, VAR, and spectral analysis for exchange rate prediction, however they are still not sufficient for medium and short term prediction. One reason why these empirical models are insufficient is because they do not include the sentiment into their analyses. Exchange rates are shown to be affected by market sentiment instead of the fundamentals (Hopper, 1997), and hence for an accurate prediction, market sentiment should also be used. This paper addresses this gap in the literature, by using natural language processing for sentiment and Dirichlet Process Mixture Model for topic extraction to predict short term movements in the
Previous studies that aim at predicting exchange rate by extracting sentiment from text data have used news or news headlines as their features. This paper takes one step ahead and uses Twitter trending topics data instead of news. Another unique approach taken in this paper is that the trending topics are taken irrespective of whether the tweets explicitly mention the currencies under consideration, whereas previous studies analyze the tweet that explicitly mention the stocks or the currencies that they are interested in. This approach enables to capture any implicit affect of the trending topics on the currency values.

Trending topics come from 400 million potential users, which represent a much greater variety of sentiment. The data comes from an online dataset, where trending topics in the world are fetched from Twitter every ten minutes since July 2013. Twitter trending topics are words, phrases and hashtags that suddenly increase in popularity, rather than being tweeted at a high volume (Xu, 2015). Trending topics are a good reflection of the concerns, interests, ideas and feelings of the users. Twitter trending topics data hence provides a good source of high frequency information, which would improve especially the short-term or intra-day exchange rate predictions. The advantage of using the Twitter trending topics data set is that it embodies news as well as people’s reactions to the news. Hence, it also provides a reflection of sentiment or mood in the economy. The datasets used in the economics literature include only the news or the mood, but not both. This project will help fill in this gap in the literature; by exploring natural language processing using Twitter trending topics data, which incorporates both news and sentiment, to forecast the movements in the foreign exchange markets.

The main variables of interest are Euro/ US Dollar exchange rate, GBP/ US Dollar exchange rate and Japanese Yen/ US Dollar exchange rate, and the short term movements of these rates are predicted using a Dirichlet Process Mixture Model. This paper is organized

\[1\]The data are being collected for the website: [http://tt-history.appspot.com/](http://tt-history.appspot.com/)
as follows. Related studies are discussed in Section-2. Section-3 introduces the data set and the preprocessing methods. Section-4 presents the model. Results are given in Section-5. Section-6 concludes.

2 Related Literature

This paper contributes to two different strands of literature in two disciplines: exchange rate prediction and machine learning. This is the first paper that makes use of data from social media to predict exchange rates with an unsupervised topic clustering model. Studies available in this literature mostly make use of news headlines as their data, and use neural networks or ensembles as their models. The novelty of this paper in exchange rate prediction literature is using Twitter data and also employing a Dirichlet process mixture algorithm to cluster the sentiment-based topics to use them in a time series model to predict the short term exchange rates. This paper is also the first paper that makes use of unsupervised learning\(^2\) in topic extraction, in exchange rate prediction using natural language processing.

There are many studies that make use of text data for short term exchange rate prediction. However, none of them uses a similar data set or methodology as employed in this paper. All papers that predict exchange rate from text data employ news data or news headlines, and although their methods vary from neural networks to ensembles to latent Dirichlet allocation, Dirichlet process mixture models are not as commonly used.

There are two main strands of machine learning literature that aim to improve upon the current exchange rate prediction. Using neural networks in exchange rate prediction has been a popular method since the last two decades, and the advances in deep learning computing techniques sparked interest in neural networks for the last five years. However, the increasing popularity of microblogs and online news started a new era in information

\(^2\)Unsupervised learning refers to the applications of machine learning techniques on data without having pre-specified outputs. This will be explained in more detail in the following sections.
technology; which makes use of text data and the sentiment from the text data. This paper falls into the second strand which uses text mining techniques in time series prediction.

2.1 Neural Network Models in Exchange Rate Prediction

Artificial neural networks are machine learning models that are inspired by the information processing of nervous system of animals. A basic neural network model consists of three sets of interconnected nodes: input, hidden output. The data enters in the input node and is outputted usually as a classifier from the output node.

Kaastra ad Boyd (1996) and Refenes et al. (1993) are two of the first studies which apply neural network models to financial and economic time series prediction. Kuan and Liu (1995), Hann and Steurer (1996), Tenti (1996), Muhammed and King (1997), Gencay (1999) and Yao and Tan (2000), are among the many earlier studies that show that using neural networks improve the prediction performance, on exchange rate data up to weekly frequency. More recent studies not only use higher frequency data but also more advanced techniques such as particle swarm optimization, fuzzy networks (both stochastic and non-stochastic), hybridization techniques with neural networks, and gene expressions (Sermpinis et al. (2013), Sermpinis et al. (2012a), Bahrepour et al. (2011), Ni and Yin (2009), Valenzuela et al. (2008) and Sermpinis et al. (2012b)).

There are many advantages of using these computationally advanced techniques due to factors such as nonlinearity of the time series, latency, noisy time series (Pailit and Popovic, 1995), and also due to their high prediction accuracy. For example, Evans et al. (2013)...

---

3There are of course many other machine learning techniques used in exchange rate prediction, such as classification and ensembles; however the only main two are discussed here. For instance, Talebi et al. (2014) use Bayesian voting to find ensemble trends in forex market and show improvement over models with only ensembles.


5These techniques are even more commonly applied in finance literature. For example, for applications in stock market forecasting, see Atsalakis and Valavanis (2009a), Atsalakis and Valavanis (2009b), Kim and Shin (2007) and Hadavandi et al. (2010).
obtain 72.5% intra-day prediction accuracy, and 23.3% annualized net return by employing artificial neural networks and genetic algorithms, and they also show that the forex series are not randomly distributed. Using a recurrent Cartesian Genetic Programming evolved artificial network, Rehman et al. (2014) are able to reach up to 98.872% prediction accuracy. Although neural networks result in such high accuracy, one major drawback is that these techniques are not intuitive and often are not easy to express with models.

2.2 Text Mining in Exchange Rate Prediction

Text data based time series prediction models are not as computationally complex, and they often are more intuitive compared to the neural networks based studies. These models became popular with the availability of big digital data and the tools to analyze these big data, such as text mining algorithms. Collecting high volumes of digital data and extracting information for description and prediction purposes is called knowledge discovery in databases (Fayyad et al., 1996). Text mining is a part of a knowledge discovery in databases, which is concerned with extracting and enumerating patterns in the text data to be used in further analyses. Text data resources vary from online news websites to microblogs or social media accounts such as Twitter\(^6\) and Facebook to customer reviews on websites such as Amazon and Yelp.

These sources provide unstructured qualitative data which require a careful pre-processing; which is especially the case for the social websites. The reason is that there are low barriers to entry to these websites, which makes it very easy for users to publish any information of their choice (Mitra and Mitra, 2011). This leads to misinformation due to factors that are often encountered in social media and online reviews such as use of sarcasm\(^7\), trolling\(^8\),

\(^6\)There is a very interesting study by Kwak et al. (2010) which analyzes the topological characteristics of Twitter and finds it to be closer to the online news media rather than social networks.

\(^7\)For a list of sarcastic Amazon product reviews, for example, see [http://www.businessinsider.com/stupidest-amazon-product-reviews-2013-1](http://www.businessinsider.com/stupidest-amazon-product-reviews-2013-1).

