UNIVERSITÀ di VERONA

PhD in ECONOMICS AND MANAGEMENT

ECONOMICS CURRICULUM

CYCLE XXXIII

Department of Economics

PhD School of Legal and Economic Sciences

University of Verona

Text Mining in Macroeconomics and Finance Using Unsupervised Machine Learning Algorithms

Supervisor: Marco Minozzo

Head of the PhD Program: Roberto Ricciuti

PhD candidate: Carlos Moreno Pérez

Acknowledgements

I would like to express my gratitude to Marco Minozzo for his invaluable guidance as my PhD advisor. The academic expertise, patience, enthusiasm, and all the encouragement he provided me with along the years have allowed me to overcome the challenges posed by the completion of this dissertation. I could not have imagined having a better advisor and mentor for my studies.

I also express my warm thanks to Roberto Ricciuti for his help, advice and guidance at the University of Verona. I would like also to thank the head of the PhD school of Law and Economics, Matteo Ortino. I am grateful to Dalibor Stevanovic for the advice and help during my visiting at UQAM and all the department of economics of UQAM for their warm reception.

I am also especially grateful to Gianluca Grimalda, Nikolas Müller-Plantenberg, Juan Carlos Moran Alvarez, Fernando Úbeda Mellina and Prosper Lamothe Férnandez for their enormous help in the PhD application process.

I would like to thank also to all the professors and research assistants of the courses I followed at University of Verona and University of Padova during the first year of PhD for their unvaluable preparation of research skills and help, Francesco De Sinopoli, Claudia Meroni, Claudio Zoli, Angelo Zago, Martina Menon, Letizia Pellegrini, Alberto Peretti, Cecilia Rossignoli, Sara Moggi, Alessandro Zardini, Ludovico Bullini, Francesco Pascucci, Nunzio Cappuccio, Chiara Dal Bianco, Antonio Nicolo, Sonal Yadav, Marco Bertoni, Lorenzo Forni, Luigi Grossi, Mauro Mussini, Marcella Veronesi, Luca Zarri, Maria Vittoria Levati and Chiara Nardi. I would like to thank also all the department of Economics of the University of Verona for their comments in my seminar and for preparing inspiring seminars every week such as Roberto Renò, Alessandro Sommacal, Emmanuele Bracco, Riccardo Fiorentini, Giam Pietro Cipriani, Alessandro Bucciol, Francesca Rossi, Simone Quercia, Luca Taschini, Federico Perali and Lubian Diego.

I thank for their invaluable suggestions Prof. Paola Zuccolotto and Prof. Stefano Tonellato who carefully read a draft of the thesis.

I am also grateful to all the administrative and IT stuff of the University of Verona, Giovanni Baietta, Rosalba Oriente, Silvia Moretto, Alessandra de Amicis, Michael Reiter, Luciano Vallone, Matteo Lovato, Chiara Bernardi, Leonardo Palma, Valter Pasquali and Francesca Limberto. I would like to thank also to Nicola Tommasi for his support downloading the Bloomberg data and his lessons of Stata. Finally, I gratefully acknowledge the financial support from the University of Verona.

I am enormously thankful to my parents, Pilar and José, who made it possible to achieve every objective and every dream that I had in my life, to my grandparents (Fernanda, Santa, Jesús and Antonio), my brothers (Alejandro and Alberto), my uncles and cousins for their great support. I would like to mention also the support of my friends of Sevilla, Lolo, Jessica, Antonio, Pedro, Minia, Pepe, Lucia, Nicolas, Cristina, Manuel, Amelia, Pedro, Elodie, Mario, Alberto, Marta, Javier, Irene, Francisco, Saray, Regina, Paco, Juan Antonio, Javier, Jose, Ana and the forces they give me.

Finally, I would like to thank again my supervisor Marco Minozzo who always supported me, guided me and wisely advised me in these years. I could not have a better supervisor than him.

Summary

This thesis presents three different applications to macroeconomics and finance of text mining techniques based on unsupervised machine learning algorithms. In particular, these text mining techniques are applied to official documents of central banks and to newspaper articles written in English and Spanish. The implementation of these techniques involved a considerable preprocessing work to remove paragraphs and articles not relevant for the analysis. To the official documents of the central banks, we assigned tags to each paragraph to indicate the date and other useful information, eliminated stop words and reduced inflected words by stemming. We then applied various computational linguistic unsupervised machine learning algorithms such as Latent Dirichlet Allocation (LDA), the Skip-Gram model and K-Means to construct text measures. These machine learning methods have an important advantage over dictionary methods since they use all terms of the text to represent paragraphs in a low-dimensional space instead of using parts of them. Moreover, unsupervised machine learning algorithms allow to create text measures without the need for human intervention and also using less time. Some of these unsupervised machine learning algorithms, which were already available for the English language, have been adapted to the Spanish language. We produced simple measures of the content of the communication to identify the topics, that is, the themes or subjects, and the tone, that is, the sentiment or degree of uncertainty, of the text. Then, we investigated the relationship between these uncertainty indices and key economic variables in macroeconomics and finance using Structural VAR and Exponential GARCH models.

The first paper investigates the relationship between the views expressed in the minutes of the meetings of the Central Bank of Brazil's Monetary Policy Committee (COPOM) and the real economy. It applies various computational linguistic machine learning algorithms to construct text measures of the minutes of the COPOM. Firstly, we infer the content of the paragraphs of the minutes with Latent Dirichlet Allocation and then we build an uncertainty index for the minutes with Word Embeddings and K-Means. Thus, we create two topic-uncertainty indices. The first topic-uncertainty index is constructed from paragraphs with a higher probability of topics related to 'general economic conditions', whereas the second topic-uncertainty index is constructed from paragraphs with a higher probability of topics related to 'inflation' and the 'monetary policy discussion'. Finally, via a Structural VAR we explore the lasting effects of these uncertainty indices on some Brazilian macroeconomic variables. Our results show that, in the period from 2000 to 2019, greater uncertainty leads to a decline in inflation, in the exchange rate, in industrial production and in the retail trade. From 2000 to 2016, we find a different effect of the two topic-uncertainty indices on inflation, exchange rate and industrial production.

The second paper studies and measures uncertainty in the minutes of the meetings of the board of governors of the Central Bank of Mexico and relates it to monetary policy variables. In particular, we conceive two uncertainty indices for the Spanish version of the minutes using unsupervised machine learning techniques. The first uncertainty index is constructed exploiting Latent Dirichlet Allocation, whereas the second uses the Skip-Gram model and K-Means. We also create uncertainty indices for the three main sections of the minutes. We find that higher uncertainty in the minutes is related to an increase in inflation and money supply. Our results also show that a unit shock in uncertainty leads to changes of the same sign but different magnitude of the inter-bank interest rate and the target interest rate. We also find that a unit shock in uncertainty leads to a depreciation of the Mexican peso with respect to the US dollar in the same period of the shock, followed by an appreciation in the subsequent period.

The third paper investigates the reactions of US financial markets to newspaper news from January 2019 to the first of May 2020. To this end, we deduce the content and sentiment of the news by developing apposite indices from the headlines and snippets of the New York Times, using unsupervised machine learning techniques. In particular, we use Latent Dirichlet Allocation to infer the content (topics) of the articles, and Word Embedding (implemented with the Skip-gram model) and K-Means to measure their sentiment (uncertainty). In this way, we arrive to the definition of a set of daily topic-specific uncertainty indices. These indices are then used to find explanations in the behaviour of the US financial markets by implementing a batch of EGARCH models. In substance, we find that two topic-specific uncertainty indices, one related with COVID-19 news and the other with trade war news, explain much of the movements in the financial markets from the beginning of 2019 up to the first four months of 2020.

Contents

Chapter 1

‘Making Text Talk’: The Minutes of the Central Bank of Brazil and the Real Economy

1.1 Introduction

Central bank communications are an important instrument in the toolbox, able to influence financial markets and the real economy. In particular, the communications of central banks provide relevant information to the markets with the aim of reducing uncertainty about their future policy decisions. Central banks communicate with the markets in different ways such as press conferences, statements of monetary policy decisions, inflation reports and the minutes of monetary policy meetings.

Central bank communications are of great importance since they provide a hint of the intensity of the risks to price stability and growth. The higher the risks, the greater the likelihood of monetary policy intervention (Rosa and Verga; 2007). In other words, the higher the degree of uncertainty about current economic conditions or monetary policy, the greater the likelihood of a change in interest rates or other monetary policy actions. The release of this information should help agents to reduce the uncertainty over the future state of the economy and influence inflation expectations.

The US Federal Open Market Committee (FOMC) opts to publish its minutes some days after the meeting. Several central banks in Latin America - such as the Central Banks of Colombia, Mexico, Chile and Brazil - also publish the minutes of monetary policy meetings.

