Unemployment rate forecasting using Google trends

Bachelor Thesis in Econometrics & Operations Research

ERASMUS UNIVERSITY ROTTERDAM

ERASMUS SCHOOL OF ECONOMICS

Author: Olivier O. Smit, 415283

Supervisor: Jochem Oorschot

July 8, 2018

Abstract

In order to make sound economic decisions, it is of great importance to be able to predict and interpret macro-economic variables. Researchers are therefore seeking continuously to improve the prediction performance. One of the main economic indicators is the US unemployment rate. In this paper, we empirically analyze whether, and to what extent, Google search data have additional predictive power in forecasting the US unemployment rate. This research consists of two parts. First, we look for and select Google search data with potential predicitive power. Second, we evaluate the performance of level and directional forecasts. Here, we make use of different models, based on both econometric and machine learning techniques. We find that Google trends improve the predictive accuracy in all used forecasting methods. Lastly, we discuss the limitations of our research and possible future research suggestions.

1 Introduction

Nowadays, search engines are intensely used platforms. They serve as the gates to the Internet and at the same time help users to create order in the immense amount of websites and data available on the Internet. About a decade ago, researchers have realized that these search engines contain an enormous quantity of new data that can be of great additional value for modeling all kinds of processes. Previous researches have already proven this to be true. For example, Constant and Zimmermann (2008) have shown that including Google - the largest search engine in terms of search activity and amount of users - query data can be very useful in measuring economic processes and political activities.

One of the most well-known economic features is the business cycle. Thorough understanding of these natural fluctuations in many economic factors is crucial for small companies, large financial institutions and even the governments of entire countries. Every player in the economic world has to take into account the business cycles in its long-term decisions and plans. Here, a very important aspect is the labor market. High unemployment rates inevitably interact with the nation's economy in a negative manner and are strongly correlated with recessions. It is therefore crucial to keep improving the forecasting performance of those macro-economic variables, and in particular the unemployment rate, such that the cycles (and especially their turning points) can be predicted increasingly accurately.

In this paper we examine the predictive power of Google search trends in relation with the US unemployment rate. We aim to show that search engine data can play a valuable role in supporting the traditional flow of data in order to lay the foundations for well-thought and sound economic decisions. For this matter we answer the question whether, and to what extent, Google trends can attribute to performance improvements in predicting the US unemployment rate. In order to provide an answer to this question, we will make various approaches in predicting and evaluating the data. Among them are conventional econometric methods, such as linear regression, and relatively new machine learning techniques.

2 Literature

With the launch of the Google trends platform, a large new set of big data has become available. Search query data are considered to be somewhat different from more conventional data, in the sense that Google searching activity reflects people's thoughts, sympathies and interests, rather than more common, quantifiable actions. In this way, it could be argued that search query data

is at another level of 'depth', because it goes beyond real-world actions and is able to capture some of the people's mental activities. Because of this new type of big data, researchers have been exploring the potential advantages of including these data into all kinds of models, and the first results are promising.

For example, Preis, Moat and Stanley (2013) have found that search query data can be very useful in capturing trading behavior in financial markets. Carrière-Swallow and Labbé (2009) show that search engine data can be helpful in predicting car sales in Chile.

On the other hand, Google trends are also involved in forecasting subjects that are less directly linked to the world of finance and trade. For example, Ginsberg et al. (2009) successfully use search query data in their approach to improve the forecasting of influenza pandemics. Artola, Pinto and De Pedraza García (2015) use Internet activity in order to predict tourism inflows.

Regarding forecasting techniques for the unemployment rates, the inclusion of Google trends in the model has been examined earlier. Askitas and Zimmermann (2009) have incorporated Google trends in their model in order to forecast the German unemployment rate. Based on four different categories of search queries, they evaluate the forecasting power of Google trends.[1] Despite the rather small dataset they have available, they show the potential of forecasting economic cycles with the help of search activity. Choi and Varian (2009) try to model initial claims for unemployment benefits. They include two categories of Google trends in their model, namely 'local/jobs' and 'social services/welfare/unemployment'. Again, the Google trends add extra predictive power to the original model.

With respect to Google trends in combination with artificial neural networks and machine learning, the amount of literature available is smaller. Yu et al. (2018) have applied artificial neural networks (ANN) in forecasting oil consumption with the help of Google search data. They show that neural networks in general have better performance than conventional econometric models, such as linear regression. This suggests that the application of ANN can increase forecasting performance in other subjects and datasets as well. However, to the best of our knowledge, there has never been executed a research that combines unemployment rates, Google trends and ANN.

[1]The categories are 'unemployment rate', 'unemployment office or agency', 'personnel consultant', and a category consisting of the largest job search engines in Germany.

3 Data

For data, we distinguish two parts. The first part of the data forms the target dataset. The second part forms the set of explanatory variables, in this particular case, the Google trends data. The data are split up in two parts, of which we use one part as estimation sample, and the other part is used for forecasting so that we can evaluate the performance empirically.

3.1 Unemployment rate

The dependent data used in this paper are monthly unemployment rates of the United States. The data are derived from the St. Louis Federal Reserve Bank database. The Federal Reserve defines the unemployment rate (UR) as the number of unemployed people as a percentage of the labor force. The labor force consists of all Americans aged 16 and older, who are living on American soil (overseas territories excluded) and who do not live in certain institutions (e.g. prison and home for the aged), and who are not on active duty in the US Armed Forces.
The unemployment rate data are available from 1948 up until today. However, since Google trends data are only available from the beginning of 2004, we choose the sample data to range from 01/2004 until 04/2018, as shown in the first graph in figure 1. When looking at the figure, we find the UR to range from 4 to 10 percent. The graph shows a wave-like pattern, which is due to the business cycles. Regarding the pattern as a wave, the sample data describe a full cycle. Besides, the strong increase between 2008 and 2010 is related to the financial recession. Recessions and periods of strongly increasing UR are clearly and empirically correlated. As for the most recent observations, the current UR is at its lowest point of the data sample, resulting from the strong upward business cycle that has started in 2010.
In total, the dataset consists of 171 observations. Similar to the data division by Yu et al. (2018), we split the data for estimation and forecasting purposes in a 80% : 20% proportion, approximately. This boils down to samples consisting of 137 and 34 observations, representing the periods of 01/2004 - 05/2015 and 06/2015 - 03/2018, respectively. In the remainder of this paper, the unemployment rate time series is defined as y_t, where $t = 1, ..., 171$.