\(^8\)Trolling is posting comments or opinions irrelevant to the context, and is especially encountered very
entertain the readers as well as the writer, or giving fake opinions about businesses or people in return for money or discount. To avoid using wrong information the pre-processing should be done in a way to detect the real meaning of the sentences.

In order to obtain the real meaning from a text, natural language processing (NLP) techniques are commonly used. NLP is the first step in a study which uses text data. It enables computer human interaction to understand and manipulate text or speech data (Chowdhury, 2003). With these techniques, researchers are able to extract the real meaning, sentiment and sarcasm from a text; as well as check the language and grammar and suggest corrections.

Once the correct meaning is extracted from text documents, data are ready to be applied the machine learning techniques. Models that use news data in exchange rate prediction are more common than the ones with social media data. This is due to mostly because news data are more informative and require less preprocessing since the language and the grammar are often correct. Peramunetilleke and Wong (2002) use news headlines to predict intra-day movements in exchange rates and their model outperforms the random walk model.

In another attempt to predict short to middle term movements, Zhang et al. (2005) use a framework where they use the news instead of only the headlines and classify the news as good and bad news. These studies use historical data whereas there are studies that aim to do real time prediction from news data as well. The end results of some of these studies are or will be available as a program to users. Forex-Foreteller by Jin et al. (2013) and the

---

9Luca and Zervas (2013) has an empirical study on Yelp review fraud which shows that the fraud reviews are very common and their extend depend significantly on business characteristics.

10Resorting to NLP with semi-supervised learning, Davidov et al. (2010) are able to recognize sarcasm in Twitter tweets and Amazon reviews with F-scores of 0.83 and 0.78 respectively.

11Einstein (2013) discusses and compares performances of various techniques to detect bad language on social media and Baldwin et al. (2013) looks at the linguistic noise in social media text.

12For the applications of news mining on stock markets, see Feuerriegel and Prendinger (2015), Leinweber and Sisk (2011), and Minev et al. (2012).
Curious Negotiator by Zhang et al. (2007) are two examples of automated exchange rate movement predictors from news data.

The methodology in these studies is to predict movements in exchange rates depending on the negativity or positivity of news. There is, however, more information that can be extracted from news or any text data. These data are usually subjective data, and extracting this subjectivity, namely “sentiment”, is called sentiment analysis or opinion mining. The sentiment in a document can be extracted using NLP and computational linguistics to be used in further analyses.

Sentiment analysis has become a popular forecasting tool in finance with the rise of the social media. The seminal work by Bollen et al. (2011) show that by using sentiment, classified as calm, alert, sure, vital, kind and happy; the prediction accuracy in the movements of the Dow Jones Industrial index increases up to 86.7%. Zhang et al. (2011) uses more sentiment classes and extends the framework to predict Dow Jones, NASDAQ, S&P 500 and VIX. Google also has a patent by Bollen et al. (2013), granted in 2013 which applies a self-organizing fuzzy neural network model on the public mood to predict the Dow Jones Industrial Index. Sentiment analysis is being commonly used in other fields as well. Tumasjan et al. (2010) is an important work in political science where they apply sentiment analysis from Twitter to predict elections. Public health also benefits from sentiment analysis. For example, Paul and Dredze (2011) has an interesting work where they predict flu trends from Twitter sentiment.

Although sentiment analysis is a very common tool for these fields, there are only a few studies that make use of it in exchange rate prediction. Using sentiment with news data is even less common compared to the ones with social media data. Nassirtoussi et al. (2015) is one of the few studies that incorporate sentiments into forex prediction from news data.

---

13 Automated news based prediction is very common in finance also. For stock price prediction, see Mittermayer and Knolmayer (2006) for an application called NewsCATS and Hagenau et al. (2012).
14 Liu (2012) provides a very thorough reference for sentiment analysis and its application in various fields.
They use a multi-layer algorithm to solve the co-referencing in text, then to obtain sentiment and finally to update the model with the latest information. With this algorithm they can reach to an accuracy level of 83.33%.

The studies that use social media data in forex prediction almost always work within a subset of tweets that the authors define to be relevant to the currency whose rate is to be forecasted. Papaioannou et al. (2013) forecast hourly EUR/USD exchange rate using a neural network algorithm with Twitter sentiment, and their model outperforms the random walk model. Their methodology is to take tweets that contain “EUR/USD” symbol, and to extract their sentiment, which is different than the framework in this paper. Here, the sentiment from all trending topics are extracted regardless of whether they contain the symbol of the currency in question, and are clustered in an unsupervised manner. In a similar study, Janetzko (2014) forecasts the EUR/USD exchange rate, which again outperforms the random walk model. Similar to Papaioannou et al. (2013), he also preselects the tweets, using a list of concepts and names related to Eurozone and sentiments. In a more supervised study, Ozturk and Ciftci (2014) predict the movements in USD/TRY by extracting a buy, sell, neutral sentiment from tweets and then employing a logistic regression. This paper is novel in its approach as it picks tweets without assuming a predefined relationship, and clusters them using an unsupervised Dirichlet process mixture model. This way, selected tweets do not impose anything on the clusters, and hence the analysis is not limited to the researcher’s definitions on what should be related to the currency rates under consideration.

### 2.3 Topic Models

Once sentiment is extracted from data, the usual next step is to cluster the documents depending on their sentiment. Some studies prefer to do classification instead of clustering, where classification is a supervised algorithm that classifies documents depending on whether they fit into pre-defined classes or not. Clustering is unsupervised; the algorithm lets the
data to decide how documents should be clustered together, without any pre-specified class labels. The researcher may decide to limit the number of clusters to a fixed number, or that could be decided by the data as well.

Topic models are the generative models that define documents as mixtures of different topics. Topics of a document is usually decided by the inverse frequency of the given words. The reason for this is that the most frequent words in a document are usually unimportant words whereas the words about the topic of the document are mentioned only a few times. Since the aim is to extract the topic in a document, the ordering of the words in the document does not matter. The most common practice to find the frequency of words is using the Bag of Words model. In this model, occurrence count of each word is recorded in a big sparse matrix, where each column corresponds to every single word that exists in all documents (corpus\textsuperscript{15}), and the rows correspond to documents. Hence, each entry shows how many times a word specified in a given column occurred in the document specified in the row. The resulting matrix, called a document-term matrix, is populated by zeros since only some of the words of a document are present in the corpus. Document-term matrix is then used as the input in the topic clustering algorithm.

The unsupervised topic clustering models are defined by a set of hidden topics. These topics represent the underlying semantic structure of the corpus (Ritter and Etzioni, 2010). One of the most popular method is latent Dirichlet allocation (LDA) by Blei et al. (2003). In LDA, documents are mixtures over latent topics, which also have distributions over words. Ritter et al. (2011), Zhao et al. (2011), Shirota et al. (2014), Ritter and Etzioni (2010), and Zhao and Jiang are some of the important studies that apply LDA to documents in various disciplines from finance to business. Jin et al. (2013) is the only study that uses LDA for exchange rate prediction from news data.