In the past, investigations into central bank communications processed the information in the text manually and categorized it as dovish or hawkish. Several papers used this manual classification of the text to investigate how the communications of the Central Bank of Brazil are related to changes in interest rate expectations (Costa-Fiho and Rocha, 2010; Cabral and Guimaraes, 2015; Garcia-Herrero, Girandin and Dos Santos, 2017). However, this methodology can introduce some bias due to personal interpretations and requires a huge amount of work. Some papers have attempted to overcome these issues by using dictionary methods, i.e. lists of words related to a sentiment or a topic. Dictionary methods lead to more consistent and faster topic and tone analysis. Dictionary techniques can determine the topic or theme of a newspaper article by searching for words related to different topics or subjects. For instance, an article that contains the words 'trade' and 'European Union' could be linked to the topic or theme 'European Union trade'. Dictionary techniques can also determine the tone by a predefined list of words related to a sentiment such as positive, negative, ambiguity or uncertainty. For instance, the sentiment dictionaries Loughran and McDonald (2011) and Harvard IV-4 Psychological are normally used in the economic literature to determine the sentiment or tone of the text. In particular, the sentiment measures are constructed via the relative frequency of the dictionary words. Chague, De-Losso, Giovannetti and Manoel (2015) apply this methodology for the communications of the Central Bank of Brazil. Nonetheless, dictionary methods still introduce some bias in the analysis since the words related to a sentiment are pre-established by the researchers with texts that might not take into consideration all the words of the text to be analyzed.

Machine learning methods are an attempt to overcome these issues by providing more objective and systematic methods. There are supervised and unsupervised machine learning algorithms, the former dealing with a set of input variables (X) that are used to predict an output variable (Y) and the latter trying to find meaningful relationships between the input data (X) without relying on any output variable (Y).

Some investigations explore the capabilities of supervised machine learning algorithms for text mining to predict the tone of the document, which is the sentiment of the text. For instance, with the supervised algorithm Support Vector Machines, Tobback et al. (2018) construct an uncertainty index for Belgium from several Belgian newspapers. However, supervised machine learning techniques work as dictionary methods since the researchers use a tag to determine the sentiment of each text document in a training database. For instance, the researchers indicate with a binary variable if the paragraph provides certain or uncertain information about the state of the economy. Furthermore, supervised machine learning techniques are also used to predict events. For instance, Garcia-Uribe (2018) uses Random Forest and Fuzzy Forest to predict tax bill approvals in the US Congress with the 177 most frequent stems appearing in US television news.

Economic investigations use unsupervised machine learning techniques to deduce content or topics. These techniques include Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) and Dynamic Topic Model (DTM). For instance, Hendry and Madeley (2010) use Latent Semantic Analysis to analyze the communications of the Central Bank of Canada. Additionally, Ortiz, Rodrigo, and Turina (2017) use the Dynamic Topic Model jointly with the Loughran McDonald dictionary to investigate the relationship between the communications of the Central Bank of Turkey and real and market variables. Finally, LDA consists in a generative probabilistic model of a corpus. The basis of LDA is that documents are depicted as a random combination of latent topics, where each topic is represented by a distribution of words (Blei et al, 2003). According to Hansen, McMahon, and Prat (2017), machine learning methods have an important advantage since they use all terms of the text to depict paragraphs in low-dimensional space instead of using parts of them as dictionary methods. They argue that machine learning techniques detect the most significant words in the data instead of imposing them. Finally, Hansen, McMahon, and Prat (2017) state that a cognizable trait of LDA compared to other algorithms for dimensionality reduction is that it is fully probabilistic. In their paper, they use LDA for topic analysis and dictionary techniques for tone analysis to investigate the communication patterns of members of the FOMC through a natural experiment. Other papers in the literature use LDA to study central bank communication. Hansen, McMahon and Tong (2019) use Elastic Net following Zou and Hastie (2005) to identify the topics (obtained through LDA) of the Bank of England inflation report with the greatest predictive power. Larsen and Thorsrud (2019) demonstrate that the topics obtained from a major Norwegian newspaper through LDA have important predictive power for key economic variables, especially asset prices.

Unsupervised machine learning techniques are also used to deduce the sentiment of the text. They include Word Embeddings introduced by Mikolov et al.(2013a) and Mikolov et al. (2013b). For instance, Soto (2021) uses Word Embeddings to investigate how commercial banks communicate in their quarterly conference calls. He constructs the Word Embeddings with the Skip-Gram model, in particular, applying the Skip-Gram model to a text comprising transcripts of commercial bank earnings calls. When the Skip-gram model is computed, Soto (2021) uses an unsupervised machine learning method called K-Means to find the vector words (Word Embeddings) closest to the vector representations of ‘uncertainty’ and ‘uncertain’ and so constructs a list or dictionary of uncertain words. This ‘uncertain’ dictionary has the advantage of being based on the text sample compared to pre-established dictionaries that might not fit the text. Then, Soto (2021) creates an uncertainty index with the frequency of these words in conference calls. Later, he applies LDA and combines the topics with the uncertainty index to create topic-uncertainty indices.

Section 2 describes the minutes of the Monetary Policy Committee (COPOM) of the

Central Bank of Brazil. The minutes of the COPOM contain relevant information about the state of the economy, inflation expectations and the reasons behind monetary policy decisions. This paper investigates the effect of a shock in uncertainty in the minutes of the Monetary Policy Committee (COPOM) on macroeconomic variables. The COPOM meets a fixed number of times a year and its minutes are released the week after the meeting. Costa-Filho and Rocha (2010) argue that the minutes of the COPOM influence financial markets because they provide information about how monetary policy decisions are taken. These authors also argue that the minutes provide information about inflation expectations and the economic situation that economic agents might have not considered. They find evidence that the release of the minutes of the Central Bank of Brazil help to reduce the volatility of 'swap pre x DI' interest rates for maturities of 30, 180 and 360 days. The ability to persuade makes the minutes of the COPOM one of the key instruments of the monetary policy of the Central Bank of Brazil for changing market expectations.[1]

Our main objective is to construct new measures of communication for the COPOM minutes. For that purpose, we suggest simple measures of communication to identify the topic and tone of the minutes of the Central Bank of Brazil.

Section 3 applies LDA to the minutes of the COPOM to understand the content of each paragraph. To the best of our knowledge, this is the first paper to use LDA to investigate the communications of the Central Bank of Brazil. We identify the paragraphs that have a higher probability of topics related to 'general economic conditions' and the paragraphs that have a higher probability of topics related to 'inflation and the monetary policy decision'.

Section 4 applies the Skip-Gram and K-Mean models following Soto (2021) to construct a list of words similar to 'uncertain', 'uncertainty', 'uncertainties' and 'fears', aka an 'uncertainty' dictionary. This 'uncertainty' dictionary is assumed to be less biased and better adapted to the text than pre-established sentiment dictionaries such as Loughran and McDonald (2011) since our dictionary is constructed with the text to be analyzed. Then, we build an uncertainty index for the minutes of the Central Bank of Brazil by counting the relative frequency of the words in our 'uncertainty' dictionary. However, there is still some degree of discretionality depending on the parameters selected to apply the Skip-Gram model since this might change some of the words in the dictionary. We then construct topic-uncertainty measures by combining the results of LDA and the Skip-Gram model for a better understanding of uncertainty shocks in paragraphs that discuss different topics. Specifically, we create two topic-uncertainty indices, one with the para-

[1] Swap pre x DI are interest rate swap agreements with pre-fixes rates that are negotiated in the Stock Exchange BM&FBovespa.

graphs more likely to include a group of topics related to ‘general economic conditions’, and a second topic-uncertainty index with the paragraphs more likely to have a group of topics related to ‘inflation’ and the ‘monetary policy decision’.

Section 5 analyzes the effect of the minutes and topic-uncertainty indices in the Brazilian real economy through a Structural Vector Auto-regression (SVAR) model.

Section 6 provides the results. Our results from 2000 to July 2019 show that higher uncertainty in the minutes of the COPOM leads in the same period to a decrease in industrial production, inflation and retail sales. Also, a unit shock in uncertainty of the minutes is associated with a depreciation of the exchange rate. Moreover, a unit shock in the two topic-uncertainty indices has diverse effects on the exchange rate, inflation and industrial production in the period 2000-2016. Finally, Section 7 presents our conclusions.

1.2 Minutes of the Central Bank of Brazil

Some decades ago, inflation in Brazil was a major economic issue. Brazil suffered hyperinflation for almost 15 years from 1980 to 1994, during which inflation racked up an astonishing 13,342,346,717,617.70 percent. It was stopped by the introduction of the ‘Real Plan’ (‘Plano Real’) which included the introduction of a new currency the ‘Real’ and the privatization of state monopolies. In the 15 years after the introduction of the ‘Real Plan’, inflation was significantly reduced, totaling 196.87 percent over the period (Corrado, 2013).

