Indian Institute of Management Calcutta
Working Paper Series
WPS No 830 /July, 2019

Deciphering Indian Inflationary Expectations through Text Mining: An Exploratory Approach

Ashok Banerjee*
Professor, Finance & Control Group, IIM Calcutta
e-mail: ashok@iimcal.ac.in

Ayush Kanodia
PhD Student, Stanford Computer Science
e-mail: kanodiaayush@cs.stanford.edu

Partha Ray
Professor, Economics Group, IIM Calcutta
e-mail: pray@iimcal.ac.in
(* Corresponding Author)

Indian Institute of Management Calcutta
Joka, D.H. Road
Kolkata 700104
URL: http://facultylive.iimcal.ac.in/workingpapers

Deciphering Indian Inflationary Expectations through Text Mining: An Exploratory Approach

Ashok Banerjee[1], Ayush Kanodia[2], Partha Ray[3]

Inflationary expectations tend to play a crucial role in macroeconomic and financial decision / policy making. In particular, in an inflation targeting framework it is of paramount importance. While traditionally, model-based and survey-based inflation expectations are being used, in recent times, a literature has emerged to forecast various macro-aggregates using text-based sentiment estimates. Taking a cue from this approach, in this paper we attempt to decipher inflationary sentiment using text mining from two leading financial dallies, viz., the Economic Times and Business Line. In our algorithm we aggregate CPI basket level (viz., food, fuel, cloth & miscellaneous) sentiment into an overall index of inflationary expectation, adapting techniques from natural language processing (NLP). Our results from this text based model indicate significant success in tracking actual inflation.

Key words: Inflation Expectations, India, Machine Learning, Natural Language Processing, Text Mining

Jel Classification: C82, E37, E52.

[1] Professor, Finance & Control Group, IIM Calcutta; e-mail: ashok@iimcal.ac.in (Corresponding Author)
[2] PhD Student, Stanford Computer Science; e-mail: kanodiaayush@cs.stanford.edu
[3] Professor, Economics Group, IIM Calcutta; e-mail: pray@iimcal.ac.in
We acknowledge the data support provided by the Financial Research and Trading Laboratory of IIM Calcutta and research grant by IIM Calcutta

Deciphering Indian Inflationary Expectations through Text Mining: An Exploratory Approach

1. Introduction

Inflationary expectations tend to play a crucial role in macroeconomic and financial decision / policy making. In particular, it is of paramount importance when monetary policy is conducted within an inflation targeting framework or when the financial market player is thinking of her return from the bond / forex markets. But a perennial question in this context is: how does one measure inflationary expectations? Three broad strands are identified in the literature. *First*, model based forecasts (univariate or multivariate variety) are often taken recourse to. *Second*, inflationary expectations are also derived from class / group-specific inflationary expectations surveys routinely conducted by central banks / financial data providers. *Third*, inflationary expectations / perceptions are also inferred from the market yields of inflation-indexed bonds.

Each of these methods is useful, each has its limitations as well. Illustratively, there can be biases in the survey based expectations or model-based expectations could perform poorly in terms of out-of-smaple forecasts. Moreover, in Indian context, RBI's inflationary expectations has come under a scanner and it has been pointed out that as per the Central Bank's forecasts, inflation has often been overestimated (Economic Survey, 2016-17).

It is in this context that in this paper we propose and adopt a novel method of inferring inflationary expectations using a machine learning algorithm by sourcing economic news from the leading financial dailies of India. In particular, we argue that Economics / Finance can leverage advances in artificial intelligence (AI), natural language processing (NLP) and big data processing to gain valuable insights into the potential fluctuations of key macro indicators and attempt to predict the direction (upward *versus* downward) of monthly consumer price inflation.

The remainder of this article is organized as follows. While section 2 discusses the motivation of this approach, the methodology is delved in section 3. Section 4 presents the results and section 5 concludes.