\textsuperscript{15}The collection of all documents under study are called a corpus. Hence the columns of the bag of words matrix are given by every word that exist in the corpus. In this study, each trending topic is considered a document, and the corpus is the all trending topics available in the data set.
Although LDA is commonly used, it is not suitable for the analysis in this paper due to two main reasons. First, the trending topics data consist of very short text, often even less than the 140 character limit imposed by Twitter. LDA is a more appropriate model for longer documents. Secondly, for LDA, the researcher needs to assume and specify a number of topics for the corpus. However, in this study the aim is to have a completely unsupervised algorithm, which decides on the number of topics in the corpus itself. Hence, this study resorts to Dirichlet process mixture models (DPMM) which are more suitable for short text data and the researcher does not necessarily have to assume a predetermined number of topics for the corpus.\footnote{Neal (2000) is an excellent resource for DPMM algorithms for different cases of priors and sampling methods. Yu et al. (2010) and Yin and Wang (2014) apply DPMM to short news and tweet data. This paper follows the methodology of Si et al. (2013) closely. In that study, tweets that contain the symbols of certain stocks (given by the $ sign) are crawled from Twitter, whose sentiment are then used in a DPMM model to create a sentiment based time series. The stock prices are then predicted using a VAR on these sentiment based time series. This study differs from that in some important aspects. That study only takes the tweets that are relevant to the stocks whose prices to be predicted, however here all trending topics are used regardless of whether they are related to the currencies in consideration or not. Hence, the sentiment extracted from the data is a more general sentiment. Also, in that study full tweets are used, which are around 140 characters whereas here the trending topics are much shorter than that since they are often given by a word or two. This makes the analysis more difficult since the tweets are clustered based on very short text (documents).} Neal (2000) is an excellent resource for DPMM algorithms for different cases of priors and sampling methods. Yu et al. (2010) and Yin and Wang (2014) apply DPMM to short news and tweet data. This paper follows the methodology of Si et al. (2013) closely. In that study, tweets that contain the symbols of certain stocks (given by the $ sign) are crawled from Twitter, whose sentiment are then used in a DPMM model to create a sentiment based time series. The stock prices are then predicted using a VAR on these sentiment based time series. This study differs from that in some important aspects. That study only takes the tweets that are relevant to the stocks whose prices to be predicted, however here all trending topics are used regardless of whether they are related to the currencies in consideration or not. Hence, the sentiment extracted from the data is a more general sentiment. Also, in that study full tweets are used, which are around 140 characters whereas here the trending topics are much shorter than that since they are often given by a word or two. This makes the analysis more difficult since the tweets are clustered based on very short text (documents).

Next section describes the data set in more detail and explains the preprocessing and challenges with that step.

\footnote{Also, since there are no pre-specified set of topics, DPMM can be applied to any human language without necessarily knowing the meaning of the words.}
3 Data Set and Pre-Processing

3.1 Data Description

The exchange rate data were gathered from the website called Forex Rate\textsuperscript{17} Historical data were extracted using per minute intervals, for Euro/Dollar, British Pound/Dollar, Dollar/Swiss Franc and Dollar/Yen exchange rates, from 1/1/2013 to 11/14/2015. The following figure shows how the exchange rates change over time.

![Per minute exchange rates from 2013 to 2015](image)

Figure 1: Exchange rates from 2013 to 2015.

The Twitter data used in this study were obtained from a website\textsuperscript{18} which keeps a track of Twitter trending topics and for how long they were trending. The data set includes the trending topics in the world (and in Turkey) from July 2013 to June 2015. The trending topics are fetched and timestamped every 10 minutes. There are 1,048,575 trending topics

\textsuperscript{17}http://www.forexrate.co.uk
\textsuperscript{18}The website \texttt{http://tt-history.appspot.com/} is built by Mustafa Ilhan. The GitHub project can be found at \texttt{https://github.com/mustilica/tt-history}
in the data set; 523,383 of which are the trending topics from the world. In the world subset, which is used in this paper, there are 102,552 unique timestamps and 139,365 unique trending topics. The first few rows of the trending topics in the world data set look as follows:

<table>
<thead>
<tr>
<th>woeid</th>
<th>time</th>
<th>trend</th>
<th>timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>#AnakCiumanGaraGaraSinetronn</td>
<td>1411716752</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>#LiamYouMakeUsHappy</td>
<td>1429925450</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>ÂžgÂ¬rBasÂ±nSusturulamaz</td>
<td>1418579426</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Britney Is My Life</td>
<td>1394316703</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>AAÂ­Â­Â­_Â­_Â­_Â­_Â­_Â­_Â­_Â­_Â­_@Â­</td>
<td>1428485055</td>
</tr>
</tbody>
</table>

Table 1: First 5 rows of the World trending topics data. Data is not sorted to give the reader an idea of the availability of the various characters.

The first column in the dataset indicates row number. The variable in the second column is “woeid”, which is the geographical id assigned to the tweets by Twitter, where woeid = 1 indicates that the tweets were trending in the world. Third row is for the time when the tweets were fetched from Twitter, and they are all equal to 10 since the trends are fetched every 10 minutes. The trending topics are given in the fourth column, and their timestamps are in the fifth column.

As also seen in the first 5 rows, the data are not clean. The first problem with the data is the foreign characters. The second problem is the hashtagged entries. Since they start with a hashtag, the functions used on strings during the pre-processing part did not work on those entries and hence their hashtags needed to be removed. Finally, most of the hashtagged entries are sentences without spaces, and to make sense of these entries, they needed to be separated into words. Next subsection explains how these problems are tackled.

### 3.2 Pre-Processing

Natural Language Processing (NLP) is the sub-division of artificial intelligence that deals with human-computer interaction. Applications of NLP techniques in research as well as in the industry vary from interpreting text data, speech recognition, part-of-speech tagging to
spam recognition. NLP applies to people’s everyday life from auto text correction to voice commands.

The first step in handling the dataset was removing the hashtags from the entries that begin with a hashtag. Hashtagged entries become trending topics when users use those hashtagged words within their 140 character tweets. Once the hashtags are removed, a binary variable called “hashtag” is created in order not to lose information from that entry being hashtagged or not. The new variables are presented in Table-2.

<table>
<thead>
<tr>
<th>trend</th>
<th>timestamp</th>
<th>nohash_trend</th>
<th>hashtag</th>
</tr>
</thead>
<tbody>
<tr>
<td>#AnakCiumanGaraGaraSinetronn</td>
<td>1411716752</td>
<td>AnakCiumanGaraGaraSinetronn</td>
<td>1</td>
</tr>
<tr>
<td>#LiamYouMakeUsHappy</td>
<td>1429925450</td>
<td>LiamYouMakeUsHappy</td>
<td>1</td>
</tr>
<tr>
<td>#Å zgÅ rBasÅ±nSusturulamaz</td>
<td>1418579426</td>
<td>Å zgÅ rBasÅ±nSusturulamaz</td>
<td>1</td>
</tr>
<tr>
<td>Britney Is My Life</td>
<td>1394316703</td>
<td>Britney Is My Life</td>
<td>0</td>
</tr>
<tr>
<td>ÅÅÅÅ ÅÅÅÅ®Å</td>
<td>1428485055</td>
<td>ÅÅÅÅ ÅÅÅÅ®Å</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Data set with hashtags removed: Contains two new variables, “nohash_trend” and the binary “hashtag”.