3.2 Google trends

In order to assess the forecasting power of Google trends for unemployment rates, we retrieve many search engine query data. Based on Askitas and Zimmermann (2009), and based on Choi and Varian (2009), the trends s_t^n $(n = 1, ..., N)$ that are evaluated form part of different

Table 1: The used Google trends s_t^n per category.

(i)	(iii)	(iv)
Unemployment	Careerbuilder	Unemployment benefits
Unemployment rate	Dice	Unemployment compensation
	Glassdoor	Unemployment insurance
(ii)	Google careers	1st PC
Unemployment agency	Indeed.com	
Unemployment office	Job search engine	
	LinkedIn	
	Monster.com	
	1st PC	
	2nd PC	
	3rd PC	

categories. We construct four categories. (i) The terms 'Unemployment (rate)'. (ii) The terms 'Unemployment office/agency'. (iii) A category consisting of several large job search engines in the US. Next to that, we include the Google search term 'Job search engine'. Because many job-related websites have ambiguous names, we restrict the data to the category of Jobs & Education. Furthermore, for each website we choose the search query with or without the *.com* term depending on which query results in more searching activity. (iv) The terms 'Unemployment benefits/compensation/insurance'.

The first category is the most obvious search term for this subject. For the second category, we expect the trends to connect mostly to people having contact or relations with an unemployment office, such that it relates to some 'flow into unemployment'. The third category is expected to relate to people who are looking for a new job, such that it reflects some 'flow into employment'. The fourth category connects to people that are either unemployed or anxious to become so within the foreseeable future. For clarity, table 1 shows an overview of all used Google trends s_t^n per category. In graphs 2-8 in figure 1 all Google trends are shown.

3.2.1 Principal component analysis

At first sight, the Google trends show some similarities with each other within their categories. These similarities are clearly present in categories (i) and (iv). Also, category (iii) contains many Google trends, and this suggests that some Google trends can perhaps be represented by a smaller amount of composite trends. For this reason, we perform principal component analysis (PCA) on category (iii) and (iv). No PCA is performed on (i) and (ii), since those categories already consist of only two trends. We pick the first principal components (PC) in a way that at least 80% of the information of the Google trends of the category is accounted for. For category (iii), the PCA results in 3 important principal components, accounting for 85%

(of which 57% by the first PC). For category (iv), we find 1 PC, accounting for 81%. We add a total of 4 principal components to the Google trends s_t^n and include them in the relationship investigation. This leads to a total of $N = 19$ trends to be investigated. The principal components are also mentioned in table 1.

4 Methodology

We include Google trends and principal components in autoregressive models. We distinguish two main steps in the procedure of forecasting with Google trends. First, we perform relationship investigation. Second, we perform prediction improvement.

4.1 Relationship investigation

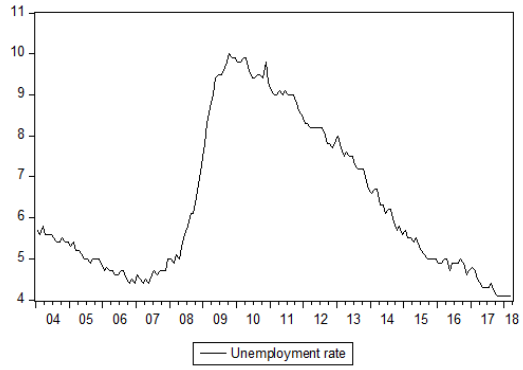
The goal here is to analyze whether the found Google trends s_t^n are related to and have effect on the unemployment rate y_t. First, a cointegration test is performed in order to assess statistically whether Google trends and unemployment rates influence each other. Cointegration holds if and only if there exists a linear combination of two time series x_t and z_t that has a lower order of integration than the series themselves. To test for this, a two-step Engle-Granger test is performed. First, the stationarity of both series is tested based on the augmented Dickey-Fuller (ADF) test. Then, for $n = 1, ..., 19$, we perform a simple linear regression: $y_t = \beta_0 + \beta_1 s_t^n + \epsilon_t$ and we test for stationarity of the residual term ϵ_t.

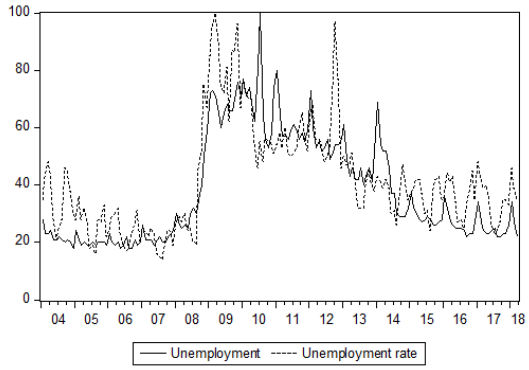
Now, a first round of filtering of the trends can be executed. We omit all trends having stationarity at a different level than the unemployment rate. This is done so because stationarity at the same level as y_t is a necessary condition for Granger causality. Next, for all remaining trends s_t^n we perform the Granger causality test to check for causal relations between Google trends and the unemployment rate. Here, we use an autoregressive (AR) model:

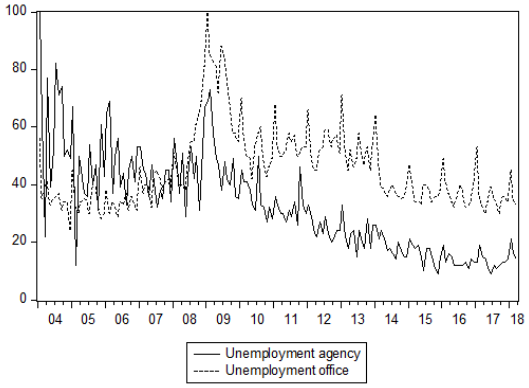
$$y_t = a_0 + a_1 y_{t-1} + ... + a_m y_{t-m} + b_1 s_{t-1}^n + ... + b_m s_{t-m}^n + u_t. \tag{1}$$