2. Motivation and Received Literature

The motivation of this approach can be traced in two distinct strands of literature. First, among the monetary economists there is a large literature of what has come to be known as the "narrative approach to monetary policy". While the origin of this approach can perhaps be attributed to Friedman & Schwartz (1972)'s *Monetary History of United States*, Boschen and Mills (1995) derived an index of monetary policy tightness and studied the relation between narrative-based indicators of monetary policy and money market indicators of monetary policy. They found, "Changes in monetary policy, as

measured by the narrative-based policy indices, are associated with persistent changes in the levels of M2 and the monetary base". More recently, Romer and Romer (2004) derived a measure of monetary policy shocks for the US. Instead of taking any particular policy as an indicator of monetary policy shock, Romer and Romer (2004) derived a series based on intended funds rate changes around meetings of the Federal Open Market Committee (FOMC) for the period 1969–1996 by combining the "information on the Federal Reserve's expected funds rate derived from the Weekly Report of the Manager of Open Market Operations with detailed readings of the Federal Reserve's narrative accounts of each FOMC meeting". But all these papers involve some degree of subjectivity of reading the policy narratives. Hence a key question remains: how does one get rid of this subjectivity? It is here that more contemporary tools of machine learning and natural language processing become helpful.

This second strand of literature comes from machine learning. To get a perspective of its emergence, it is important to note that there has been a healthy scepticism and conscious efforts on the part of the academics to avoid forecasting economic / financial variables. Smith (2018) in a recent article attacked the profession and went on to say:

> "Academic economists will give varying explanations for why they don't pay much attention to forecasting, but the core reason is that it's very, very hard to do. Unlike weather, where Doppler radar and other technology gathers fine-grained details on air currents, humidity and temperature, macroeconomics is traditionally limited to a few noisy variables.... collected only at low frequencies and whose very definitions rely on a number of questionable assumptions. And unlike weather, where models are underpinned by laws of physics good enough to land astronauts on the moon, macroeconomics has only a patchy, poor understanding of individual human behavior. Even the most modern macro models, supposedly based on the actions of individual actors, typically forecast the economy *no better* than ultra-simple models with only one equation ....whatever the reason, the field of macroeconomic forecasting is now exclusively the domain of central bankers, government workers and private-sector economists and consultants. But academics should try to get back in the game, because a powerful new tool is available that might be a game-changer. *That tool is machine learning*" (emphasis added).

But what is machine learning? Loosely speaking, "Machine learning refers to a collection of algorithmic methods that focus on predicting things as accurately as possible instead of modelling them precisely" (Smith, 2018). With rapid advances in storing and analysing large amounts of unstructured data, there is increasing awareness that these data could be a rich source of useful information for assessing economic trends. Various attempts have emanated in forecasting macroeconomic and financial variables. Illustratively, Nyman and others (2016) used the Thomson-Reuters News archive (consisting of over 17 million English news articles) to assess macroeconomic trends in the UK. More recently, using machine learning techniques, Shapiro & others (2018)

developed new time series measures of economic sentiment based on computational text analysis of economic and financial newspaper articles from January 1980 to April 2015. Similarly, using macro news sentiment scores provided by a professional database agency, Brandt and Gao (2019) find news related to macro fundamentals have an impact on the oil price in the short run and significantly predict oil returns in the long run. Thus, there is now a burgeoning literature on this issue, thanks to the Rational Expectations thinking that policy making needs to have some sense of the future sentiment / expectations. However, the policy maker needs to take care of the popular adage of Goodhart's law whereby "when a measure becomes the target, it can no longer be used as the measure", so that forecasts fail when used for policy prescription and when used as targets naively (DeLong, 2002)

Detection of sentiment from the newspaper seems to less prone to this syndrome. Much of this literature asks the machine to find out the recurrence of some key words with appropriate identifiers in the newspaper articles so that the detection does not get corrupted by any subjective bias. Our paper tries to decipher systematically inflationary sentiment from newspaper articles.

3. Methodology

Literature on sentiment analysis shows that mere frequency of information arrival (news articles) may not explain change in an economic variable. What drives economic agents is the sentiments (i.e., quality) of information. Extracting sentiment from newspaper reports is one of the major contributions of our paper.

Main Idea

Inflation is measured as a year on year increase in the percentage of CPI (Consumer Price Index). The CPI is the overall price of a weighted combination of goods and services. The CPI basket is divided into 6 major sub baskets (described later). In our system, we classify each article into at most one of these baskets.

First, we use Naive Bayes Text Classification to determine each article's basket along with some assistance by a secondary rule based filter.

Second, we use a two-step approach to determining article level sentiment towards inflation -- identification of latent topics in subsections of the article, and identification of sentiment towards those topics. This is a well-studied topic in Natural Language Processing, most specifically in the analysis of sentiment from product reviews. The established problem is to take the user reviews of a given type of product (say, a particular laptop), and to aggregate sentiment towards specific product aspects (say performance, battery life, ergonomics, quality of the screen etc). The output is then useful to rate the product overall, as well as for each of its specific aspects. We work on a problem of a similar nature, and our system is detailed later. We use word IDFs (Inverse Document Frequencies) and Negation Detection as sub-modules here, described later.