After removing hashtags, all entries were turned into lower-case entries since the functions used in the next steps consider all text with capital letters as meaningful words and disregard them. The next step was to separate the entries with no spaces into words. To do that, every letter combination in the entry is compared against the items in a use specified dictionary<sup>19</sup>. The python code to put spaces into sentences with no spaces is available in the appendix.

Table-3 shows the data set which includes the lower-cased and separated entries after running this code.

Also, to reduce noise, only English entries are selected. This was done using the “langid”<sup>20</sup>.

<sup>19</sup>The dictionary used here is the one called “words by frequency” which includes the most frequently and commonly used English words, including different verb forms and slang words. The dictionary is available at: https://raw.githubusercontent.com/aliseambusher/columbus/master/words-by-frequency.txt. The function used to separate the entries with spaces is “infer_spaces()” by Mark Hal (from Mankind Software) and can be found at https://github.com/mankindsoftware/tweetsec/blob/master/lib/wordFinder.py. The second function “put_spaces” was used to apply “infer_spaces” to only entries with no space.

<sup>20</sup>The langid module was developed by Lui and Baldwin (2012), and can be obtained at https://github.com/saffsd/langid.py.
module in Python, in which the “langid.classify()” function returns the expected language of a given string with its probability of being in that language. After removing the non-English entries, there are 196,567 rows left in the dataset.

### 3.2.1 Stop-Word Removal

Stop-words are the words which do not add any meaning to the sentiment of the sentences, but rather help to make grammatical sense, such as “the”, “a”, “this” etc. Removing these words results in a more refined data for sentiment analysis purposes. Stop-words were identified by the “stopwords” corpus in the nltk module, and were removed by searching each entry for these words. Table-4 shows the data set which includes the entries with no stop-words, which are sorted in an ascending order of the timestamps.

<table>
<thead>
<tr>
<th>trend</th>
<th>timestamp</th>
<th>trend with stop-word removed</th>
<th>hashtag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monumental</td>
<td>1373234195</td>
<td>monumental</td>
<td>0</td>
</tr>
<tr>
<td>Andy Murray</td>
<td>1373236603</td>
<td>andy, murray</td>
<td>0</td>
</tr>
<tr>
<td>PRI</td>
<td>1373250449</td>
<td>pri</td>
<td>0</td>
</tr>
<tr>
<td>PREP</td>
<td>1373258875</td>
<td>prep</td>
<td>0</td>
</tr>
<tr>
<td>Lanata</td>
<td>1373259477</td>
<td>lanata</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4: Data set with stop words removed.
3.2.2 Tokenization

Tokenization is the method of making all words in a sentence an independent entry such as words, phrases or symbols. Before tokenization, each entry in the data set were one string including multiple words separated by spaces. After tokenization, each entry becomes a list of strings (where strings are the words), which are separated by commas. For tokenization, “word_tokenize” was used from the nltk package. Table-5 shows the data set with tokenized entries.

<table>
<thead>
<tr>
<th>trend</th>
<th>timestamp</th>
<th>tokenized trend</th>
<th>hashtag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monumental</td>
<td>137324195</td>
<td>['monumental']</td>
<td>0</td>
</tr>
<tr>
<td>Andy Murray</td>
<td>137326603</td>
<td>['andy', 'murray']</td>
<td>0</td>
</tr>
<tr>
<td>PRI</td>
<td>137325049</td>
<td>['pri']</td>
<td>0</td>
</tr>
<tr>
<td>PREP</td>
<td>1373258875</td>
<td>['prep']</td>
<td>0</td>
</tr>
<tr>
<td>Lanata</td>
<td>1373259477</td>
<td>['lanata']</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5: Tokenized data. -Data are sorted in an ascending order of the timestamps.

3.2.3 Stemmer

Stemming is a method used for collapsing the words into distinct forms. This was needed to obtain the words with the same roots as well as combining derivationally related words as one single word. There are three main stemmers available in nltk module (Porter, Lancaster and Snowball). Porter stemmer, by Martin Porter is the most commonly used stemming algorithm, probably due to being one of the oldest stemming algorithms. Although it is a popular algorithm, since it is computationally intensive, it was not used in this paper. Lancaster stemmer is considered to be an aggressive stemmer, and often returns unintuitive words. It is often used due to being the fastest stemming algorithm. Snowball, again by Martin Porter, is a slightly faster version of the Porter Stemmer, and less aggressive than Lancaster stemmer. In this paper Snowball stemmer was used due to being fast enough and not aggressive. This stemmer returns unicoded string values, which were encoded to ASCII
after stemming.

Table-6 shows the data set which includes the lower-cased and separated entries.

<table>
<thead>
<tr>
<th>trend</th>
<th>timestamp</th>
<th>tokenized trend</th>
<th>hashtag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monumental</td>
<td>1373234195</td>
<td>['monument']</td>
<td>0</td>
</tr>
<tr>
<td>Andy Muray</td>
<td>1373236603</td>
<td>['andi', 'murray']</td>
<td>0</td>
</tr>
<tr>
<td>PRI</td>
<td>1373250449</td>
<td>['pri']</td>
<td>0</td>
</tr>
<tr>
<td>PREP</td>
<td>1373258875</td>
<td>['prep']</td>
<td>0</td>
</tr>
<tr>
<td>Lanata</td>
<td>1373259477</td>
<td>['lanata']</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6: Tokenized data. -Data are sorted in an ascending order of the timestamps.

3.2.4 Sentiment Lexions

In order to extract the sentiment from the stemmed trending topics, the dictionary developed by Loughran McDonald was used\[21\]. This extensive dictionary contains more than 80,000 words, and was updated in 2014. The words included in this lexicon contain at least two letters, so one letter words are excluded since they are not regarded as critical to the content. Table-A.1 in the appendix prints the first 70 rows of this lexicon.

The lexicon includes the word count, proportion of the word compared to the other words in the scanned documents, the average proportion and the standard deviation, number of documents the word occurred, whether the word has a negative, positive, uncertain, litigious, constraining, superlous or interesting sentiment. The variable modal indicates how strong the modal is, taking 3 possible values where “1” corresponds to the “strong modal” (such as “always”), “2” corresponds to moderate modal” (such as “usually”), and “3” corresponds to “weak modal” (such as “almost”). Irregular verb shows whether the verb is irregular, syllables indicate how many syllables the word has. Harvard IV is the sentiment

from derived from “Harvard Psychosociological Dictionary”, which is constructed from applications in psychology and sociology. Finally, source shows which dictionary the authors used, and 2of12inf is the dictionary that includes words without abbreviations, acronyms, or names; and the words are not in the stemmed form in order to capture more content.

In order to make use of this dictionary, it was preprocessed to suit the Twitter dataset. Since this dictionary includes the inflections and different word form for the words originating from the same root, the entries in the dictionary were stemmed using the same algorithm which was used to stem the words in the trending topics dataset. After the words are stemmed, there were multiple entries with the same stem coming from different word forms. These multiple entries were reduced into one single entry. The final sentiment dataset is given in Table-A.3 in the appendix.

4 Model

Topics and words from the data set are exchangeable (de Finetti, 1990) and have a representation as an infinite mixture distribution. When exchangeability is satisfied, the most commonly used topic clustering method is Latent Dirichlet Allocation (LDA), following Blei et al. (2003). According to LDA, words are the basic unit of discrete data who are of documents, whose collection make up the text corpus. LDA is a generative probabilistic model where documents are random mixtures over latent topics, and each topic has a distribution over words. Although it is common practice to use LDA in this context, for 2 main reasons it is not suitable for the analysis in this paper.