In 1999, an inflation targeting regimen was adopted which allowed the ‘Real’ to fluctuate in response to market foreign-exchange mechanisms. The same year, the Central Bank of Brazil’s Monetary Policy Committee (COPOM) was created to increase transparency and trust in the monetary policy decision-making process. The COPOM is responsible for setting the stance on monetary policy and the short-term interest rate. The main goal of the COPOM is to achieve the inflation target established by the National Monetary Council. Moreover, the Central Bank of Brazil releases four types of documents related to monetary policy. First, an inflation report is published at the end of every quarter. Second, a summary of the decision of the COPOM is published after each meeting. Third, a focus report is released weekly containing projections for inflation, economic activity, the Selic rate and other economic indicators. Finally, the minutes of the meetings of the COPOM are published the week after the meeting.

In this paper, we analyze solely the minutes of the meetings of the COPOM. Our sample comprises all the minutes of the COPOM from the last meeting in 1999 to September

2019, which are available on the website of the Central Bank of Brazil. Hence, we have 184 minutes of the COPOM. From the end of 1999 until 2005, the COPOM met once a month, with an additional meeting in 2002. In 2006, the COPOM reduced the number of yearly meetings to eight. The meetings last two days. On the first day, current economic and financial conditions are illustrated by the various departments and discussed by the members of the COPOM. On the second day, the members and head of the Research Department discuss the updated projections for inflation. Then, the COPOM takes its monetary policy decision. Since the 200th meeting of the COPOM in 2016, the statement of the final decision of the COPOM has included a summary of the domestic risks for the baseline scenario. Hence, part of the information in the minutes is not new for economic agents.

We use the English version of the minutes of the COPOM as a proxy of the Portuguese version. The English version is published one or several days after the Portuguese version. Since the 94th meeting in 2004 until the 199th meeting in 2016, the Portuguese version of the minutes was released on Thursday at 8:30 a.m. the week after the meeting. Since the 200th meeting, in 2016, the Portuguese version of the minutes is released on Tuesday at 8:30 a.m. the week after the meeting. The minutes are made public before the Brazilian Stock Exchange (BM&FBOVESPA Exchange) opens at 9:30.

1.3 Topic Analysis: Latent Dirichlet Allocation

We use simple measures of communication to identify the topic and the tone of the minutes of the Central Bank of Brazil. First, we apply Latent Dirichlet Allocation to identify the content or tone of each paragraph. We identify the paragraphs of the minutes that have a higher probability of the group of topics related to the current state of the economy, as well as paragraphs that have a higher probability of the group of topics related to inflation and monetary policy decisions. We then compute the tone of each paragraph. By tone, what is meant is the sentiment or degree of uncertainty in each paragraph of the minutes. To compute the tone, we apply the Skip-Gram and K-means algorithms to create a list of words similar to 'uncertain', 'uncertainty', 'uncertainties' and 'fears'. Later, we build an uncertainty index by counting the number of times words from our 'uncertainty' list appear in each paragraph. Finally, we combine both topic and tone measures to construct two topic-uncertainty measures. The first topic-uncertainty index is constructed from paragraphs with a higher probability of topics related to general economic conditions. The second topic-uncertainty index is constructed from paragraphs with a higher probability of topics related to inflation and monetary policy decisions.

1.3.1 Latent Dirichlet Allocation model

Latent Dirichlet Allocation (LDA) is a machine learning technique introduced by Blei, Ng and Jordan (2003) that can be used for textual analysis. It is an unsupervised machine learning technique that aims to identify the topics or content of the text of all the documents interest without a person needing to read the text. The capacity of LDA to produce easy interpretable topics is one of its advantages. In order to do that, a name is assigned to each topic, for instance, 'industrial production' since the words most likely to appear are 'industry', 'production', 'goods', 'workers' and 'supply'. This labelling does not affect the results.

LDA is based on a generative probabilistic model of a corpus. The corpus comprises a set of documents that are indexed by $(d = 1, 2, ..., D)$. Each document, d, is a series of N_d words ($n = 1, ..., N_d$) represented by $\mathbf{w}_{dn} = (w_{d1}, w_{d2}, \ldots, w_{dN_d})$, where w_{d1} is word 1 of document d. In our paper, a document is a paragraph of the minutes and the corpus comprises all the paragraphs in all the minutes. The total number of words in the corpus is equal to $\sum_{d=1}^{D} N_d = N$. Moreover, there are $\{1, ..., V\}$ unique terms in our corpus in the list of N terms.

LDA assumes a generative process that produces two main outputs.

1. The first important output is the probability distribution of words over topics (K), which is represented by β_k. Words can be assigned to different topics. In other words, each topic is a group of weighted words in a similar theme. LDA allocates a symmetric Dirichlet prior η to the distribution of words in each topic, β_k, for $k = 1, ..., K$.

$$\beta_k \sim \text{Dirichlet}(\eta). \tag{1}$$

2. The second output is the probability distribution of topics over documents. In other words, a document consists of a mixture of K latent topics given by θ_d, that is the probability of topic k in document d (Hansen, McMahon, and Prat, 2017). A Dirichlet prior α is selected for the distribution of topics across documents, θ_d, for $d = 1, ..., D$.

$$\theta_d \sim \text{Dirichlet}(\alpha). \tag{2}$$

Theoretically, each word w_{dn} in document d is created from the following two-step process:

1. First, each word w_{dn} in document d is independently assigned to a topic. The topic assignment of each word w_{dn} is represented by z_{dn}. In addition, the topic assignment is selected from the multinomial distribution θ_d. The topic assignments are unobserved, becoming latent variables.

$$z_{dn} \sim \text{Multinomial}(\theta_d). \tag{3}$$

2. Second, a word w_{dn} is selected from the multinomial distribution β_k depending on the topic assignment z_{dn} of the previous step. This represents the word-topic assignment, $w_{z_{dn}}$.

$$p(w_{dn}|z_{dn}, \beta) \sim \text{Multinomial}(\beta_{z_{dn}}). \tag{4}$$

However, the distributions of the two main outputs of LDA (topics per documents and words per topics) are unobservable. To compute both outputs, we use a Bayesian method that assumes prior distributions to compute the posterior distribution. In fact in LDA, the inference issue is to calculate the posterior distribution over $\mathbf{z}_{dn}$, $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ given the Dirichlet parameters and the corpus $\mathbf{w}$.

$$Pr(\mathbf{z} = z_i|\mathbf{w}, \boldsymbol{\theta}, \boldsymbol{\beta}) = \frac{Pr(\mathbf{w}|\mathbf{z} = z_i, \boldsymbol{\theta}, \boldsymbol{\beta})Pr(\mathbf{z} = z_i|\boldsymbol{\theta}, \boldsymbol{\beta})}{\sum_{z_i} Pr(\mathbf{w}|\mathbf{z} = z_i, \boldsymbol{\theta}, \boldsymbol{\beta})Pr(\mathbf{z} = z_i|\boldsymbol{\theta}, \boldsymbol{\beta})}. \tag{5}$$

We cannot estimate a closed-form solution for the posterior distribution of the model described above since the computation of the denominator in Equation (5) is an intractable problem. We should approximate the posterior distribution by the Markov chain Monte Carlo Method (MCMC) that provides a stochastic approximation of the true posterior. We select the Gibbs sampling algorithm among the various Markov chain Monte Carlo methods to estimate LDA.[2] The Gibbs sampling algorithm for LDA integrates the terms θ_d, β_k and samples only z_{dn} (Hansen, McMahon, and Prat, 2017).

1.3.2 Corpus pre-processing and LDA estimation

In order to apply LDA, we manually transform the PDF of each set of minutes into text files. We remove from the minutes the parts that are not relevant for the LDA model such as the cover, the introduction, the footnotes and acronyms. We also assign tags to each paragraph to identify the date, the number and section of the minutes. All the words are changed to lower case and the data are 'cleaned' before applying LDA. The

[2] We implement the Latent Dirichlet Allocation model using the code delivered by Hansen, McMahon, and Prat (2017).

‘cleaning’ data process for LDA requires three steps eliminating non-relevant information from the text. The first step is to remove the punctuation and stop words such as ‘the’, ‘all’, ‘because’, ‘this’, not relevant since they provide no information about the theme of the paragraph. Second, we stem the remaining words. Stemming is a process that consists in reducing words into their word stem or base root. For instance, the words ‘inflationary’, ‘inflation’, ‘consolidate’ and ‘consolidating’ are transformed into their stem ‘inflat’ and ‘consolid’, respectively. Finally, we rank these stems according to the term frequency-inverse document frequency (tf-idf). This index grows proportionally with the number of times a stem appears in a document. However, it decreases by the number of documents that contain that stem. This index serves to eliminate common and unusual words. We disregard all stems that have a value of 3,000 or lower. This cutoff of 3,000 seems reasonable with the tf-idf ranking.