Thirdly, we aggregate article level sentiment into basket level sentiment using an aggregation function.

Finally, we use these basket level sentiments as independent variables to predict the next month's inflation.

System Design and Algorithm Details

We develop a system over **Python** and several accompanying libraries to access large chunks of newspaper data, parse and process the news content, and make a well-founded estimate of the direction of inflation (CPI) in the near (next) month using sentiments generated from news content for the current month.

Input: To forecast inflation for a month (released in the middle of the month), we take as input news from 20th of the previous month to 10th of the current month. We can take as many newspapers as we like. For instance, for inflation of April 2015 (released on Apr 15), we use news from March 20th 2015 to April 10th 2015. Note that CPI numbers are released in the middle of a month (12th to 18th).

Output: For each month, we consume all the input news content and after processing, produce a single number which denotes the sentiment towards inflation for that month.

System Architecture: Our system consists of the following components arranged in the pipeline of Figure (1).

[Figure 1 to come here]

Classification of an Article into a CPI basket

We use news from two business dailies (Economic Times and Business Line). This is directly extensible to more newspapers. We crawl the news from the "Factiva" database online using a python tool. This is our input to the rest of the system.[4] This is component (1) in Figure (1).

The "Topic Classifier" module (2) takes as input a news article and classifies it into one of the sub baskets used to calculate the overall CPI. The sub baskets are *fuel*, *food*, *cloth*, *housing*, *intoxicants* (alcohol and various tobacco consumables) and *miscellaneous*.[5] The classifier can classify an article into one (or none) of these multiple baskets.

Following McCallum and Nigam (1998) we use the Naive Bayes classifier for this purpose. We use a bag of words model as opposed to binary models as our features. For training the model, we manually labelled about 4000 articles for two random months (November -December 2015 and 2016) to serve as our training set. It is interesting to note that the training dataset included the period of demonetisation.

[4] We may extend it to use news from "Business Standard" and "Financial Express".

[5] The relevant weights for each of these groups are 45.86%, 2.38%, 6.53%, 10.07%, 6.84%, and 28.32%, receptively.

In order to fine tune our system, we also supplemented the Naive Bayes model with a rule based filtering system which works by checking for the presence of certain words or phrases towards the beginning of the article. Starting from the Headline, we progressively taper the value of finding topic specific features in an article from our bag of words in this rule based system.

Article level sentiments

The next step in our system is to compute per article sentiment measures. The technical challenge in solving this problem is two-fold; topic identification and sentiment analysis. Broadly, there are two approaches to addressing these sub modules: A two stage pipeline which first identifies topics and then identifies sentiments (Hu and Liu, 2004; Popescu & Etzioni, 2005), or joint modelling of topics and sentiments (Jin and Ho, 2009; Lakkaraju *et al.*, 2011). We use the former approach in our system.

We chunk our articles at the sentence level, which simplifies the model to assume no cross sentence topic - sentiment interactions. For this, we use the nltk library in Python (Looper & Bird, 2002). We then build a latent topic classifier, which classifies a sentence into talking of one, both, or none of *price* and *quantity.*

Price/Quantity Classifier (Per line of text)

Information on volume (quantity produced) will have effect on price and hence inflation. Therefore, it is imperative that our classifier looks at these two keywords for every item basket. This module takes as input a line of text from an article and flags it as talking of price of the commodity, quantity of the commodity, of both, or of neither. Each line is classified into one of four categories - talking of neither price nor production, of only price or only production, or both. The idea is that inflation expectation is triggered by knowledge of either demand pull inflation (higher prices due to higher demand) or cost push inflation (higher prices due to lower supply).

For instance, “Oil prices unlikely to rise” would be classified as talking of only price, “OPEC cuts output on breakdown of talks” would be classified as talking of only production, “While prices seem to be rising, production is not falling commensurately, leading to inventory accumulation” would be classified as talking of both price and production. Likewise, “new policies on the horizon” would be classified as talking of neither price nor production.

Our topic classifier is a simple rule based system, which checks for the presence of certain words. For instance, some words for price are: *rate, demand* and *cost,* while some words for production are: *yield, supply* and *produce.* If a sentence contains words from both of these categories, the line is flagged as talking of both *price* and *production*.