Firstly, LDA requires large documents and hence a large corpus. In this study, each trending topic is considered to be a document, which are limited to 140 characters by Twitter, and in fact are almost always shorter than 140 characters. Secondly, LDA requires a pre-specified number of topics to cluster the documents into. However, the topics in the trending
topics dataset are too broad to pre-specify a number. Hence, in order to avoid limiting the analysis to a pre-specified number of topics and to be able to treat all tweet entries as one document, a Dirichlet Process Mixture model was used instead.

4.0.5 Dirichlet Process Mixture Model

The model that would be applicable the short documents (tweets) and high number of topics is a Dirichlet Process Mixture Model. This model accommodates very short documents and hence it is widely used in studies that make use of Twitter data, and also does not limit the analysis to a pre-specified number of topics. A Dirichlet process is a probability distribution over probability distributions. Dirichlet process mixture models are hierarchical models that consist of $K$ Dirichlet components. $K$ can be a finite number; in that case there are a finite number of components (in the scope of this paper, "topics"), or $K$ can be taken to infinity to represent infinite number of topics (clusters).

Following the Algorithm 3 from Neal (2000), topics are clustered as follows:

- Each trending topic (document) $x$, exchangeably and independently comes from a mixture of distributions $F(\theta)$, over the parameter $\theta$:

$$x_i|\theta_i \sim F(\theta_i)$$

- Each $\theta_i$ is the parameter of the mixing distribution $G$:

$$\theta_i|G \sim G$$

- where $G$ is generated from a Dirichlet process prior with a concentration parameter $\alpha$ and a base measure $\zeta$:

$$G \sim DP(\zeta, \alpha)$$
• Then, $x_i$ can be represented as:

$$x_i \sim \lim_{k \to \infty} \sum_{k=1}^{K} \pi_k p(x_i | y_i = k)$$

where $y_i$ is the cluster label assigned to document $x_i$ and $\pi_k$ is the weight of the corresponding cluster ($\pi_k \geq 0$ and $\sum_k \pi_k = 1$), $K$ is the number of topics, and $p(x_i | y_i = k)$ is the topic models $\{\phi_k\}_{k=1}^{K}$. Since the trending topics entries are very small (even less than the 140 character limit by Twitter), there is only one topic $y_i$ in each trend $x_i$.

The number of topics, $K$ is an unfixed number and is estimated from the given timestamp’s trending topics. This is a similar framework to that of Si et al. (2013), where the authors estimate topics for each day whereas here instead of each day, in order to estimate the topics for each timestamp; in order to capture the intra-day changes. Similar to that framework, it is also acknowledged in this paper that the neighboring timestamps might have the same or related topics and hence they evolve with time. For this reason, the model has a dynamic generative process. The observed trending topics $\{x_{n,i}\}_{i=1}^{T_n}$ are grouped for each timestamp $T_n$, $(n : 1, .., N)$, and are generated by the latent topics $\{\theta_{n,i}\}_{i=1}^{T_n}$. The topics are generated for each timestamp $T_n$, so a DPM model is built on every time stamp. The topics learned in the previous timestamp are used as priors for the topics in the current timestamp, which is the continuous nature of this model.

For the very first timestamp, there is no previous topic that was learned that could work as prior. Although the model is a continuous DPM, for the first entry, a standard DPM was needed to be used where only two parameters that start the generating process are $\alpha_0$ and $\zeta_0$. For all other timestamps at any time $n$, the priors are $\alpha_n$, $\zeta_n$ and $G_{n-1}$. Then, the
probability that $x_i$ belongs to cluster $k$ is given in two components as follows:

$$P(y_i = k|y_{-i}, x_i, \alpha, \zeta) \propto P(y_i = k|y_{-i}, \alpha)P(x_i|x_{-i}, y_i = k, y_{-i}, \zeta)$$ (1)

The first component can take four different values depending on whether $k$ is a new topic or not, and whether it takes a symmetric base prior or a topic as a prior:

$$P(y_i = k|y_i, \alpha) = \begin{cases} \frac{\alpha}{l-1+\alpha} & \text{if } k \text{ is a new topic and takes a symmetric base prior} \\ \frac{\alpha \pi_{n-1,k}}{l-1+\alpha} & \text{if } k \text{ is a new topic and takes a topic as prior} \\ \frac{l_k^{-1}}{l-1+\alpha} & \text{if } k \text{ is an existing topic} \end{cases}$$ (2)

The second component can be rewritten as:

$$P(x_i|x_{-i}, y_i = k, y_{-i}, \zeta) = \int P(x_i|\theta_k) \left[ \prod_{j \neq i, y_j = k} P(x_j|\theta_k) \right] G(\theta_k|\zeta) d\theta_k$$ (3)

The following figure shows the process of the model graphically:

4.0.6 Model Inference

Following Bishop (2006) and Si et al. (2013), a collapsed Gibbs sampling was used with hyper-parameters following from Si et al. (2013), Sun et al. (2010) and Teh et al. (2006). Accordingly, the concentration parameter $\alpha$ is taken to be constant at 1, and each base measure follows a Dirichlet distribution, $\zeta_n \sim Dir(\beta)$, where $\beta = 0.5$.

When sampling for the current topic $y_i$, there are two things that can happen. The label of the topic can be a new topic $k^*$ or can be an existing topic $k$. In both cases, there are two different scenarios. If it is a new topic, then $k^*$ can either take a symmetric base prior $Dir(\beta)$, or it can take one of the topics from $\{\phi_{n-1,k}\}_{k=1}^{K_{n-1}}$ learned from tweets $T_{n-1}$, where
$K_{n-1}$ is the number of topics learned at timestamp $n-1$. If $k^*$ is new and takes a symmetric base prior, then the posterior becomes:

$$p(y_i = k^* | y_{-i}, x_i, \zeta) \sim \frac{\alpha}{l - 1 + \alpha \Gamma(\beta|W|) + l_i} \frac{\Gamma(\beta|W|)}{\Pi_{w=1}^{W} \Gamma(\beta + l_{i,w})} \frac{\Pi_{w=1}^{W} \Gamma(\beta + l_{i,w})}{\Pi_{w=1}^{W} \Gamma(\beta)}$$

(4)

where $|W|$ is the size of the vocabulary, $l_i$ is the length of tweet $x_i$, and $l_{i,w}$ is the “term frequency”\footnote{Term frequency of a word shows how frequently that word occurs in the given document.} of word $w$ in $x_i$. If $k^*$ is new but takes a topic from $\{\phi_{n-1,k}\}_{k=1}^{K_{n-1}}$ from the learned topics from $T_{n-1}$, then the posterior becomes:

$$p(y_i = k^* | y_{-i}, x_i, \zeta) \sim \frac{\alpha \pi_{n-1,k}}{l - 1 + \alpha \Gamma(\beta|W|) + l_i} \frac{\Gamma(\beta|W|)}{\Pi_{w=1}^{W} \Gamma(\beta \phi_{n-1,k}(w) + l_{i,w})} \frac{\Pi_{w=1}^{W} \Gamma(\beta \phi_{n-1,k}(w))}{\Pi_{w=1}^{W} \Gamma(\beta \phi_{n-1,k}(w))}$$

(5)

where $\phi_{n-1,k}(w)$ is the probability of word $w$ in the topic $k$ of the previous day.