In our research, we apply LDA with 9 topics to the 9,484 paragraphs that comprise all the minutes from the end of 1999 to September 2019. In our analysis, each paragraph corresponds to a document of the corpus. Our corpus comprises 2,900 unique stems and the total number of stems is 450,174.

Furthermore, we follow the suggestions of Griffiths and Steyvers (2004) to set the two hyperparameters of the Dirichlet priors. First, we set the Dirichlet prior on topics to $200/V$, where V is the number of single or unique vocabulary items. Second, we set the hyperparameter of the Dirichlet prior on document-topic distributions equal to $50/K$ where K is the number of topics (Hansen and McMahon, 2019). We run 1000 iterations before running the sample. Then, we twice run 20 samples from points in the chain thinned with a thinning interval of 50.

After several trials with a different number of topics (from 30 to 5), the optimal number of topics turns out to be 9. This number of topics is used to differentiate paragraphs discussing topics related to ‘general economic conditions’ and paragraphs discussing topics related to ‘inflation expectations’ and the ‘monetary policy decision’. A smaller number of topics do not allow this differentiation since topics mix with each other.

1.3.3 First LDA output: words per topic

Table 1 shows the first output of LDA, i.e. the word-topic matrix. We display the first twelve words with the highest probability for each topic. Word 1 is the word or stem with the highest probability in that topic. Word 2 is the word with the second highest probability and so on. Most of the topics are easily understandable. We can divide the topics into two groups, those that include words related to ‘current economic conditions’ and those that include words related to ‘inflation’ and the ‘monetary policy decision’. The aim of this division is to assign each paragraph of the minutes to one of the two previous groups of topics as in Hansen and McMahon (2016).

The first group of topics discusses ‘general economic conditions’ and comprises topics 2, 4, 6, 7 and 8. We assign a tag to each topic for mere interpretation. For instance, to topic 8 we assign the tag ‘industrial production’ since it comprises mainly stems related to industrial production such as ‘product’ with a probability of 0.081, also ‘industr’, ‘good’, etc. The topics related to ‘current economic conditions’ represent the first day of the COPOM meeting during which the various heads of department inform COPOM board members of the current economic and financial situation of Brazil and international markets.

The second group contains topics that are related to the ‘current situation of inflation and its expectations’ and the ‘monetary policy decision’. This group includes topics 0, 1, 3 and 5. Usually, the description of the ‘current state of inflation’ takes place on the first day of the meeting and discussions of ‘inflation expectations’ and the ‘monetary policy decision’ occur on the second day.

1.3.4 Second LDA output: topics per document

The second output of LDA is the distribution of probabilities of each topic per document represented by the term β_k. In our paper, we assign each paragraph to one of the two groups of topics. We determine that a paragraph is part of the ‘general economic conditions’ group of topics if the sum of the β_k probabilities of the topics of this group is higher than or equal to 0.555$\%$ since 5 topics over 9 belong to the ‘general economic conditions’ group of topics. However, if the value of the sum of β_k of the ‘general economic conditions’ group of topics is smaller than 0.555$\%$, the paragraph is assigned to the group of topics related to ‘inflation’ and the ‘monetary policy decision’.

For illustrative purposes, we estimate the distribution of topics in the minutes. Figure 1 shows the probability of topics related to the ‘current economic situation’ in the minutes and Figure 2 shows the probability of topics related to ‘inflation’ and the ‘monetary policy decision’. In the figures there are events due to a change in the format of the minutes or to a change in the governor of the Central Bank of Brazil. Two events have a considerable effect. The first significant event occurs in the 181st minute due to a change in the format of the minutes. The second event is in the 200th minute in 2016 where the format of the minutes is changed and the governor of the Central Bank of Brazil was replaced. Since the 200th minute, topics related to ‘general economic conditions’ and ‘inflation’ have a lower probability than topics related to the ‘monetary policy decision’.

Table 1: This table shows the first twelve words with the highest probability for each of the nine topics of the LDA results. A tag is included for each topic to provide a better understanding of the topic. These tags do not influence the results.

Topic	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10	Word 11	Word 12
0. Inflation	price	twelv	chang	index	ipca	food	agricultur	accumul	di	compar	regul	reflect
	0.164	0.054	0.039	0.033	0.021	0.021	0.02	0.019	0.016	0.015	0.015	0.014
1. Inflation / COPOM	inflat	expect	core	measur	copom	last	futur	pressur	short	monetari	smooth	mean
	0.161	0.057	0.031	0.029	0.019	0.017	0.015	0.015	0.014	0.014	0.013	0.012
2. Economic activity	economi	econom	market	intern	activ	remain	recoveri	global	growth	despit	financi	continu
	0.042	0.024	0.024	0.022	0.018	0.017	0.017	0.016	0.015	0.015	0.014	0.013
3. COPOM meeting	rate	project	meet	scenario	consid	copom	interest	target	exchang	market	selic	inflat
	0.104	0.044	0.038	0.036	0.036	0.032	0.031	0.025	0.024	0.019	0.019	0.019
4. Trade / credit operations	billion	total	credit	oper	reach	averag	period	export	trade	matur	day	respect
	0.081	0.042	0.041	0.039	0.035	0.032	0.025	0.025	0.024	0.02	0.019	0.018
5. COPOM meeting	monetari	polici	committe	will	risk	demand	copom	effect	factor	econom	process	time
	0.034	0.029	0.025	0.022	0.018	0.017	0.017	0.015	0.014	0.013	0.011	0.01
6. Sales retails	quarter	sale	decreas	retail	accord	adjust	end	survey	index	data	growth	confid
	0.056	0.048	0.045	0.026	0.025	0.024	0.023	0.021	0.021	0.02	0.019	0.018
7. Employment	rate	employ	compar	indic	sector	real	accord	record	labor	reach	thousand	result
	0.029	0.027	0.027	0.026	0.025	0.025	0.023	0.021	0.018	0.017	0.017	0.017
8. Industrial production	product	industri	good	capit	adjust	consum	season	accord	durabl	manufactur	expans	decreas
	0.081	0.073	0.07	0.03	0.03	0.03	0.026	0.02	0.019	0.017	0.016	0.015

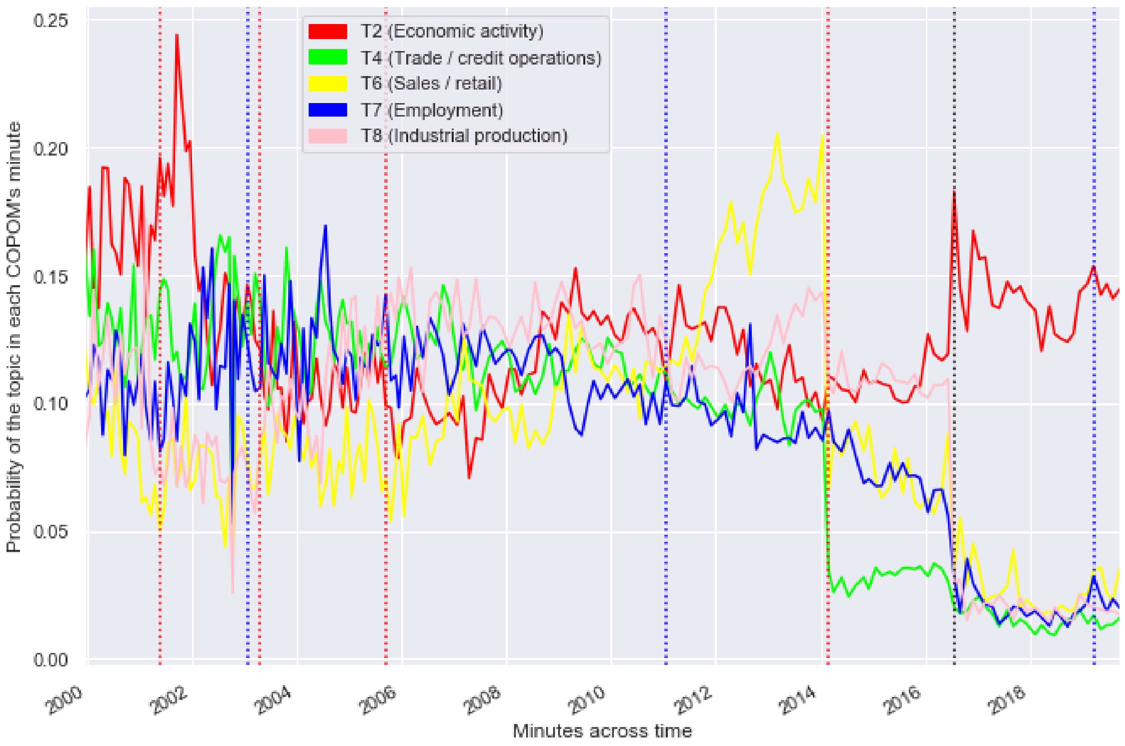


Figure 1: Weights of topics 2, 4, 6, 7 and 8 in the minutes from December 1999 to 2019. Notes: The bold lines are the probabilities of each topic in each set of COPOM minutes. The dotted blue lines represent a change in the Governor of the Central Bank of Brazil. The dotted red lines represent a change in the format of the minutes. The dotted black lines indicate a change in the format of the minutes and of the governor of the Central Bank of Brazil.