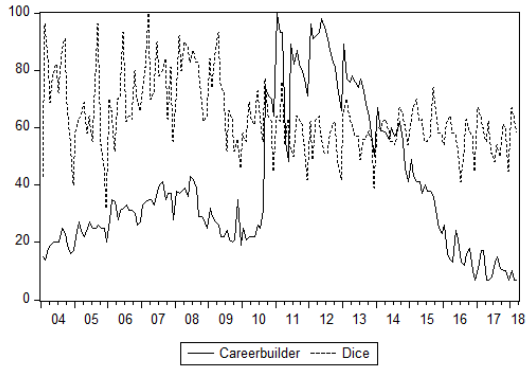
Here, for the residual terms it holds that $u_t \sim \texttt{iid}(0, \sigma^2)$. Furthermore, m autoregressive terms are included. The selection of m is based on the Akaike Information Criterion (AIC) and the Schwartz Information Criterion (SIC). Now, by means of an F-test the significance of the coefficients b_i is assessed. If the coefficients $b_1, ..., b_m$ are jointly significantly different from zero, we can assume Granger causality from s_t^n to y_t, i.e. there exists a Granger causal relationship of this particular Google trend on the unemployment rate. This means that the trend might be

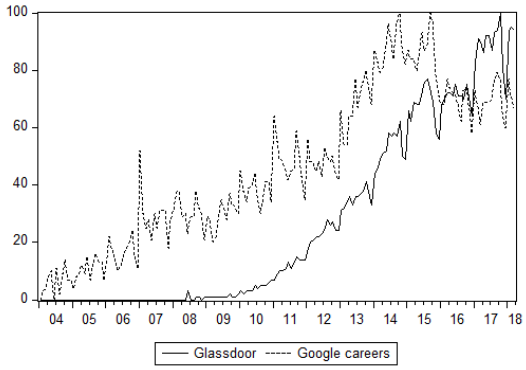
Figure 1: Graphs of the unemployment rate and the used Google trends. Graph 1: US unemployment rate (%). Graph 2-8: Google trends.

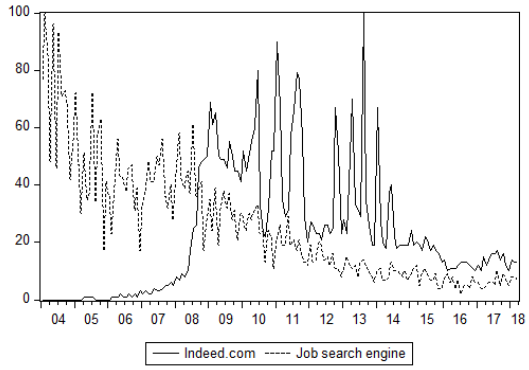


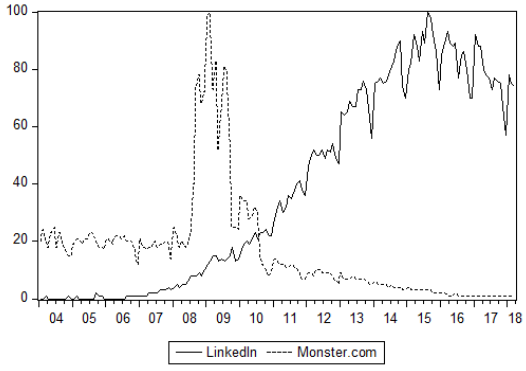


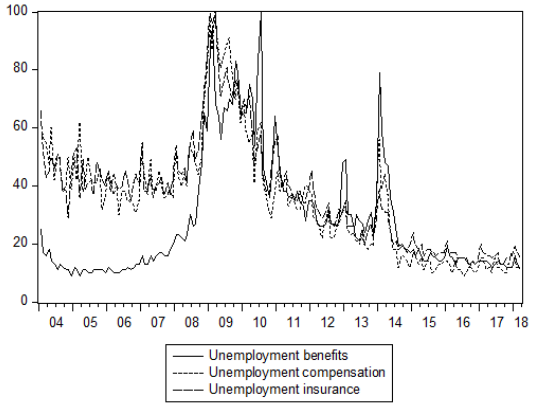












useful in forecasting the unemployment rate.

After the Granger causality test, we can make a final selection of trends. We pick the Google trends that display a Granger causal relationship with y_t, and therefore might influence the unemployment rate. The remaining set of Google trends and related principal components are defined as s_t^k, where $k = 1, ..., K$ and $K \leq N = 19$.

4.1.1 Handling of principal components

After having obtained the possibly useful trends s_t^k, there exists a possibility that both Google trends and principal components based on the same Google trends are still included. Since principal components are in fact linear combinations of their related Google trends, these trends can show a high degree of (multi)collinearity. Multicollinearity can cause severe problems with regressing and forecasting data. To account for this problem we cannot include both types of trends and a decision has to be made to omit either the Google trends or the corresponding principal component. As a solution, we select the trends with the largest potential influence on the unemployment rate, in terms of the Granger causality test statistic. We use the principal component if its test statistic is larger than the test statistic of each Google trend that it is based on. Otherwise, we use the Google trends and discard the PC.

4.2 Prediction improvement

In this section, we will perform forecasts and try to find improvements in the prediction quality using several methods. In general, we consider two types of models. The first type is the so-called benchmark one-step ahead prediction model. In this case, the benchmark model is defined as follows:

$$\hat{y}_{t+1} = f(Y_t) = f(y_t, y_{t-1}, ..., y_{t-m+1}), \tag{2}$$

where m is the lag order of autoregression. The other type of models also includes the Google trends s_t^k resulting from the cointegration and causality tests, which leads to the following general formula:

$$\hat{y}_{t+1} = f(Y_t, s_t^k) = f(y_t, y_{t-1}, ..., y_{t-m+1}, s_t^1, ..., s_t^K). \tag{3}$$

Now, for $f(\cdot)$, we define various different functions, among them econometric and AI methods. Furthermore, we distinguish two types of prediction approaches, namely both directional predictions and level predictions.

4.2.1 Directional predictions

For directional predictions, we will use two different techniques to forecast the unemployment rate, which are Logit (Huang, Yang & Chuang, 2008) and BPNN (Groth & Muntermann, 2011).