Negation Detection (Per line of text)

This module takes as input a line of news text and checks whether parts of the line are negated using negation words as “not”, “unlikely”, “improbable” etc.

This is done in a two-step process. In the first step, an article is *dependency parsed* (Honnibal and Johnson, 2015). A dependency parser analyses and establishes the grammatical structure of a sentence, marking "head" words and their corresponding modifiers. For example, Figure 2 shows a sentence which has been dependency parsed.

[Figure 2 to come here]

In the second step, this dependency parse is used to determine words which have been negated. The system that we use for negation detection uses the marked dependencies as inputs (Gkotsis *et al*., 2016). This system codifies numerous rules over a dependency parse to detect negated contexts. For example, consider the line: "Oil prices not likely to rise". In the dependency parse of this sentence (shown in Figure 2), we see that "not" is a modifier of "likely" in terms of negation. This is detected by the system, and negation detector returns as output "Oil prices <negated scope> not likely to rise <\negated scope>". The additional markers inform us of the part of the sentence whose adjectives (in this case *rise*) are negated in meaning. This is utilised downstream in sentiment detection (see example in later section on sentiment detection).

IDF (Inverse Document Frequency) Calculator

TF (Term Frequency) of a word is defined as the number of times a word is seen in an article (document). IDF of a word is defined as a measure of the salience (or contribution to new information) of a word, based on how likely the word is to appear as a prior. If we merely use Term Frequency, we would not account for the fact that words which have a high prior of occurring will bias our estimate. For instance, determiners like *a*, *the*, will have naturally high TF, and they need to be down-weighted as they do not seem to be adding any sentiment related information. .

This module looks at sentiment words (generally adjectives) which we use downstream to infer sentiment from news articles, and calculates their IDF (Inverse Document Frequency) over our news articles dataset. Hence, for a word which appears in every document, IDF is zero, whereas it is most for a term which occurs in only one document.

Our IDF is calculated based on the news articles for the first six months of our dataset (months of July-December 2015). This has broad coverage to give us a principled and correct estimate of IDF.

Sentiment Detection

This module takes as input an article, and rates each line of text in the article with a number which describes whether the line indicates a rise or a fall in inflation. For this, it uses the information inferred upstream - whether the line talks about price/production, its negated scopes, and the relative strengths of sentiment adjectives. Sentiment adjectives which appear closer to the mentions of price/production are weighted more.

Each adjective is given a base rating of its IDF as determined upstream (Liu, 2012). Then, we look for the latent topic (*price* or *production)* it describes. The further away it is from the topic it describes, the more we down-weight its sentiment score. Finally, if it is negated, we negate its sentiment score.

Illustratively, consider the sentence "Oil prices not likely to rise". As explained earlier, this sentence is classified as talking of price (due to the word "*prices")*. Consider the IDF of *rise* to be 2.0. Suppose the dampening factor for inter word separation between the adjective (*rise)* and subject (*prices)* to be 0.8. We exponentiate the dampening factor to the number of words in between, which is 3 in this instance *(not likely to)*. Therefore, our sentiment score is $0.8^3 * 2.0 = 1.024$. However, the sentence adjectives are negated as determined by the output of the Negation Detection module ("Oil prices <negated scope> not likely to rise <\negated scope>"). Therefore, the value for *rise* is negated to -2.0, and we end up with a sentiment score of -1.024.

After scoring each line, the article sentiment score is a weighted sum of the scores for the individual lines. It strongly attributes more weight to the headline and lines towards the beginning of the article. This method provides an unscaled number (which takes all real values, but is likely in the range (-10, 10) which determines the strength (and direction) of sentiment of each line in the article).

CPI-Basket level sentiment

Once we have per article sentiments for all articles addressing one of the CPI sub baskets, we aggregate these into a single number. Each of the article sentiments is an un-scaled number. We dampen this un-scaled number to a number between (0, 1) (exclusive), using a dampening function. We use a demeaned sigmoid function for this purpose

$$(1) \qquad f(x) = \frac{1}{(1+e^{-x})} - \frac{1}{2};\ with\ f(0) = 0$$

The higher the number, the more it indicates a positive sentiment towards a rise in inflation. We need to demean (make the sigmoid zero mean) since we desire the behaviour that no news means no sentiment.