Figure 2: Continuous Dirichlet Process Mixture Model
If \( k \) is an existing topic, then its prior is already known. In this case, \( k \) would either take a symmetric base prior \( \phi_{n,k}(w) = (\beta + l_{k,w}) / (\beta |W| + l_{k,(.)}) \), where \( l_{k,w} \) is the frequency of word \( w \) in and \( l_{k,(.)} \) is the marginalized sum over all words; or it can take \( \phi_{n-1,k} \) as its prior, which then will be used to calculate \( \phi_{n,k}(w) = (\beta |W| \phi_{n-1,k}(w) + l_{k,w}) / (\beta |W| + l_{k,(.)}) \). Hence, if \( k \) is an existing topic which takes a symmetric base prior, then the posterior becomes:

\[
p(y_i = k | y_{-i}, x_i, \zeta) \sim \frac{l_{-i}^{-i}}{l - 1 + \alpha} \frac{\Gamma(\beta |W| + l_{-i}^{-i})}{\prod_{w=1}^{|W|} \Gamma(\beta + l_{i,w} + l_{-i}^{-i})} \frac{\prod_{w=1}^{|W|} \Gamma(\beta |W| + l_{i,w} + l_{-i}^{-i})}{\prod_{s=1}^{\gamma} \Gamma(\beta + l_{-i}^{-i})} \tag{6}
\]

where \( l_{-i}^{-i} \) is the number of tweets assigned to topic \( k \) except for the current tweet \( x_i \), \( l_{-i}^{-i} \) is the term frequency of the word \( w \) in topic \( k \), excluding the current tweet \( x_i \), and \( l_{-i}^{-i} \) is the marginalized sum over all words in topic \( k \), again excluding \( x_i \). If, on the other hand, \( k \) is an existing topic with the prior \( \phi_{n-1,k} \), then the posterior becomes:

\[
p(y_i = k | y_{-i}, x_i, \zeta) \sim \frac{l_{i}^{-i}}{l - 1 + \alpha} \frac{\Gamma(\beta |W| + l_{i}^{-i})}{\prod_{w=1}^{|W|} \Gamma(\beta \phi_{n-1,k}(w) + l_{i,w} + l_{i}^{-i})} \frac{\prod_{w=1}^{|W|} \Gamma(\beta \phi_{n-1,k}(w) + l_{i,w} + l_{i}^{-i})}{\prod_{s=1}^{\gamma} \Gamma(\beta \phi_{n-1,k}(w) + l_{-i}^{-i})} \tag{7}
\]

Topic weights for each day are calculated as:

\[
\pi_k = \frac{l_k}{\sum_{k'} l_{k'}} \tag{8}
\]

The sampling process for the current topic \( y_i \) given all other topics \( y_{-i} \) is summarized as follows:

### 4.0.7 Incorporating Topic Clusters into Sentiments

In the NLP section, each day’s sentiment in the data set was matched using Loughran McDonald’s sentiment lexicon. In order to simplify the model, only 3 sentiments were extracted from the tweets: negativity, positivity and uncertainty. This sentiments are collected under one variable named “sentiment”, \( S \in \{-1, 0, 1\} \), where the class labels \( c = \{-1, 0, 1\} \) corre-
Algorithm 1  Conditional distribution of $y_i$ given all other assignments $y_{-i}$

1: if $k^*$ is a new topic then there are two candidate priors:
2:   if $k^*$ takes a symmetric base prior $\text{Dir}(\beta)$ then
3:       $p(y_i = k^*|y_{-i}, x_i, \zeta) \sim \frac{\alpha}{l - 1 + \alpha} \frac{\Gamma(\beta|W|)}{\Gamma(\beta|W| + l_i)} \frac{\prod_{w=1}^{W} \Gamma(\beta + l_{i,w})}{\prod_{w=1}^{W} \Gamma(\beta)}$

3:   if $k^*$ takes a topic from $\{\phi_{n-1,k}\}_{k=1}^{K_{n-1}}$ from the learned topics from $T_{n-1}$ then
4:       $p(y_i = k^*|y_{-i}, x_i, \zeta) \sim \frac{\alpha \pi_{n-1,k}}{l - 1 + \alpha} \frac{\Gamma(\beta|W|)}{\Gamma(\beta|W| + l_i)} \frac{\prod_{w=1}^{W} \Gamma(|W|\beta\phi_{n-1,k}(w) + l_{i,w})}{\prod_{w=1}^{W} \Gamma(|W|\beta\phi_{n-1,k}(w))}$

4: if $k$ is an existing topic then the prior is known:
5:   if $k$ takes a symmetric base prior $\text{Dir}(\beta)$ then
6:       $p(y_i = k|y_{-i}, x_i, \zeta) \sim \frac{l_k^{-i}}{l - 1 + \alpha} \frac{\Gamma(\beta|W| + l_{k,(i)})}{\Gamma(\beta|W| + l_i + l_{k,(i)})} \frac{\prod_{w=1}^{W} \Gamma(\beta + l_{i,w} + l_{k,w}^{-i})}{\prod_{w=1}^{W} \Gamma(\beta + l_{i,w}^{-i})}$

6:   if $k$ takes a topic from $\{\phi_{n-1,k}\}_{k=1}^{K_{n-1}}$ from the learned topics from $T_{n-1}$ then
7:       $p(y_i = k|y_{-i}, x_i, \zeta) \sim \frac{l_k^{-i}}{l - 1 + \alpha} \frac{\Gamma(\beta|W| + l_{k,(i)})}{\Gamma(\beta|W| + l_i + l_{k,(i)})} \frac{\prod_{w=1}^{W} \Gamma(|W|\beta\phi_{n-1,k}(w) + l_{i,w} + l_{k,w}^{-i})}{\prod_{w=1}^{W} \Gamma(|W|\beta\phi_{n-1,k}(w) + l_{i,w}^{-i})}$
spond to negative, neutral and positive respectively. The entries whose all three sentiment were zero are assumed to have neutral sentiment.

In order to incorporate topic clusters into sentiment dataset, the usual practice would be to create a topic based sentiment score. Sentiment score for topic $k$ at timestamp $n$ is given as the weighted sum of sentiment (opinion) classes, $c(o)$, weighted by the word probability for topic $k$, $\phi'_{n,k}(o)$:

$$S(n,k) = \sum_{o=1}^{|O|} \phi'_{n,k}(o)c(o)$$  \hspace{1cm} (9)

However, since since each trending topic is considered to be a document, not to lose any information, the weight of words are assumed to be 1. The final dataset is converted to a sparse 0 – 1 matrix, using CountVectorizer from scikit-learn package in Python. The size of this matrix is 34 GB, consisting of 196,567 rows and 20,979 columns. The columns are given by the unique words in the data set; hence there are 20,979 unique words in the corpus. However, most of these words (which make up the columns) were not meaningful or not full words. Instead of working with all words, word count and sentiment (positive, negative, uncertain) at each timestamp is used to cluster the topics. The algorithm divided the tweets into 9 unique topics.