Logit Logistic regressions are one of the most common techniques used for binomial estimating and forecasting. It calculates the probability p of the occurrence of a particular event, in this case the probability of an upward directional change of the unemployment rate. One of multiple ways to interpret this model in general is the following. Since we are looking for a probability, the outcome should have a value between 0 and 1. Because a standard linear regression is not restricted to this range, a logit transformation is applied on the probabilities. This results in a linear model for the log-odds of the occurrence of the event, rather than for the probability, as shown in equation 4. Here, the values of β_i $(i = 1, ..., l)$ are the partial regression coefficients of the corresponding explanatory variables x_i $(i = 1, ..., l)$. We will call this function z.

$$z := ln\left(\frac{P(\hat{y} = 1)}{1 - P(\hat{y} = 1)}\right) = \beta_0 + \beta_1 x_1 + ... + \beta_l x_l \tag{4}$$

However, instead of the value of z, we are interested in the probability of an upward trend. For this reason, we rewrite the equation, yielding the following formula:

$$P(\hat{y} = 1) = \frac{\exp(z)}{1 + \exp(z)}. \tag{5}$$

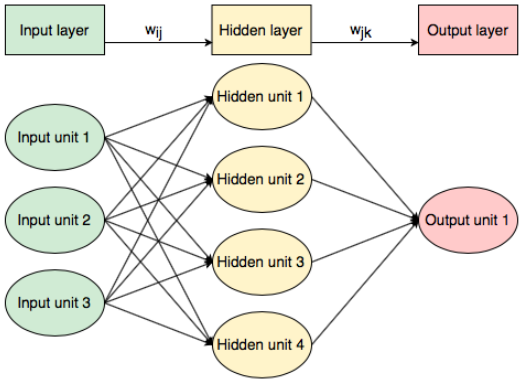
In this particular paper, we define z as a linear combination of a constant, all autoregressive terms and the relevant Google trends and PCs, as shown in equation 6:

$$z = \beta_0 + \beta_1 y_t + ... + \beta_m y_{t-m+1} + \beta_{m+1} s_t^1 + ... + \beta_{m+K} s_t^K. \tag{6}$$

For the benchmark model we restrict the values such that $\beta_i = 0$, for $i > m$. A general-to-specific approach is then performed to omit all insignificant variables from the model. Here, we work at the common 5% significance level. The value of $P(\hat{y} = 1)$ is compared to an arbitrary threshold, which in this case equals 0.5, and predicts the trend accordingly. This will result in the prediction of an upward trend $(\hat{y} = 1)$ if $P(\hat{y} = 1) \geq 0.5$ and in the prediction of a downward trend $(\hat{y} = 0)$ if $P(\hat{y} = 1) < 0.5$.

BPNN The second level prediction technique is based on back-propagation neural networks (BPNN), which is a specific case within the set of artificial neural networks (ANN). An ANN is a computing system with a structure based on the human brain. The idea behind an ANN is to

Figure 2: Example of an artificial neural network with one hidden layer, and $m + K = 3$, i.e. three input neurons, four hidden neurons, and one output neuron.



replicate or simulate a network of connected brain cells in an computer and to make it capable of learning and recognizing relationships and patterns.

In general, an ANN consists of different neurons, which can be considered as brain cells. The neurons are grouped in different layers, resulting in one input layer, one or more hidden layers, and one output layer. Input data walks through the network starting at the input layer, through the hidden layer(s) and produces some output in the output layer. Across layers, all units are connected by arcs so that a complex network takes shape. The arcs indicate to which neurons the output is given and they carry weights on them, which is of later use for the self-learning feature of the network.

In this paper, we make use of a BPNN. One of the main advantages of a BPNN compared to other ANNs, is the efficiency and relatively straight-forward algorithm of the training routine. The process of back-propagation means that the network improvement starts by comparing the output it produces to the true value and then calculates the difference or error term. Then, by back-propagating through the network, it updates the weights of the arcs such that the next output is closer to the target value. By repeatedly comparing output values to target values and constantly decreasing the difference between them, the network is learning to give the correct output for some given input.

Walking through the network in more detail, it begins with the network structure. The input layer consists of the exact number of neurons equaling the amount of explanatory variables. In this paper, this number equals $m + K$, the number of autoregressive lags plus the number of Google trends. The hidden layer consists of the number of neurons in the input layer plus one, equaling $m + K + 1$. The output layer consists of only one neuron. An example of a small network structure (with $m+K = 3$) can be seen in figure 2. All weights on the arcs between the

different layers' neurons are initialized with a random value, drawn from a uniform distribution on the $(0, 1)$ interval. Now, some input is given to the network. For each neuron feeding forward to all of its following neurons we use an activation function, the so-called sigmoid function. The sigmoid function is an S-shaped function with output ranging from 0 to 1, and is defined as $s(x) = 1/(1 + \exp(-x))$. One of the main advantages of this function is the simplicity of its derivative, defined as $s'(x) = s(x) \cdot (1 - s(x))$. For any neuron j in the hidden layer, this results in the following formula:

$$h_j = \frac{1}{1 + \exp\{-(\sum_i^{m+K} w_{ij} x_{t,i} + \theta_h)\}}, \tag{7}$$

where w_{ij} represents the weight from unit i in the input layer to neuron j. $x_{t,i}$ represents the normalized value of the ith explanatory variable for predicting y_t and is used as input value in unit i, and θ_h represents the so-called 'bias', or threshold for activation, corresponding to the hidden layer. For the output neuron, the activation function is defined as follows:

$$o_k = \frac{1}{1 + \exp\{-(\sum_j^{m+K+1} w_{jk} h_j + \theta_k)\}}, \tag{8}$$

where w_{jk} represents the weight between neuron j in the hidden layer and output neuron k, and θ_k represents the bias of the output layer.

After the calculation of o_k, the result is compared to the true value and a squared error term, E, is calculated. Then, in order to reduce this error term, the weights are adjusted until optimal with the use of back-propagation and gradient descent. Starting with the weights between the hidden and output layers, followed by the weights between the input and hidden layers, in each step, the weights are updated slightly:

$$w_{jk}^* = w_{jk} - \eta \cdot \frac{\partial E}{\partial w_{jk}}, \tag{9}$$

where w_{jk}^* represents the updated weight, η represents an arbitrary learning rate, and the last term represents the partial derivative of the error term to the weight, which can be interpreted as the influence of a change of the value of the specific weight to E. The updates for w_{ij}^* are executed in a similar way.