Inflation Prediction

We follow a short-term prediction approach with monthly updation of parameters. We use individual sentiment values for each article in a month, and aggregate them into a single predicted inflation number for the next month.

We learn a multivariate regression model over our training months, which is as follows:

$$(2) \quad I(t) = k + a\,S_{food}(t-1) + b\,S_{fuel}(t-1) + c\,S_{cloth}(t-1) + d\,S_{misc}(t-1) + e\,S_{gen}(t-1) + fI_{dm}$$

Where I(t) is inflation and $S_x(t)$ is sentiment for basket 'x' at time t (month t). I_{dm} is an indicator variable which is 0 for all months before demonetization and 1 afterwards. This is to inform the model that an external event, which affects public sentiment strongly, has occurred. We note that the introduction of this variable results into a significant improvement in our prediction model. Further, we taper this variable after about 10 months of demonetization, which is in line with the tapering effects of demonetization after this period. One of the major indicators for us to pick the period of 10 months is that other economic indicators such as GDP growth rates also show a recovery from this external shock after this 10 month period. We may introduce such variables for other similar external shocks to the economy, which strongly affect inflation.

We predict inflation for month *i*, using the sentiments of the previous month. Whenever the actual numbers are available, we replace the sentiment forecasts with the actual numbers to predict future inflation sentiment forecasts.

Input Size

Per newspaper, there are about 50 articles per day. That adds up to 1500 articles per month per newspaper. We currently work with two newspapers (*Economic Times* and *Business Line*), therefore we have around 3000 articles per month (Table 1).

[Table 1 to come here]

Each sub basket component of CPI has about 5 - 400 articles per month. The *general* category is not a CPI component, but we measure sentiment as belonging to this category if an article directly addresses inflation (and does not talk of any of the subcomponents of the CPI basket). This serves as a useful signal to measure sentiment towards overall inflation.

As a rough estimate, *misc* contains the most number of articles per month at 350 - 400 on average, and the other baskets contain 50 - 250 articles per month on average. There are very few articles (about 7 on average) per month addressing the *general* category.

4. Inflation Forecasts

It takes about 60 - 120 seconds of real time on a commodity PC to process an entire month's news and present the sentiment score for the month. The configuration used is an i5 2.30 GHz processor, with 2 cores and 2 threads per core (although at present we do not use multithreading). Our system is built in python and these measurements were made on Linux. We should note that this is unoptimized code, so within python itself, optimizing should lead to faster performance. Further, using a lower level language and libraries should lead to even faster performance. There is great scope for parallelization in our system, since each article is processed independently of the other. Leveraging this could lead to orders of magnitude improvements too.

We calculated monthly sentiment for each of the CPI baskets and performed a univariate regression of basket sentiment on basket inflation. We found high univariate correlation for the *food, fuel, cloth* and *miscellaneous (misc)* baskets, as well as high correlation between sentiment for the *general* category and overall inflation. The results are given in Table 2.

[Table 2 to come here]

Further, the multiple regression leads to a high significance for fuel, food as well as general sentiments. Table 3 reports the regression results of equation (2).

[Table 3 to come here]

Prediction Results

How is the predictive performance of the model we have developed? Figure 3 below charts a scatter plot of true inflation *versus* predicted inflation using our model. We achieve a correlation score of 0.70.

[Figure 3 to come here]

5. Limitations and Agenda of Further Work

One of the key limitations of our system is in the media that we use to make our predictions. We have used two English newspapers as of now. We could extend this to more newspapers, but we believe that the novelty of incoming news might not be very high, so this might not be relevant.

Our model considers only newspapers. We can try to predict macroeconomic variables using other sources of information such as social media posts and web search queries (Levenberg *et al*., 2014), to supplement our prediction.

Further, the technique we mention is a generic method to predict any macroeconomic variable. We can use it predict other such variables such as GDP growth rates, unemployment rates and the forex rate.

One of the key limitations of our system is that the only labelled data we have utilised is for sub basket topic classification. We believe that given more expert labelled training data, we can further improve our model and predict with much greater accuracy. For one, we have yet only observed (largely) direction of sentiment of an article. However, if the article talks about something relatively unimportant, we may wish to discount its contribution to overall sentiment. This may be based on the commodity it is talking about, its source, the tense of the text, and the timing of the article (within our prediction horizon which is monthly), none of which we have yet included in our model. Future work in this direction may aim to incorporate such factors. This will require much larger amounts of finely labelled training data.