1.4 Tone Analysis: Estimation of Uncertainty and Topic-Uncertainty Indices

Our next step is to determine the degree of uncertainty in each of the minutes. To measure the degree of uncertainty, we apply the Skip Gram model and K-Means following Soto (2019) to construct a list of words related to ‘uncertain’, ‘uncertainty’, ‘uncertainties’ and ‘fears’. We count the number of times that words from this ‘uncertainty’ list appear in each set of minutes compared to the total number of words in each set. Following the same procedure, we create two topic-uncertainty indices. First, we build an uncertainty index for the paragraphs more likely to contain topics related to the ‘current state of the economy’. Second, we construct an uncertainty index for the paragraphs more likely to contain topics related to ‘inflation’ and ‘monetary policy decisions’.

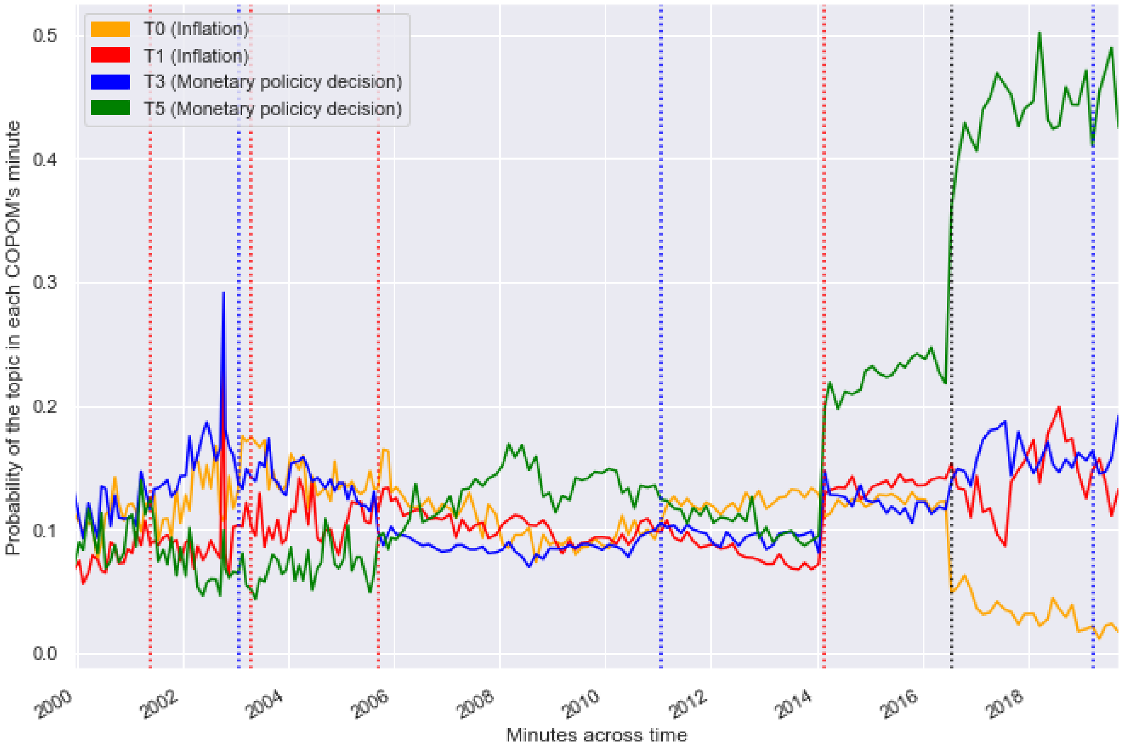


Figure 2: Weights of topics 0, 1, 3 and 5 in the minutes from December 1999 to 2019. Notes: The bold lines are the probabilities of each topic in each set of COPOM minutes. The dotted blue lines represent a change in the Governor of the Central Bank of Brazil. The dotted red lines represent a change in the format of the minutes. The dotted black lines indicate a change in the format of the minutes and of the governor of the Central Bank of Brazil.

1.4.1 Word Embeddings theory and the Skip-Gram model

The Word Embeddings model was introduced by Mikolov et al. (2013a). Word Embeddings are continuous vector representations of words with syntactical and semantic similarities between words in a Euclidean Space, decreasing the size of the text. The main idea of Word Embeddings is that we obtain a lot of meaning from a word by its context, i.e. the words around it or where it is embedded. For instance, consider the following documents:

Document 1: the economy experienced a period of growing **uncertainty** about the growth capacity

Document 2: the economy experienced a period of growing **concerns** about the growth capacity

The words ‘uncertainty’ and ‘concerns’ have similar meanings related to doubt and worry. In addition, the words ‘uncertainty’ and ‘concerns’ are preceded by ‘the economy experienced a period of growing' and followed by 'about the growth capacity'. The basic idea of Word Embeddings is to create a dense vector for each word type that is good at predicting the words that appear in their context and are also represented by a vector. In that case, we prefer a machine learning method that puts the vectors of words with similar meaning such as 'uncertainty' and 'concerns' in the same part of the vector space since they appear in the same context. To create the Word Embeddings in this way, we utilize the Skip-Gram model introduced by Mikolov et al. (2013a). The Skip-Gram model is a Neural Network machine learning method that tries to predict context words on the basis of a center word. This process is repeated for all the unique terms in the corpus, and for each term a vector of probabilities is created and placed in the vector space. For instance, uncertainty is the input or center word in document 1. The rest of the words are the output or context words.

$$\underbrace{\text{economy experienced growing}}_{\text{Output}}\ \underbrace{\textit{uncertainty}}_{\text{Input}}\ \underbrace{\text{about the growth capacity}}_{\text{Output}}$$

In the previous example, the Skip-Gram model provides the probability distribution of each of the context words based on the word uncertainty, which is the center word. For instance, P(growing | uncertainty) or P(about | uncertainty). For each word ($t = 1, ..., T$), the number of words in the context is given by the size of the window, m, that determines the number of context words before and after each center word. A window size of five means we estimate the probabilities of the five output words previous to the input word and the five output words following the input word.

The objective function consists in maximizing the probability of any context word given the current word as in Equation (6):

$$J(\Phi) = \prod_{t=1}^{T} \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(w_{t+j}/w_t; \Phi), \tag{6}$$

where, the term Φ is a representation of all the variables that have to be optimized. The term w_t represents the center or input word where t indicates the position in the text. The term w_{t+j} is the context word j of the center word w_t. For computational ease, the Skip-Gram model uses the negative log likelihood transformation of the objective function, aka the loss function, shown by the following equation:

$$J(\Phi) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} logP(w_{t+j}/w_t), \tag{7}$$

here, $P(w_{t+j}/w_t)$ is the probability of predicting an output word $t + j$ based on the input word t. The conditional probability $P(w_{t+j}/w_t)$ can be expressed in a simpler way by applying the softmax function as in the following equation:

$$P(0/I) = \frac{exp(u_O^T v_I)}{\sum_{w=1}^{V} exp(u_w^T v_I)}, \tag{8}$$

where, the term O is the output or context word and I is the input or center word. Moreover, v_I and u_O are 'input' and 'output' vectors respectively indexed by 'I' and 'O'. The dot product is equal to the multiplication of the vector $u_T.v = uv = \sum_{i=i}^{n} v_i$, that gives the probability of predicting the context word depending on the center word. We apply the softmax function to the dot product for two reasons. First, the exponential of the dot product makes the values higher than 0. Second, the denominator of Equation (8) forces the values to be between 0 and 1. Equation (8) is similar to the multinomial probability of the logit model. Moreover, each word has two vector representations, a first vector representation as the center or input word and a second vector representation as the output or context word. These two vector representations of the same word do not coincide in the model.