The back-propagation procedure repeatedly goes on until the satisfaction of the stop criterion. In this case the iteration stops as soon as the average improvement of E over the last 100 iterations is lower than some small (of the order of $1 \cdot 10^{-6}$) and arbitrary value, ϵ. In this manner we still allow for small increases in E. This is necessary since E sometimes inevitably rises from a local minimum in order to obtain a larger fall. Now, in order to predict the trend

direction in out-of-sample data, we compare the output o_k to a threshold value of 0.5. An output value below the threshold results in the prediction of a downward trend ($\hat{y} = 0$), and an output above the threshold represents the prediction of an upward trend ($\hat{y} = 1$).

4.2.2 Level predictions

For level predictions, we will use three different approaches for forecasting the unemployment rate. These are OLS (Brey, Jarre-Teichmann & Borlich, 1996), BPNN (Yu, Zhao & Tang, 2014) and ELM (Huang, Zhu & Siew, 2006).

OLS Ordinary least squares (OLS) is one of the simplest and most common methods in the econometric world. It consists of many different categories and subtypes. For the benchmark function we will perform OLS based on the following formula:

$$y_t = \beta_0 + \beta_1 y_{t-1} + ... + \beta_m y_{t-m} + \varepsilon. \tag{10}$$

Following a general-to-specific approach, all insignificant variables are omitted from the model. Then, based on the variables and their estimated corresponding coefficients, one-step ahead forecasts can be constructed according to the following equation:

$$\hat{y}_{t+1} = \hat{\beta}_0 + \hat{\beta}_1 y_t + ... + \hat{\beta}_m y_{t-m+1}. \tag{11}$$

Similarly, for the case with Google trends, the K different Google variables s_t^k are added to the model and included in the OLS estimation. Again, a general-to-specific approach ensures that all coefficients of the variables in the model are significantly different from zero. Now the directional one-step ahead forecast can be constructed based on the extended version of equation 11:

$$\hat{y}_{t+1} = \hat{\beta}_0 + \hat{\beta}_1 y_t + ... + \hat{\beta}_m y_{t-m+1} + \hat{\beta}_{m+1} s_t^1 + ... + \hat{\beta}_{m+K} s_t^K. \tag{12}$$

BPNN Again, we use the BPNN method to make predictions. BPNN is suitable for both directional and level predictions. In the level prediction case, the estimated output is represented by $o_k \geq 0$, as defined before.

However, in order to facilitate this type of predictions, the output data has to be normalized first. This is the consequence of the fact that the underlying sigmoid activation function of o_k only ranges from 0 to 1 and the original data do not. The normalization of the data is performed by means of a so-called 'min-max normalization'. This method is a linear function

and reshapes all data to fit within the range from 0 to 1. The normalized value of y_t is defined as follows: $y_t^* = (y_t - min)/(max - min)$, where min and max represent the minimum and maximum value of the observations Y, respectively.

ELM Although the structure is very similar to that of a BPNN, an extreme learning machine (ELM) is a special case of a single-hidden layer artificial neural network. We make use of the same notation that has been introduced for the BPNN method. In the case of ELM, the weights w_{ij} on the arcs between the input and hidden layers are set randomly, without any further fine-tuning of their values. This feature has the advantage of saving time in comparison with the earlier mentioned BPNN. These savings in time result from the fact that the back-propagation only iterates over one set of weights - the weights w_{jk} between the hidden and output layers - instead of both sets of weights.

4.2.3 Cross-validation

A major risk of training neural networks is a consequence of the relatively large amount of variables and parameters, such as the weights and the number of layers, that are involved in the model. This could theoretically lead to an over-fitting situation in the estimation of the data. This means that the model might be adjustable in a way that every observation in the training sample can be predicted almost perfectly. Although a good fit of the training data is desirable, this does not directly imply a comparable performance on the testing sample, since the network is not trained for these observations. If this is the case, it would mean that the model is not generalizable to data outside of the training sample.

In order to test for the presence of this characteristic, we perform a 6-fold cross-validation. This cross-validation involves splitting up the training data randomly in six equally sized parts. For each run, one of six parts is omitted from the model, after which the network is trained for the five remaining parts. After completion, based on the estimation, we evaluate the prediction performance of the sixth, omitted data part, which is then used as a temporary testing sample. We compare the prediction performance in terms of the squared error term E of the six different runs. We check whether the six performances are comparable to each other, which is an indication of generalizability. We apply cross-validation on the ELM method including Google trends, for time-saving considerations.

4.3 Evaluating results

After the relationship investigation and predictions have been executed, the results are investigated. In order to assess possible improvements in comparison with the benchmark model, the results and statistics are then compared to each other. We make a distinction between the directional and level prediction results.

4.3.1 Directional prediction results

For the assessment of the directional predictions, we use one criterion: PCC (Edwards, Cutler, Zimmermann, Geiser & Moisen, 2006). We calculate the value of the percentage correctly classified (PCC), which is defined as follows:

$$PCC = \frac{1}{M}\sum_{t=1}^{M} I_t, \tag{13}$$

where I_t is an indicator function taking the value 1 if $\hat{y}_t = y_t$ and the value 0 otherwise. The PCC can range from 0 to 1. A PCC equaling 1 would indicate that the predictor is perfectly forecasting the directions.

Next, we construct a variable to calculate the relative improvement in error terms that follows from adding Google trends, compared to the benchmark models. For the PCC, the improvement rate (IR) is defined as

$$IR_{\text{PCC}} = \frac{\text{PCC}_G - \text{PCC}_B}{\text{PCC}_B} \cdot 100\%. \tag{14}$$

In the formula, the PCC of the benchmark model B is compared to that of the extended Google trend model G.

4.3.2 Level prediction results

For the assessment of the level predictions, we use two common criteria: the root mean squared error (RMSE) and the mean absolute percentage error (MAPE) (Wang, Yu, Tang & Wang, 2011). The criteria are defined as follows:

$$RMSE = \sqrt{\frac{1}{M}\sum_{t=1}^{M}(\hat{y}_t - y_t)^2}, \tag{15}$$

$$MAPE = \frac{1}{M}\sum_{t=1}^{M}\left|\frac{\hat{y}_t - y_t}{y_t}\right|, \tag{16}$$

with M being the sample size. Again, we construct the improvement rate. The IR for RMSE and MAPE are defined in a comparable way as the IR for PCC in equation 14, except for the fact that we multiply by -1 here. This leads to the following formula for RMSE, and similarly for MAPE:

$$IR_{\text{RMSE}} = -\frac{\text{RMSE}_G - \text{RMSE}_B}{\text{RMSE}_B} \cdot 100\%. \tag{17}$$

5 Results

For clarity purposes, the analyses of the results are split up in two parts. First, the results with respect to the relationship investigation will be evaluated in 5.1. Second, the results from the unemployment rate predictions will be investigated in 5.2.