4.1 Sentiment Time Series Prediction

Two different methods are used to predict exchange rates. First, an AR model is fitted to the exchange rate series $\{z_t\}$ from July 2013 to June 2015 in order to obtain baseline values to compare to, without the topics from the tweets:

$$z_t = \alpha_0 + \alpha_1 z_{t-k} + \epsilon_{z,t}$$  \hspace{1cm} (10)

where $\{\alpha\}$ are the model coefficients, $\{k\}$’s are the lags and $\{\epsilon\}$ are white noises.

Table-7 gives the number of observations from each currency during this period. Both
AIC and BIC were considered while choosing the optimal lag, and they both resulted in 1-period lag. This implies that the best lagged predictor of a currency value is its values 10 minutes ago.

<table>
<thead>
<tr>
<th></th>
<th>no. obs.</th>
<th>AIC</th>
<th>BIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR/USD</td>
<td>89389</td>
<td>-1174842.737</td>
<td>-1174814.535</td>
</tr>
<tr>
<td>GBP/USD</td>
<td>89419</td>
<td>-1256379.221</td>
<td>-1256351.017</td>
</tr>
<tr>
<td>CHF/USD</td>
<td>89419</td>
<td>-998922.197</td>
<td>-998893.994</td>
</tr>
<tr>
<td>JPY/USD</td>
<td>89419</td>
<td>-306488.525</td>
<td>-306460.322</td>
</tr>
</tbody>
</table>

Table 7: The minimum AIC and BIC from AR models. Both criteria yield a 1-period lag as optimum.

In order to test against the prediction results against the results from this model, a linear model which includes 1-period lagged value of the currency itself and the 9 topics as 8 dummies are fitted as follows:

\[
z_t = \alpha_0 + \alpha_1 z_{t-k} + \alpha_2 D_1 + \ldots + \alpha_9 D_8 + \epsilon_{z,t}
\]  

(11)

where \(\{\alpha\}\) are the model coefficients, \(\{k\}\)’s are the lags, \(\{D\}\)’s are the dummies for the topics and \(\{\epsilon\}\) are white noises.

Secondly, following Si et. al (2013), after clustering the tweets, a VAR model is built for the tweet series \(\{x_t\}\) and each exchange rate series \(\{z_t\}\).

\[
x_t = \nu_11 x_{t-1} + \nu_12 z_{t-1} + \epsilon_{x,t}
\]

\[
z_t = \nu_21 z_{t-1} + \nu_22 x_{t-1} + \epsilon_{z,t}
\]

(12)

where \(\{\nu\}\) are the model coefficients and \(\{\epsilon\}\) are white noises.
From the timestamps with multiple trending topics, the ones with the most word count is taken to increase the information that can be obtained from the tweets data. This reduced dataset includes 89428 entries with 10 minute intervals from July 2013 to June 2015. Before fitting the VAR, the series are divided into training and test subset in order to test the predictive power of the models. 70% of the observations are taken as training set (from July 2013 to November 2014), and VAR is trained on this set. The remaining 30% (from December 2014 to June 2015) is used to test the prediction. Table-8 shows the results from AIC and BIC. While according to BIC the optimal number of lags is 12 for each currency, according to AIC the optimal lags are 29, 29, 28 and 19 for Euros, British Pound, Swiss Franc and Japanese Yen respectively.

<table>
<thead>
<tr>
<th></th>
<th>Lags from AIC</th>
<th>Lags from BIC</th>
<th>AIC</th>
<th>BIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR/USD</td>
<td>29</td>
<td>12</td>
<td>-15.38</td>
<td>-15.37</td>
</tr>
<tr>
<td>GBP/USD</td>
<td>29</td>
<td>12</td>
<td>-15.81</td>
<td>-15.80</td>
</tr>
<tr>
<td>CHF/USD</td>
<td>28</td>
<td>12</td>
<td>-14.72</td>
<td>-14.71</td>
</tr>
<tr>
<td>JPY/USD</td>
<td>19</td>
<td>12</td>
<td>-5.056</td>
<td>-5.048</td>
</tr>
</tbody>
</table>

Table 8: The minimum AIC and BIC from VAR models.- Each criteria yields different number of lags as optimum.

Next section compares the prediction errors from these 4 models.

5 Results

The prediction errors from the AR(1) models during the test period is given in Figure-3. Prediction errors are not too far from zero most of the time, although they are either below or above zero for long intervals instead of fluctuating around zero.

One important exception to this is for Swiss Franc on 1/15/15, where the big spike in the errors is observed (which is also observed in the currency value in Figure-1). This is when the Swiss Central Bank announced after an unscheduled meeting that it would discontinue...
Figure 3: Prediction Errors from AR(1) model during test period. Errors are close to zero but they tend to stay above or below zero instead of oscillating around zero.

keeping the exchange rate at a minimum of 1.20 CHF per Euro, and that it is lowering the interest rate to $-0.75\%$, and changing the target range for the three month Libor further from the $(-0.75\% - 0.25\%)$ interval to $(-1.25\%, -0.25\%)$ interval \footnote{The press release from the Swiss Central Bank can be found at: https://www.snb.ch/en/mmr/reference/pre_20150115/source/pre_20150115.en.pdf}.

In order to improve the prediction over AR(1), the topics are fitted as dummy variables. This improved the sum of squared prediction errors, as well as the errors across time. Figure-4 and Table-9 document these error values.

Prediction errors are smaller compared to those from the AR(1) across time, and they oscillate around zero except for a few observations, and except for CHF on 1/15/15. There seems to be almost no improvement in terms of prediction error on 1/15/15 for CHF with the model with topics compared to that from AR(1), although after that day the error falls back to zero; whereas with AR(1), errors stays above zero for two weeks.

One of the main reasons why the model with topics could not capture this movement is...
Figure 4: Prediction Errors from linear regression during test period with 1-period lagged variables and the topics as dummy variables. Errors oscillate around zero, except for CHF on 1/15/15, and a few other observations.

<table>
<thead>
<tr>
<th></th>
<th>AR</th>
<th>Linear Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR/USD</td>
<td>0.0109</td>
<td>0.0069</td>
</tr>
<tr>
<td>GBP/USD</td>
<td>0.0022</td>
<td>0.0020</td>
</tr>
<tr>
<td>CHF/USD</td>
<td>0.0674</td>
<td>0.0673</td>
</tr>
<tr>
<td>JPY/USD</td>
<td>76.4965</td>
<td>70.7816</td>
</tr>
</tbody>
</table>

Table 9: Sum of squared errors during the test period. Sum of squared prediction errors are smaller when topic dummies are added to the model than in AR(1).

that the Swiss Central Bank meeting was an unannounced emergency meeting and hence the movement in the currency value was probably simultaneous to if not preceding the reaction in the social media\(^\text{24}\). Although the model with the topics could not predict this surprise event, it was still successful to predict the value of the currency right after the event, which

\(^{24}\)This event was trending in Twitter with the hashtag “#Francogeddon”. This hashtag does not appear in the data set used here, probably because it was not trending in the world.
AR(1) failed to predict for two weeks.

Figure 5: Prediction Errors from VAR following AIC criterion.- VAR does not improve the prediction over AR(1).

In addition to the linear model with topics with dummies, VAR models (depending on AIC and BIC) are fitted in order to capture the effects of topics on exchange rates as well as the effects of rates on topics. Figures 5 and 6, and Table-10 show the prediction errors from these models.