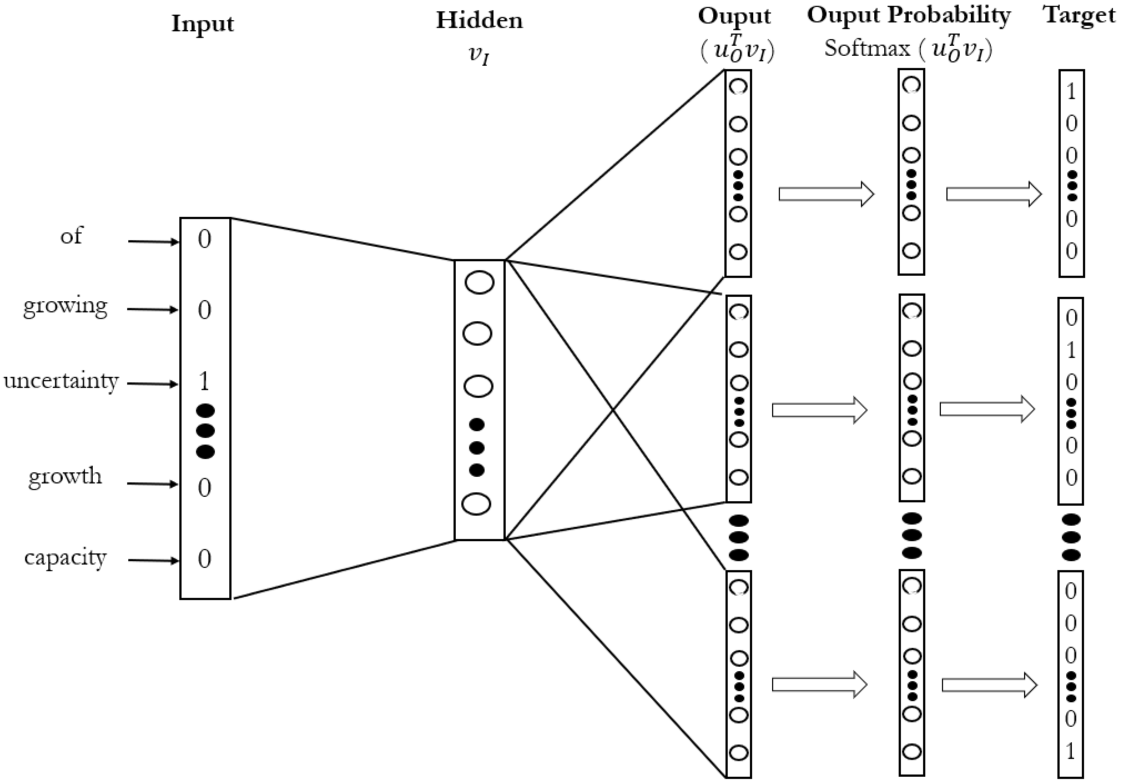


Figure 3: Representation of the Skip-Gram model.

Figure 3 shows the Skip-Gram model structure in detail and the optimization process for only a center word, i.e. 'uncertainty' in the figure. This center word is represented by one-hot vector of length V depicted in the input layer. The one-hot vector assigns value 1 to the center word and 0 to the other terms. Then, the input layer is multiplied by an

H-by-V matrix U, where each column corresponds to each center word in the text. The product of both matrices is v_I which is known as the Hidden layer of dimension H-by-1. The dimension of the Hidden layer can be established by researchers. To obtain the Output layer, multiply the Hidden layer by the Output word representation matrix L of dimension V-by-H in which each row represents a context word. Hence the different vectors of the Output layer are obtained by multiplying the Hidden layer by the rows of matrix L that correspond to the various context words. We then apply the softmax function to the vectors of the Output layer to obtain probabilities between 0 and 1 as described above in relation to the dot product. The vectors of the Output Probability layer describe the probability of each context word appearing given a certain input word. For instance, we expect to obtain the highest probability in the first term in the first vector of the Output Probability layer. Since the output word we are trying to predict in the first vector is the first term of the Target layer with value 1 (corresponding to the term 'of' in our example). The steps from one layer to another are summarized in the following equations (Soto, 2021):

$$\text{Input} = x_{w_t}$$

$$\text{Hidden layer (Word Embedding)} = v_I = Ux_{w_t}$$

$$\text{Output} = x_O = Lv_I = [u_1^T v_I \;\; u_2^T v_I \;\; ... \;\; u_w^T v_I]$$

$$\text{Output Probability} = \text{softmax}(u_w^T v_I)$$

The Matrix U is the important element of the Skip-Gram model since the column of word w_t in matrix U represents the Word Embeddings of word w_t in R^H. This column is the one used to identify semantic and syntactical differences. Moreover, L could be represented as a Word Embedding too, in which case the rows would be the Word Embeddings but they are not used here (Soto, 2021).

We define the set of all parameters in the model in terms of a vector Φ. This vector Φ comprises all the vectors of all the unique terms V as input terms and context words. To optimize the parameters of vector Φ, we minimize the log-likelihood function represented in Equation (7) in order to maximize the Output Probability for each context word. Equation (7) can also be expressed as:

$$J(\Phi) = -log \frac{exp(u_O^T v_I)}{\sum_{w=1}^{V} exp(u_w^T v_I)}, \tag{9}$$

or

$$J(\Phi) = -(u_O^T v_I) + log \sum_{w=1}^{V} \exp(u_w^T v_I). \tag{10}$$

Initially, the word vectors are randomly computed. To estimate the optimal parameter of U and L, we apply a gradient descendent to the entire corpus for all the windows. The model adjusts the parameters through backpropagation so the Output Probability and Target are the lowest. The gradient values of U and L are as follows:

$$L^{new} = L^{old} - \alpha \frac{\partial}{\partial L} J(\Phi), \tag{11}$$

$$U^{new} = U^{old} - \alpha \frac{\partial}{\partial U} J(\Phi). \tag{12}$$

1.4.2 K-Means Clustering

K-Means Clustering is a technique that attempts to link observations that are close to each other in the input space. In this paper, we use K-Means to cluster the Word Embeddings, which are vector representations contructed with the Skip-Gram model, into C disjoint groups or clusters. We then identify the cluster that encompass the words 'uncertain', 'uncertainties', 'uncertainty' and 'fears' as in Soto (2021).

K-Means is a centroid-base algorithm. This algorithm aims to find the cluster assignments of all m observations to C clusters that minimize the within cluster distances between each point x_i and its cluster centre μ_c (Chakraborty and Joseph, 2017). The within cluster distances is normally measured by the Euclidean distance. The corresponding cost function is:

$$ERR(X, C) = \frac{1}{m} \sum_{c=1}^{C} \sum_{x_i \in C_c} || \, x_i - \mu_c \, ||^2. \tag{13}$$

Here, the sum of squares is normalized by the number of observations, as required to compare clusters of different sizes. In order to establish a fixed number of clusters C, we alternate steps of cluster assignment and centroid shifting. During clustering assignment, we assign each observation x_i to its closest centroid C_i. In centroid shifting, we compute the new position for each centroid. Moreover, highly correlated features must be avoided since they might cause spurious clustering. Finally, the number of clusters needs to be decided. Several evaluation methods can be used including the 'silhouette coefficient' and 'elbow-method' (Chakraborty and Joseph, 2017).

1.4.3 Estimation of Word Embeddings

The Skip-Gram model is applied to the same corpus of minutes of the Central Bank of Brazil. Nonetheless, there are some differences in the preprocessing of the corpus. First, the words in the Skip-Gram corpus are not stemmed because of the risk of losing information due to the semantic differences between words. Second, we identify bigrams or pairs of words that appear with a frequency higher than 10. The bigrams identify couples of words that represent the same term or idea. Finally, the text in the Skip-Gram model is a whole unique document instead of different documents comprising paragraphs as in LDA.

We attempt different combinations of the Hidden layer and the window size in the Skip-Gram model.[3] We select parameters that provide logical results. In particular, we estimate Skip-Gram with a Hidden layer (H) of 200 and a context window size (m) of 10. Furthermore, 140 clusters are selected for the application of K-Means.

After applying the Skip-Gram and K-Means models, we select all the words in the same clusters as 'uncertainty', 'uncertain', 'uncertainties' and 'fears' to construct a dictionary or list of words related to uncertainty. We assume that the words in the same clusters share a similar semantic meaning. The words in the same clusters as the words 'uncertainty', 'uncertain', 'uncertainties' and 'fears' are shown in Table 1. The list of 'uncertainty' words includes words such as 'unstable', 'ambiguous_influence', 'turmoil' and 'risks'. Other words describe critical events such as 'earthquake', 'brexit', 'mortgage_crisis' or 'war'. Besides, there are terms related to oil-producing countries that might be in trouble as 'iraq', 'opec' or 'venezuela'. Some words are related to the business cycle such as 'widespread_disinflation', 'devaluation' or 'dollar_appreciation'. Moreover, our results might not fully show the potential of the Skip-Gram model since the data available for the minutes of Brazil are limited compared to the size of current databases as in the case of social media.