5.1 Relationship investigation

Table 2: Results of the stationarity test and the cointegration tests, both in terms of t-statistics (p-values). Bold and underlined results are significant at the 5% level.

		Panel A		**Panel B**
		Stationarity test		Cointegration test
	Time series	Original level	1st-order difference	1st-order difference
	Unemployment rate	-0.5089 (0.4943)	-2.1523 (**0.0307**)	
(i)	Unemployment	-0.4938 (0.5005)	-4.3649 (**0.0000**)	-10.0303 (**0.0000**)
	Unemployment rate	-2.6143 (0.0925)	-12.0559 (**0.0000**)	-9.6102 (**0.0000**)
(ii)	Unemployment agency	-1.8281 (0.3656)	-17.0283 (**0.0000**)	-15.6606 (**0.0000**)
	Unemployment office	-0.4502 (0.5180)	-3.3303 (**0.0010**)	-2.6541 (**0.0082**)
(iii)	Careerbuilder	-2.2923 (0.1760)	-13.7496 (**0.0000**)	-2.8984 (**0.0040**)
	Dice	-7.0223 (**0.0000**)	-	-
	Glassdoor	1.7595 (0.9997)	-0.2938 (0.9214)	-
	Google careers	-0.2676 (0.9254)	-11.3060 (**0.0000**)	-10.2066 (**0.0000**)
	Indeed.com	-0.7314 (0.3979)	-7.4364 (**0.0000**)	-7.5247 (**0.0000**)
	Job search engine	-3.1009 (**0.0288**)	-	-
	LinkedIn	1.8157 (0.9998)	-1.9778 (0.2963)	-
	Monster.com	-1.5581 (0.1117)	-10.7378 (**0.0000**)	-10.0593 (**0.0000**)
	1st PC	-0.5094 (0.8847)	-13.2156 (**0.0000**)	-2.3598 (**0.0182**)
	2nd PC	-2.9595 (**0.0414**)	-	-
	3rd PC	0.5195 (0.8267)	-10.2193 (**0.0000**)	-13.1894 (**0.0000**)
(iv)	Unemployment benefits	-2.5571 (0.1045)	-11.2426 (**0.0000**)	-11.2807 (**0.0000**)
	Unemployment compensation	-1.3111 (0.1748)	-15.4985 (**0.0000**)	-13.8131 (**0.0000**)
	Unemployment insurance	-1.4168 (0.1453)	-13.0893 (**0.0000**)	-11.8726 (**0.0000**)
	1st PC	-1.3008 (0.1779)	-10.1265 (**0.0000**)	-12.3594 (**0.0000**)
	2nd PC	-1.9519 (**0.0490**)	-	-

The research begins with investigating whether the Google trends s_t^n show any relationship with the unemployment rate. The first test that is conducted is a test for stationarity by

means of the ADF test, with an intercept included. The corresponding results are shown in Panel A in table 2. The results show that the US unemployment rate is stationary at the first-difference level. The same holds for the Google trends of category (i) and (ii). For category (iii), two Google trends and one principal component are stationary at the original level. The Google trends for 'Glassdoor' and 'LinkedIn' are not stationary at the original level nor the first-difference level. For category (iv), all Google trends and the first principal component are stationary at the first-difference level. Being a necessary condition for cointegration and Granger-causality, the unemployment rate and the Google trends must be stationary at the same level. In other words, all trends that are not stationary at the first-difference level, are omitted from the dataset.

The next step is testing for cointegration. This test is performed at the first-difference level because of the US unemployment rate results from the stationarity test. The results of the cointegration test are reported in Panel B in table 2. Here, it becomes clear that all Google trends that have passed the stationarity test successfully also show significant cointegration relationships with the US unemployment rate at the 5% level. Therefore, no additional variables are omitted from the model in this step.

The last step of the relationship investigation involves testing for Granger causality. Using the test we can assess statistically whether the Google trends might be able to predict the unemployment rate. First, for equation 1 the correct value of m is selected based on the AIC and SIC. Both criteria show an optimal number of lags of $m = 6$. For this reason, each time the test is performed over lags 1 to 6. A concise overview of the relevant results can be found in table 3, and a complete overview of all results can be found in the Appendix. We observe that in each category one Google trend shows Granger causality over all six lags at the 5% significance level. In category (iv), also the first principal component appears to Granger cause the unemployment rate. We omit all Google trends and principal components that do not meet the requirement of Granger causality over all six lags.

Based on the relationship investigation, we select the Google trends s_t^k that might be able to help predicting the unemployment rate. Thus, more specifically, the selection consists of the Google trends of 'Unemployment', 'Unemployment office', 'Monster.com', 'Unemployment benefits', and the first principal component of the fourth category. However, we observe that the latter two trends both are part of the fourth category. Due to imminent collinearity, we wish to omit one of both from the model. As mentioned earlier, we select the trend having the strongest Granger causal relationship with the unemployment rate, in terms of the Granger

Table 3: Results of the Granger-causality analysis with 6 lags, in terms of F-statistics (p-values), under the null hypothesis that the Google trend does not Granger-cause the unemployment rate. Bold and underlined results are significant at the 5% level. Only the significant Google trends are shown; for the complete results, see Appendix.

	Time series	F-stat	(p-value)
(i)	Unemployment	2.9525	(**0.0101**)
(ii)	Unemployment office	2.6995	(**0.0172**)
(iii)	Monster.com	4.2341	(**0.0007**)
(iv)	Unemployment benefits	3.8270	(**0.0016**)
	1st PC	2.7643	(**0.0150**)

causality test statistic. The p-values resulting from the F-test over 6 lags of 'Unemployment benefits' and the first PC are 0.0016 and 0.0150, respectively. For this reason we discard the first principal component of the fourth category. This leads to a remainder of four Google trends, all originating from a different category.