Compared to AR(1), or the linear model with dummies and lagged currency values, on average VAR models do not result in better prediction. However VAR models result in a better prediction for the Swiss Franc after the announcement shock. This shows the effect
Figure 6: Prediction Errors from VAR following BIC criterion. VAR does not improve the prediction over AR(1).

<table>
<thead>
<tr>
<th></th>
<th>VAR (AIC)</th>
<th>VAR (BIC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR/USD</td>
<td>93.9337</td>
<td>92.5235</td>
</tr>
<tr>
<td>GBP/USD</td>
<td>45.9247</td>
<td>45.9361</td>
</tr>
<tr>
<td>CHF/USD</td>
<td>53.1797</td>
<td>53.1658</td>
</tr>
<tr>
<td>JPY/USD</td>
<td>58258314.8725</td>
<td>55696777.3534</td>
</tr>
</tbody>
</table>

Table 10: Sum of squared errors from VAR’s during the test period. Sum of squared prediction errors are slightly smaller when lag=12 is picked following BIC.

of sentiment from the topics on the currency value.
6 Conclusion

This paper investigated whether incorporating sentiment extracted from Twitter’s trending topics would improve the intra-day exchange rate predictions. What makes this paper unique is that unlike previous similar studies which only consider tweets that contain the symbol or the name of the currency or stock, it looks at all trending topics irrespective of whether they contain the name or the symbol of the currency. This allows to capture the general sentiment among the users, which implicitly affects the exchange rates.

One of the important results from this study is that although this approach cannot predict surprise events, since events need to be heard by the users before a user sentiment can be obtained, it performs better than the benchmark AR(1) model when predicting what will happen after such unexpected events. Moreover, when the topics are added as dummies to the lagged values of the currencies, the prediction errors are found to be lower than the ones from using only the lagged values AR(1). This shows the implicit effect of the sentiment among users on the currency values.

VAR models performed better after the unexpected shock to Swiss Franc, but they did not result in a improved prediction compared to the benchmark AR(1) on average. The reason why VAR’s could not over-perform AR(1) on average could be due to the choice of the data set. The fact that the trending topics can only include an implicit sentiment about the currencies worked well for the linear model; however for VAR to work it might require to use tweets that explicitly mention the currencies, or the countries in which those currencies are used. In fact, only using the tweets that explicitly mention the currencies is the general approach taken in most of all previous studies. However since that approach ignores the sentiments that might implicitly appear in the tweets, it was not preferred in this study.

A further step that can be taken after this paper would be to expand the dataset to include the whole tweets instead of only the trending topics as using only trending topics
had many limitations. First of all, the trending topics are too short in length, almost always shorter than the 140 character limit imposed by Twitter. This makes it difficult to extract the extract sentiment from those short expressions. Moreover, most clustering algorithm such as LDA do not work for such short text. Secondly, the trending topics could be in any language, which needed to be detected by the algorithm. Although the language detection algorithm is a powerful one, due to typos, extra or missing characters or numbers it might not have worked as well as desired. Moreover, some of the expressions with missing or extra characters, although the algorithm corrected for the ones that it could detect, were still present in the data set after preprocessing, which also made sentiment extraction and clustering more difficult.

Overall, despite the linguistic limitations from the Twitter’s trending topics, this paper successfully showed that by incorporating sentiment from trending topics intra-day exchange rate predictions can perform better than the predictions from AR(1). The next step would be to improve the predictions from VAR. Using longer tweets or taking the whole tweets which contain a trending topic would be a good starting point for this.
References


web, ACM, 591-600.


Appendix
|--------|----------------|------------|------------|------------|---------|------------|------------|------------|-------------|-------------|--------------|---------------|-------------|---------|---------|-----------|------------|-----------|--------|

Table A.1: First rows of the sentiment lexicon used in this study.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aardvark</td>
<td>3.23E-07</td>
<td>0.000571784</td>
<td>0.000482292</td>
<td>0.000483819</td>
<td>0.000228539</td>
<td>7.12E-10</td>
<td>1.666666667</td>
<td>0.000492471</td>
<td>1.56E-05</td>
<td>0.000604568</td>
<td>0.000572326</td>
<td>abomin</td>
<td>3.88E-09</td>
<td>0.000343182</td>
<td>1.09E-05</td>
<td>0.000228539</td>
<td>7.12E-10</td>
<td>1.666666667</td>
<td>0.000492471</td>
<td>1.56E-05</td>
<td>0.000604568</td>
<td>0.000572326</td>
</tr>
</tbody>
</table>

Table A.2: Processed Sentiment Lexicon — Dictionary is reduced to a smaller dataset by using the stemmed version of the words and grouping the ones with the same root.
<table>
<thead>
<tr>
<th>hashtag</th>
<th>low_nohash_trend</th>
<th>nohash_trend</th>
<th>spaced_trend</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>monumental</td>
<td>Monumental</td>
<td>monumental</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>andy murray</td>
<td>Andy Murray</td>
<td>andy murray</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>pri</td>
<td>PRI</td>
<td>pri</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>prep</td>
<td>PREP</td>
<td>prep</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>lanata</td>
<td>Lanata</td>
<td>lanata</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>timestamp</th>
<th>trend</th>
<th>woeid</th>
<th>en</th>
<th>nostop_trend</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Monumental</td>
<td>1</td>
<td>1</td>
<td>monumental</td>
</tr>
<tr>
<td>1</td>
<td>Andy Murray</td>
<td>1</td>
<td>1</td>
<td>andy murray</td>
</tr>
<tr>
<td>2</td>
<td>PRI</td>
<td>1</td>
<td>1</td>
<td>pri</td>
</tr>
<tr>
<td>3</td>
<td>PREP</td>
<td>1</td>
<td>1</td>
<td>prep</td>
</tr>
<tr>
<td>4</td>
<td>Lanata</td>
<td>1</td>
<td>1</td>
<td>lanata</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>tokenized_trend</th>
<th>str_trend</th>
<th>tokenlist_trend</th>
<th>stemmed_trend</th>
<th>stem</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>monumental</td>
<td>monumental</td>
<td>'monumental'</td>
<td>'monument'</td>
</tr>
<tr>
<td>1</td>
<td>andy murray</td>
<td>andy murray</td>
<td>'andy', 'murray'</td>
<td>'andi', 'murray'</td>
</tr>
<tr>
<td>2</td>
<td>pri</td>
<td>pri</td>
<td>'pri'</td>
<td>'pri'</td>
</tr>
<tr>
<td>3</td>
<td>prep</td>
<td>prep</td>
<td>'prep'</td>
<td>'prep'</td>
</tr>
<tr>
<td>4</td>
<td>lanata</td>
<td>lanata</td>
<td>'lanata'</td>
<td>'lanata'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word Count</th>
<th>Word Proportion</th>
<th>Average Proportion</th>
<th>Std Dev</th>
<th>Doc Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>14717</td>
<td>1.03E-06</td>
<td>2.66E-05</td>
<td>4909</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>250</td>
<td>1.76E-08</td>
<td>1.45E-07</td>
<td>0.000117722</td>
</tr>
<tr>
<td>3</td>
<td>5070</td>
<td>3.56E-07</td>
<td>6.19E-07</td>
<td>8.75E-05</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Negative</th>
<th>Positive</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table A.3: Final version of the Twitter data.