1.4.4 Estimation of uncertainty and topic-uncertainty indices

An uncertainty index for the minutes of the Central Bank of Brazil is constructed by assigning an uncertainty score to each set of minutes. As we show in Equation (14), the uncertainty score of each set of minutes is computed as the number of times any word in our 'uncertainty' list appears divided by the total number of words in that set of minutes. We standardize the uncertainty score by multiplying it by 100 and dividing it by the mean score of all the minutes used to construct the uncertainty index as shown in Equation (15).

[3] We implement the Skip-Gram model with the Gensim library (Word2Vec) in Python.

Table 2: List of words in the same cluster as the words 'uncertain', 'uncertainty', 'uncertainties' and 'fears'.

abrupt, absence, abundant, abundant_global, actually, adjust, adverse, affirm, africa, alternative, america, american, ample, another_concern, apparently_little, asian, asset, assign_low, assume, asymmetric, attack, attacks, band, benign_inflationary, brazilian_assets, brexit, capital_flows, causing, chances, chinese_economy, clear_identification, closely_monitored, committee_understands, commodities, commodity, complex, complexity, complexity_surrounding, comprise, concerns, concretization, consequences, consequent, considerable_degree, constitute, constraints, contaminate, could, could_affect, decades, deficits, degree, deleverage, depends, depreciating, derive, derived, deriving, despite_identifying, deteriorate, deterioration, devaluation, developed_countries, deviates, diagnosis, dollar_appreciation, dollar_depreciation, earlier, earthquake, ease, eased, eastern, economic_blocks, elections, electoral_process, emerging, emerging_countries, enable_natural, environment, episodes, equity_markets, european_countries, evaluates, existence, exporting_countries, extent_reflect, external_environment, external_financing, extraordinary, extreme_events, faced, facts, fashion, favoring, fear, fears, financial_markets, financing_conditions, fragility, fueled, generate, geopolitical, geopolitical_tensions, global_outlook, gradual_normalization, handling, heating, heightened, heterogeneous, highly_volatile, identifies, imply, impose, impose_adjustments, incidentally, industrialized_countries, industrialized_economies, inflationary, initially_localized, initiatives_taken, instability, international, international_financial, iraq, justified, latent, latin_america, less_likely, likelihood, localized, low_probability, major, major_advanced, major_economies, manifest, markets, markets_quotations, mechanisms, middle_east, midst, might, minor, mitigate, mortgage_crisis, movements, moves, nevertheless, news, normalization, north, northern_hemisphere, notably, nuclear, observes, ongoing_deleveraging, opec, originally, originated, particularly, persists, pessimism, political, pondered, pose, positive_spillovers, possible, potentially, predominantly, premature, pressuring, prevalence, pricing, problems, producing_countries, promptly_converges, prospectively, provoked, prudent, quotations_remains, reacting, reaction, reactions, realignments, reassessment, recently, recurrent_geopolitical, remain_tied, remains_complex, repercussions, risk, risk_appetite, risk_aversion, risks, risky_assets, satisfactory, scarcity, selected_commodities, shortage, show_resistance, significant_deterioration, since_mid, speculative, spillovers, stem, strongly_impacted, subdued, subsequent_years, substantial_share, suffer, surround, surrounded, surrounding, swings, tension, tensions, tensions_despite, tightened, towards_normality, traditionally, transition, transitory, turmoil, uncertain, uncertainties, uncertainty, uncertainty_concerning, unstable, valuation, venezuela, volatility, volatility_affecting, war, wave, weaken, wealth, widening, widespread_disinflation, winter, world, world_economy, worldwide, worries, would, yen.

$$S_s = U_s / N_s, \tag{14}$$

$$F_s = 100 \frac{S_s}{\frac{1}{M} \sum_{m=1}^{M} S_m}, \tag{15}$$

where, the term U_s is the number of uncertainty words in minute s, and N_s is the total number of words in that set of minutes. Furthermore, S_s and F_s are the uncertainty score and the uncertainty index of minute s, respectively. The denominator of Equation (15) is the mean of all the values of the uncertainty score.

Figure 4 shows the evolution of the uncertainty index. We compare it with the Economic Policy Uncertainty (EPU) index for Brazil created by Baker, Bloom, and Davis (2016) from the Brazilian newspaper 'Folha de Sao Paulo'. The Brazilian EPU index consists in counting the number of articles that contain at least one word in each of three groups of words pre-established by the researches. The first group of words contains words related to policy terms such as 'regulation' or 'deficit', and the second group of words comprises the words 'uncertain' and 'uncertainty'. The third group of words comprises the words 'economic' and 'economy'. We standardize the EPU index following Equation (15) so the mean of the EPU index is 100 for our sample. Figure 4 shows that the uncertainty index follows a similar pattern to the EPU index of Baker, Bloom, and Davis (2016). However, the index increases significantly in 2016 and the 200th minute, coinciding with the replacement of the governor of the Central Bank of Brazil and a change in the format of the minutes. However, the increase is captured by the index of Baker, Bloom, and Davis (2016) after 2014. During the years 2014 and 2016, Brazil suffered one of its worst economic crises in recent decades.

We construct two topic-uncertainty indices, creating the first topic-uncertainty index for the paragraphs more likely to include topics related to 'general economic conditions'. Another topic-uncertainty index is created for the paragraphs more likely to include topics related to 'inflation' and the 'monetary policy decision'. To build the two topic-uncertainty indices, we follow the same procedure as described for the general uncertainty index. With the two topic-uncertainty indices, we can identify the origin of uncertainty either in the 'general economic situation' paragraphs or the 'inflation' and 'monetary policy decision' paragraphs. Figure 5 shows the evolution of the two topic-uncertainty indices and we compare them again to the EPU index of Baker, Bloom, and Davis (2016) for Brazil. From 2000 until 2014, the 'inflation' and the 'monetary policy decision' topic-uncertainty index is higher for almost all the periods than the 'general economic conditions' topic-uncertainty index. In 2014, there was an economic crisis

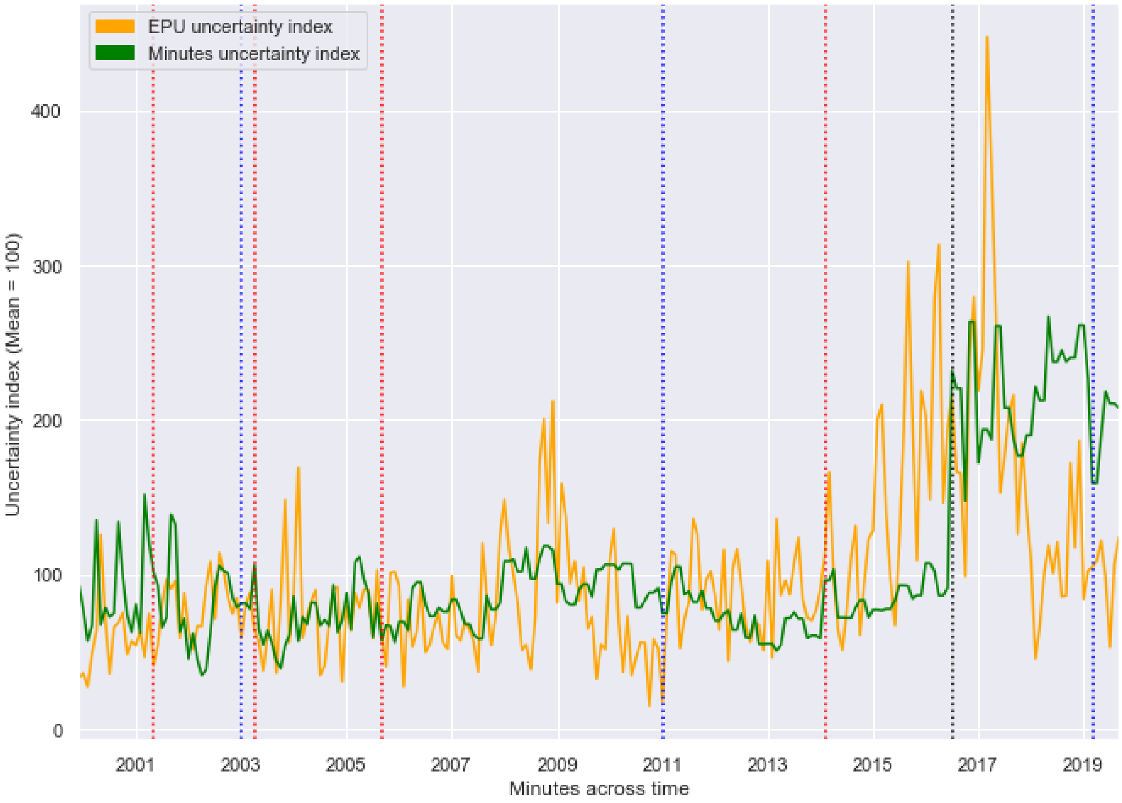


Figure 4: Minutes uncertainty index - December 1999 to 2019.

in Brazil, reflected by the fact that the 'general economic conditions' uncertainty index outscores the 'inflation' and 'monetary policy decision' topic-uncertainty index. Finally, again there was a considerable increase in both topic-uncertainty indices after the 200th minutes, especially in the 'general economic conditions' uncertainty index. Nonetheless, the number of paragraphs covering the 'general economic conditions' decreases drastically after the 200th meeting in 2016, leading to more volatility in this index, including values equal to zero. Therefore, our analysis discards the 'general economic conditions' topic-uncertainty index after the 200th minutes.