5.2 Prediction results

After the useful Google trends have been selected, we can make predictions with and without these trends included, and evaluate the results. We distinguish two types of predictions, namely the directional and the level predictions.

5.2.1 Directional predictions

Table 4: Comparison of the results of different directional forecasting methods in terms of PCC.

	Logit	BPNN
Without Google trends	73.53%	64.71%
With Google trends	76.47%	73.53%
IR	4.00%	13.63%

We have used two different methods for directional forecasting, each with and without Google trends. We observe the PCC values and observe an increase after including the Google trends. For the Logit model, the value of the PCC increases by four percentage points. Now, for the directional forecasts based on the BPNN method, we observe a larger improvement than for the Logit model, namely around 14% percentage points.

Next to comparing the performances of the models with and without Google trends per method, we can evaluate possible differences existing between the methods. Regarding the models without Google trends, we observe that the Logit model has better performance than BPNN, with its PCC value being 9 percentage points higher. Between the models with Google trends, the Logit model now only performs three percentage points better, in terms of the PCC. However,

it must be noted that, after inclusion of the Google trends, the performance of the BPNN still only is exactly as good as the Logit model without Google trends.

5.2.2 Level predictions

Table 5: Comparison of the results of different level forecasting methods in terms of MAPE.

	OLS	BPNN	ELM
Without Google trends	2.02%	3.81%	10.86%
With Google trends	1.98%	2.86%	6.49%
IR	1.98%	34.97%	40.24%

Table 6: Comparison of the results of different level forecasting methods in terms of RMSE.

	OLS	BPNN	ELM
Without Google trends	0.1344	0.2100	0.5848
With Google trends	0.1147	0.1602	0.3769
IR	14.66%	23.71%	35.55%

We have performed one-step ahead forecasts based on three different methods, and for each method we have used both the model without and the model with Google trends. We start the analysis of the level prediction performance by evaluating the value of the MAPE. Regarding the improvement rate, all three models show improvement after the inclusion of Google trends. The IR of the OLS method (2%) is significantly smaller than those of BPNN and ELM (35% and 40%, respectively). On the other hand, the value of the MAPE of OLS is already smaller than the other two models, both without and with Google trends. The ELM can be considered as a simplification of the BPNN method. In comparison with BPNN, the performance of ELM has worsened. This means that in predicting the unemployment rate, compared to BPNN, the advantage of faster estimation with the ELM method comes at the cost of worse prediction performance.

Regarding the values of the RMSE for all methods, again we observe that the OLS method has the best performance for both models, followed by BPNN, and ELM performs the worst. For all three methods it holds that the improvement rates are considerably large.

As an overall indication of the performance, we note that the average values of the MAPE and RMSE of all three methods are smaller (3.78% and 0.2173, respectively) for the models with Google trends than for the models without the trends (5.56% and 0.3097, respectively). Once again, this confirms the additional predictive power as provided by the inclusion of relevant Google trends.

The 6-fold cross-validation, applied to ELM, results in 6 squared error terms: 0.090, 0.082,

0.085, 0.087, 0.074, 0.082. These values are close to each other and all fall within the 95%-confidence interval of (0.073, 0.093). This is a fairly reasonable argument to assume that the data are generalizable to unknown data outside of the training set.

6 Conclusion

To summarize, the most important conclusions that can be drawn from the results, are the following. First of all, different relationship tests, such as the cointegration and Granger causality tests, show that there exist several Google trends that have the potential of being effective predictors of the American unemployment rate. The useful Google trends are all related to different categories and are all representing query data of a different nature, indicating that a wide range of Google trends might be capable of improving unemployment rate predictions.
Second, for the prediction of directions, the addition of Google trends slightly improves the forecasting performance in terms of the PCC value. For the Logit method the improvement equals four percentage points, and for the BPNN the improvement equals roughly 14 percentage points. Next to that, we observe that the Logit method, which is technically less sophisticated and expensive, outperforms the BPNN method.
Third, for the prediction of levels, for all methods the addition of Google trends improves the forecasting performance, in terms of both the MAPE and the RMSE values. Again, OLS, technically the least sophisticated method, outperforms the methods based on artificial neural networks. In terms of the IR, the addition of Google trends is most useful for neural networks with improvement rates well over 20%, whereas for the OLS method Google trends are less effective in increasing the forecasting accuracy.
Returning to the research question as stated in the introduction, we answer the question whether, and to what extent, Google trends can attribute to performance improvements in predicting the US unemployment rate. We have shown empirically that Google trends can indeed add predictive power to both level and directional prediction methods. The Google trends especially contribute to better predictions for those methods with relatively poor performance in the benchmark models, which are the methods based on artificial neural networks.

7 Discussion

Although the results show some promising results with respect to forecasting improvements for econometric and AI methods using Google trends, the research has several limitations. De-

spite the fact that the selection of potentially useful Google trends was performed carefully and based on earlier literature, it nevertheless remains an arbitrary process. The selection of Google trends leaves room for improvement. More categories could be set up and the already existing categories can be expanded.

Second, there are difficulties with respect to the replicability of the research. Google trends are automatically standardized for any requested dataset. This results in the fact that Google trends for any search term can be different, even for the same location and keyword. Another limitation related to the standardization of the Google trends, is the fact that the results are rounded off to their nearest integers, resulting in a loss of accuracy.

For further research, we suggest a more thorough exploration of the possibilities of artificial neural networks. The effects of adding hidden layers, adjusting the learning rate and using different activation functions could be looked after, as well as making use of other gradient descent methods such as the Quasi-Newton method.

References

[1] Artola, C., Pinto, F., & García, P. D. (2015). Can internet searches forecast tourism inflows? *International Journal of Manpower*, 36(1), 103-116. doi:10.1108/ijm-12-2014-0259

[2] Askitas, N., & Zimmermann, K. F. (2009). Google Econometrics and Unemployment Forecasting. *SSRN Electronic Journal*. doi:10.2139/ssrn.1480251

[3] Brey, T., Jarre-Teichmann, A., & Borlich, O. (1996). Artificial neural network versus multiple linear regression:predicting P/B ratios from empirical data. *Marine Ecology Progress Series*, 140, 251-256. doi:10.3354/meps140251

[4] Carrière-Swallow, Y., & Labbé, F. (2011). Nowcasting with Google Trends in an Emerging Market. *Journal of Forecasting*, 32(4), 289-298. doi:10.1002/for.1252

[5] Choi, H., & Varian H. (2009). Predicting initial claims for unemployment benefits. Available at: https://static.googleusercontent.com/ media/research.google.com/nl//archive/ papers/initialclaimsUS.pdf (accessed May 14 2018).