One of the principled limitations of our approach is that in a developing country like India (versus say, the United States), financial news often does not percolate (fast enough) to the rural population. Hence, using only financial news sentiment to measure perception towards inflation may not be good enough. To this end, we performed our predictions (as above) on urban inflation instead of overall inflation, and saw a slight improvement in results.

6. Concluding Observations

Measuring inflation expectation is a key component of economic and financial policy making. We use text mining to predict inflation expectations. This experiment of investigating whether inflation perception can be measured using newspaper text essentially consists of two sub experiments. The first is to check how well an automated system, when invoked on a single newspaper article, can infer its general sentiment about inflation. This is to say that if an expert (economist) were to read the same article and conclude that this article says price is going to rise, the system should be able to provide the same result. The second is to check whether the aggregated sentiment from the monthly news remain significant and relevant to predict actual inflation numbers?

The first of these subtasks, we must say, has achieved high accuracy. When we evaluated the sentiment inferred by the system on individual articles by hand, the system performed accurately almost all the time, and even corrected the human labeller on some occasions! To this extent we believe that the underlying Natural Language Processing used to generate such sentiment may not benefit much from improvements, as our "simple" model appears to do quite well. The most significant way to improve this now could be to target a finer level of granularity in terms of inferring article sentiment. That is to say, the challenge is to build a system which not only tells us whether price is going to rise or fall, but also by how much. Note that this task is indeed hard for even an expert to complete.

This is a novel approach to quantify public sentiment towards inflation by inferring it from newspaper text for India. Besides, it compares well with the IESH (Inflation Expectation Survey of Households) conducted by the RBI in order to measure inflation perception as IESH achieves a correlation coefficient of 0.50 in predicting the actual inflation figure, whereas we obtain 0.70 with our method. We hope that such approaches to predicting macroeconomic variables are investigated further by the research community, and fruitful results applied in public policy decision making.

References

Boschen J. and S. Mills (1995): "The relation between narrative and money market indicators of monetary policy", *Economic inquiry*, January, pp 24-44.

DeLong, J. Bradford, Macroeconomics, McGraw-Hill, 2002

Friedman, Milton and Anna J Schwartz (1972): A Monetary History of United States, 1867 - 1960, Princeton: Princeton University Press.

Nyman, Rickard., David Gregory, Sujit Kapadia, Paul Ormerod, David Tuckett & Robert Smith (2016): "News and narratives in financial systems: Exploiting big data for systemic risk assessment", Bank of England *Working Paper* No. 704.

Romer, Christina D. And David H. Romer (2004): "A New Measure of Monetary Shocks: Derivation and Implications", *American Economic Review*, September, 94(4): 1054 - 1083.

Shapiro, Adam Hale, Moritz Sudhof, and Daniel Wilson (2018): "Measuring News Sentiment". *Federal Reserve Bank of San Francisco Working Paper*, available at https://doi.org/10.24148/wp2017-01

Brandt, Michael W, and L. Gao (2019): "Macro fundamentals or geopolitical events? A textual analysis of news events for crude oil". *Journal of Empirical Finance* , 51, pp 64-94

Smith, Noah (2018): "Want a Recession Forecast: Ask a Machine", *Bloomberg*, May 13, 2018, available at https://www.bloomberg.com/view/articles/2018-05-11/want-a-recession-forecast-ask-a-machine-instead-of-an-economist

Hu, M. and B. Liu (2004): "Mining and summarizing customer reviews", *Knowledge and Data Discovery (KDD)*, pp 168-177

Popescu A.M., O. Etzioni (2005): "Extracting Product Features and Opinions from reviews", *Empirical Methods in Natural Language Processing (EMNLP)*, pp 339-346

Jin, W. and H.H. Ho (2009): "A novel lexicalized hmm-based learning framework for web opinion mining", *International Conference on Machine Learning (ICML),* pp 465-472

H. Lakkaraju, C.Bhattacharya, I. Bhattacharya, S. Merugu (2011): "Exploiting coherence in reviews for discovering latent facets and associated sentiments", *Siam International Conference on Data Mining (SDM),* pp 498-509

McCallum, A. and K.Nigam (1998): "A comparison of event models for naive bayes text classification", *Association for the Advancement of Artificial Intelligence (AAAI)*, pp 41-48

Looper, Edward and Steven Bird (2002): "NLTK: The natural language toolkit", *Effective Tools and Methodologies for Teaching Natural Language Processing, (ETMTNLP),* pp 63-70.