[6] Constant, A., & Zimmermann, K. F. (2008): Im Angesicht der Krise: US-Prsidentschaftswahlen in transnationaler Sicht. *DIW Wochenbericht*, 44, 688 - 701.

[7] Edwards, T. C., Cutler, D. R., Zimmermann, N. E., Geiser, L., & Moisen, G. G. (2006). Effects of sample survey design on the accuracy of classification tree models in species distribution models. *Ecological Modelling*, 199(2), 132-141. doi:10.1016/j.ecolmodel.2006.05.016

[8] Engle, R.F., & Granger, C. W. J. (1987): Co-integration and error correction: representation, estimation, and testing. *Econometrica*, 55, 251-276. doi: 10.1017/ccol052179207x.009

[9] Federal Reserve Bank of St. Louis. *Civilian Unemployment Rate*. Available at: https://fred.stlouisfed.org/series/UNRATE (accessed May 15 2018).

[10] Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, 457(7232), 1012-1014. doi:10.1038/nature07634

[11] Groth, S. S., & Muntermann, J. (2011). An intraday market risk management approach based on textual analysis. *Decision Support Systems*, 50(4), 680-691. doi:10.1016/j.dss.2010.08.019

[12] Huang, C., Yang, D., & Chuang, Y. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications*, 34(4), 2870-2878. doi:10.1016/j.eswa.2007.05.035

[13] Huang, G., Zhu, Q., & Siew, C. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3), 489-501. doi:10.1016/j.neucom.2005.12.126

[14] Preis, T., Moat, H. S., & Stanley, H. E. (2013). Quantifying Trading Behavior in Financial Markets Using Google Trends. *Scientific Reports*, 3(1). doi:10.1038/srep01684

[15] Wang, S., Yu, L., Tang, L., & Wang, S. (2011). A novel seasonal decomposition based least squares support vector regression ensemble learning approach for hydropower consumption forecasting in China. *Energy*, 36(11), 6542-6554. doi:10.1016/j.energy.2011.09.010

[16] Yu, L., Zhao, Y., Tang, L., & Yang, Z. (2018). Online big data-driven oil consumption forecasting with Google trends. *International Journal of Forecasting*. doi: 10.1016/j.ijforecast.2017.11.005

[17] Yu, L., Zhao, Y., & Tang, L. (2014). A compressed sensing based AI learning paradigm for crude oil price forecasting. *Energy Economics*, 46, 236-245. doi:10.1016/j.eneco.2014.09.019

APPENDIX

Results of the Granger-causality analysis, in terms of F-statistics (p-values), under the null hypothesis that the Google trend does not Granger-cause the unemployment rate. Bold and underlined results are significant at the 5% level.

	Time series	1 lag	2 lags	3 lags	4 lags	5 lags	6 lags
(i)	Unemployment	9.2645 (**0.0028**)	4.3631 (**0.0147**)	3.2642 (**0.0236**)	2.6150 (**0.0384**)	3.5021 (**0.0054**)	2.9525 (**0.0101**)
	Unemployment rate	21.7652 (**0.0000**)	7.6608 (**0.0007**)	3.3598 (**0.0209**)	2.8009 (**0.0288**)	2.1473 (0.0644)	1.6215 (0.1471)
(ii)	Unemployment agency	16.0192 (**0.0001**)	7.0415 (**0.0012**)	2.4963 (0.0628)	1.0842 (0.3673)	0.8972 (0.4854)	0.5955 (0.7334)
	Unemployment office	48.4532 (**0.0000**)	21.4072 (**0.0000**)	9.6444 (**0.0000**)	5.8412 (**0.0002**)	4.2809 (**0.0013**)	2.6995 (**0.0172**)
(iii)	Careerbuilder	8.9821 (**0.0033**)	2.3066 (0.1037)	1.0040 (0.3934)	0.9284 (0.4498)	0.9021 (0.4821)	0.9078 (0.4919)
	Google careers	7.0904 (**0.0087**)	2.1269 (0.1233)	1.9604 (0.1233)	1.2473 (0.2945)	2.1377 (0.0655)	2.0127 (0.0693)
	Indeed.com	8.8541 (**0.0035**)	4.5369 (**0.0125**)	4.8504 (**0.0031**)	3.7862 (**0.0061**)	3.7131 (**0.0037**)	1.9769 (0.0743)
	Monster.com	75.3664 (**0.0000**)	28.2977 (**0.0000**)	12.3198 (**0.0000**)	7.6557 (**0.0000**)	7.2772 (**0.0000**)	4.2341 (**0.0007**)
	1st PC	13.7948 (**0.0003**)	5.0231 (**0.0079**)	1.4593 (0.2288)	0.8429 (0.5006)	1.7043 (0.1387)	1.4170 (0.2137)
	3rd PC	2.1165 (0.1481)	2.1537 (0.1202)	2.1122 (0.1019)	0.8457 (0.4988)	0.9607 (0.4448)	0.9309 (0.4756)
(iv)	Unemployment benefits	30.2239 (**0.0000**)	10.0040 (**0.0001**)	4.3810 (**0.0057**)	4.6148 (**0.0017**)	5.1902 (**0.0002**)	3.8270 (**0.0016**)
	Unemployment compensation	36.9522 (**0.0000**)	12.3384 (**0.0000**)	6.0537 (**0.0007**)	3.6079 (**0.0081**)	2.6964 (**0.0240**)	1.9055 (0.0855)
	Unemployment insurance	40.0771 (**0.0000**)	13.5389 (**0.0000**)	4.5756 (**0.0045**)	2.3071 (0.0618)	1.5464 (0.1805)	1.3415 (0.2442)
	1st PC	45.8629 (**0.0000**)	15.8430 (**0.0000**)	6.2595 (**0.0005**)	4.2161 (**0.0031**)	3.8737 (**0.0027**)	2.7643 (**0.0150**)