Honnibal, Matthew and Mark Johnson (2015): "An improved, non-monotonic system for dependency parsing", *Empirical Methods in Natural Language Processing (EMNLP)*, pp 1373-1378.

Gkotsis, George, Sumithra Velupillai, Anika Oellrich, Harry Dean, Maria Liakata and Rina Dutta (2016): "Don't Let Notes Be Misunderstood: A Negation Detection Method for Assessing Risk of Suicide in Mental Health Records", *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology,* pp 95-105

Liu, Bing (2012): Sentiment Analysis and Opinion Mining, Morgan and Claypool Publishers

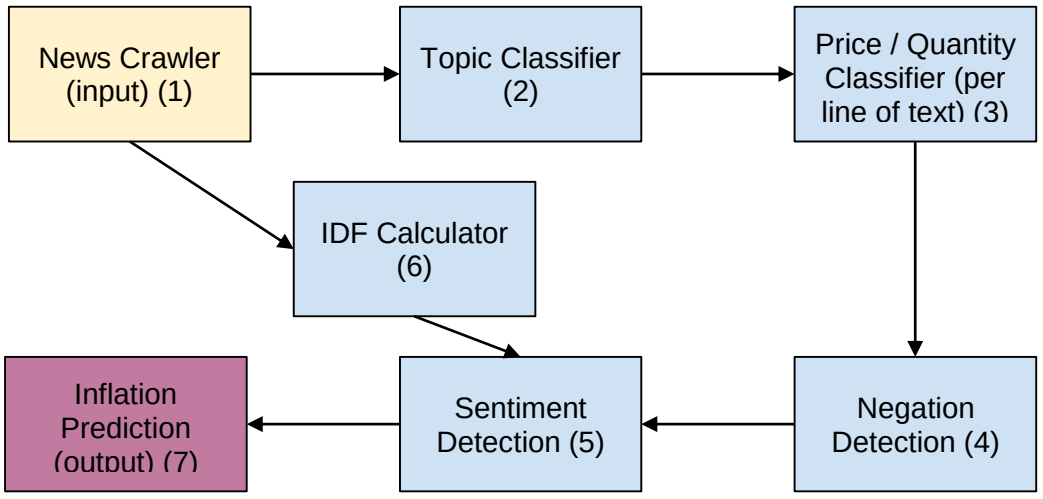
Levenberg, Abby, Stephen Pulman, Karo Moilanen, Edwin Simpson and Stephen Roberts (2014), "Predicting Economic Indicators from Web Text Using Sentiment Composition", *International Journal of Computer and Communication Engineering* vol. 3, no. 2, pp. 109-115

Table 1: Summary statistics for number of articles in each component of CPI				
Topic	Mean	Median	Min	Max
Fuel	108	100	38	234
Food	149	138	94	239
Cloth	30	29	13	48
House	71	73	40	132
Pan and intoxicants	6	6	1	15
Miscellaneous	337	338	236	446
General	6	5	0	22

Table 2: Univariate Regressions of each components of CPI (General form: $I_i(t) = k + a\ S_i(t-1)$; for i = food, fuel, cloth, misc, general)					
	coefficient	coefficient values	Standard errors	t-values	probabilities
Food	k	0.43	1.07	0.40	0.69
	a	0.4797	0.068	7.099	0.0
Fuel	k	2.44	0.598	4.076	0.0
	a	0.2326	0.040	5.842	0.0
Cloth	k	5.25	0.720	7.29	0.00
	a	0.8382	0.25	3.42	0.00
Misc	k	2.40	0.75	3.187	0.002
	a	0.16	0.041	3.801	0.00
General	k	5.17	0.51	10.11	0.00
	a	0.42	0.17	3.56	0.001

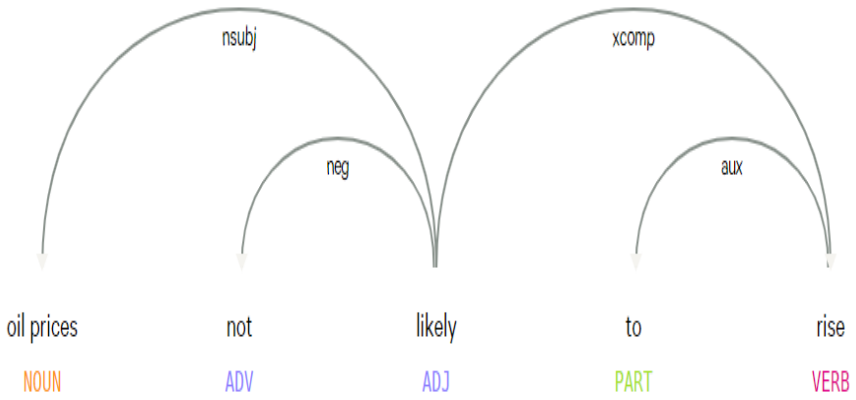
Table 3: Projecting General CPI Inflation: Base line Regression				
coefficient	coefficient values	Standard errors	t-values	probabilities
k	2.1106	0.849	2.486	0.015
a	0.0848	0.041	2.071	0.042
b	0.2377	0.048	5.000	0.000
c	0.1760	0.179	0.986	0.328
d	-0.0588	0.051	-1.147	0.255
e	0.1697	0.084	2.023	0.047
f	-2.2455	0.463	-4.851	0.000

Figure 1: System pipeline



Source: Authors

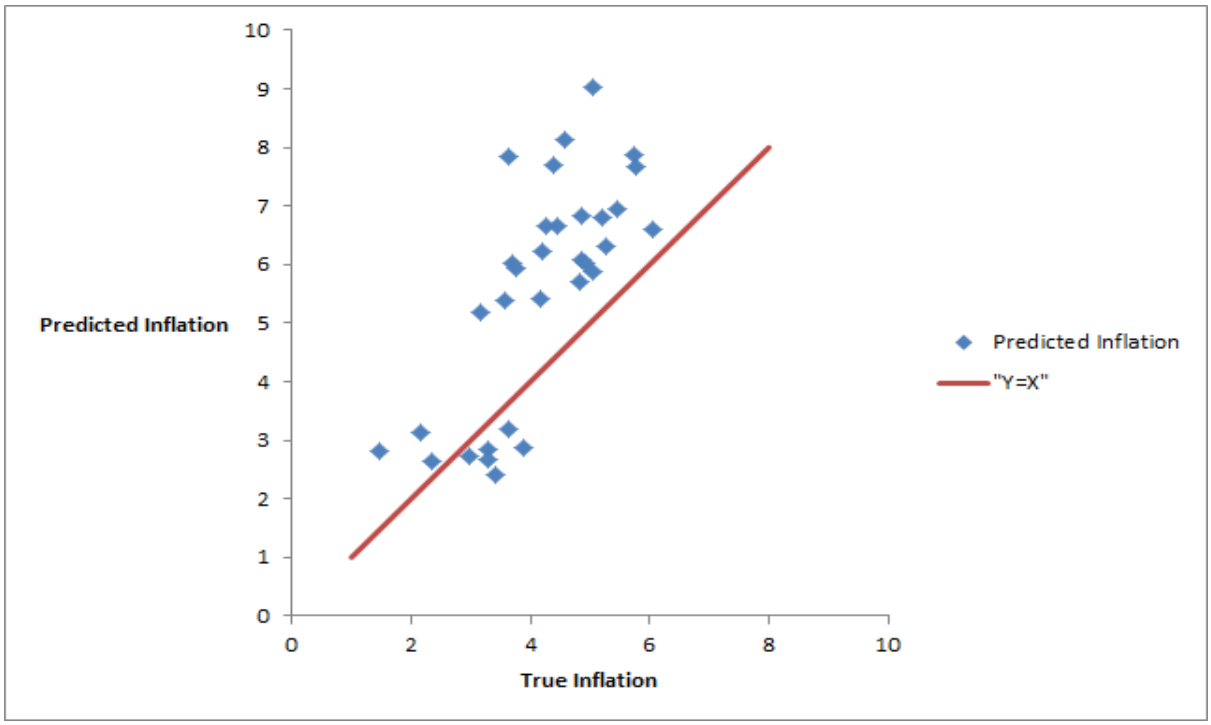
Figure 2: Example of a Dependency Parse



Legends: nsubj: Nominal Subject; neg: Negation Modifier; xcomp: Open Clausal Complement; aux: Auxilliary; ADV: Adverb; ADJ: Adjective; PART: Participle.

Source: Displacy Library (Part of Spacy), available at https://spacy.io/usage/visualizers

Figure 3: Inflation (True *versus* Predicted)



Source: Authors' Calculations