1.5 Structural VAR

The most similar paper to ours is Hansen and McMahon (2016) who investigate FOMC statements. With LDA and manually they identify the parts of FOMC statements that discuss 'current economic conditions' or the 'monetary policy decision'. For the part related to 'current economic conditions' they create a positive-negative index with words associated with expansion and recession in the dictionary list of Apel and Blix Grimaldi (2012). For the 'monetary policy decision' parts of FOMC statements, they estimate a topic-uncertainty index by counting the relative frequency of the words in the uncertainty dictionary of Loughran and McDonald (2011). Later, they estimate a Factor-Augmented

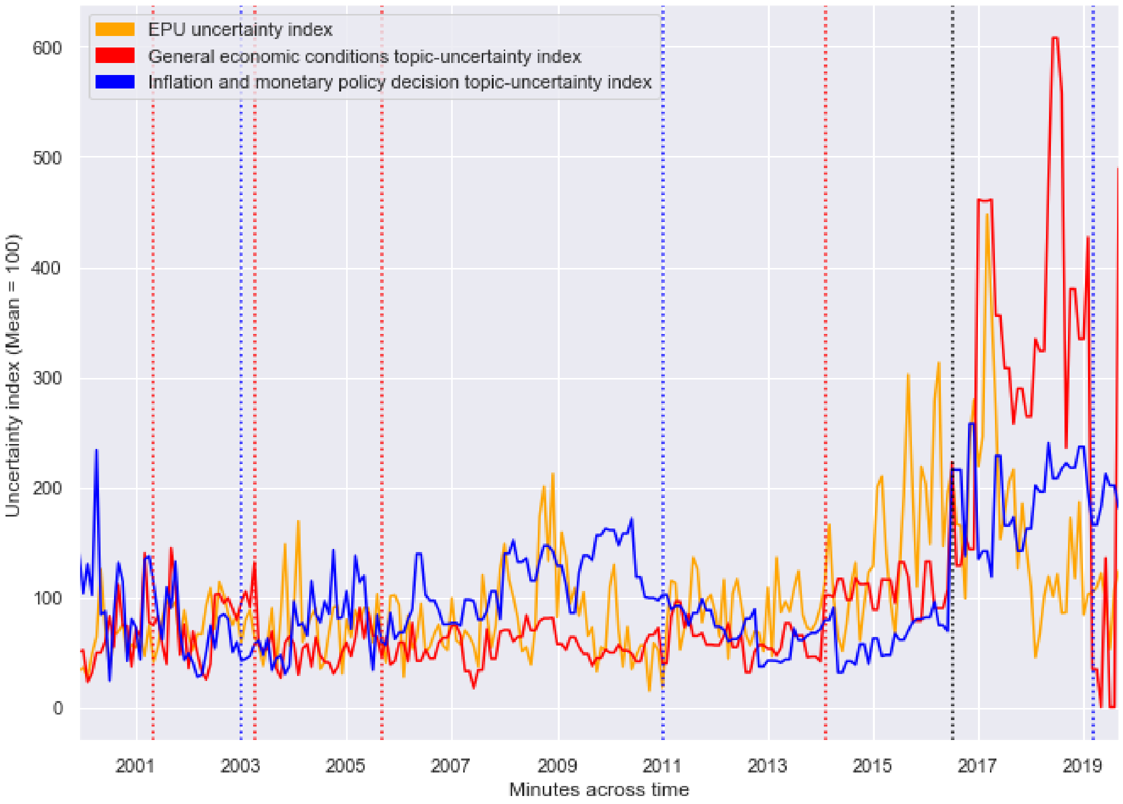


Figure 5: Topic-uncertainty indices - December 1999 to 2019.

Vector Autoregression (FAVAR) to investigate the effect of the text measures in the market and real variables. They observe that the effect of communications' shocks in 'current economic conditions' in market and real variables is lower than the effect of communications' shocks in the 'monetary policy decision' part of the FOMC statements.

We investigate the effect of the uncertainty index and the two topic-uncertainty indices in the Brazilian economy. For this purpose, we compute a Structural Vector Autoregression (SVAR) model:

$$B_0 Y_t = \sum_{i=1}^{p} B_i Y_{t-i} + \omega_t, \tag{16}$$

where, ω_t refers to a structural innovation or structural shock, but also represents the mean zero serially uncorrelated error term. The term Y_t is a K-dimensional time series $t = 1, ..., T$. The term Y_t is approximated by a vector autoregression of finite order p. The matrix B_0 represents the simultaneous associations of variables in the model (Kilian and Lütkepohl; 2017). The model can be expressed in reduced form as:

$$Y_t = \underbrace{B_0^{-1}B_1}_{A_1}Y_{t-1} + ... + \underbrace{B_0^{-1}B_p}_{A_p}Y_{t-p} + \underbrace{B_0^{-1}\omega_t}_{u_t}, \tag{17}$$

where, the new error vector, u_t, is a linear transformation of the old error vector, ω_t. Once we estimate the reduced form, the problem is to recover the structural representation of the VAR model which is represented by Equation (16). In particular, the main issue is how to obtain B_0 since it can estimate ω_t due to $\omega_t = u_t B_0$, and also estimate B_i since $B_i = A_i B_0$ for $i = 1, ..., p$. To obtain ω_t, we 'orthogonalize' the reduced form error which consists in making the errors mutually uncorrelated. This can be achieved by defining the lower-triangular KxK matrix P with positive main diagonal such as $PP' = \sum_u$, where $\sum_u$ is the variance-covariance matrix of u_t. We know that the matrix P is the lower-triangular Cholesky decomposition of $\sum_u^2$. Therefore, one of the solutions to obtain ω_t is the condition $\sum_u = B_0^{-1}B_0^{-1'}$ in which $B_0^{-1} = P$ (Kilian and Lütkepohl; 2017).

In our paper, the vector $Y_t = [\Delta F_t, \Delta E_t, \Delta\pi_t, \Delta P_t, \Delta C_t]$ where ΔE_t stands for the difference in the Real broad effective exchange rate for Brazil, $\Delta\pi_t$ indicates the difference in the consumer price index in Brazil, ΔP_t is the difference in total industrial output in Brazil, and ΔC_t is the difference in total retail trade. ΔF_t stands for the difference in the range of the uncertainty indices. For clarification, differences indicate first differences of time series, taken over subsequent time instants. For the months with no meetings, we assume the value of the uncertainty index of the previous set of minutes Moreover, all the macroeconomic variables are extracted with monthly frequency from the Federal Reserve Bank of St. Louis. All variables are differentiated to overcome the non-stationary problem in light of the augmented Dickey-Fuller test indicating I(1).

The optimal number of lags is in line with Akaike Information Criteria (AIC), the Bayesian Information Criterion (SBIC), and the Hannan and Quinn Information Criterion (HQIC). The SVAR model complies with the stability condition since all roots of the characteristic polynomial are outside the unit circle. The identification of structural shock is obtained by appealing to the usually estimated Cholesky decomposition put forward by Sims (1980). The Cholesky decomposition involves the so-called recursiveness assumption, an economic assumption about the timing of the reaction to shocks in the variables. In other words, the recursiveness assumption imposes order between the variables. In our paper, the uncertainty index (ΔF_t) simultaneously affects the other variables, but is not affected by the remainder as in Bloom (2009) and Nodari (2014). Hence, ΔE_t simultaneously affects $\Delta\pi_t, \Delta P_t$ and ΔC_t. $\Delta\pi_t$ has a simultaneous impact on ΔP_t and ΔC_t. Subsequently, it continues this way for the last two variables. We estimate the Structural VAR model for each of the uncertainty indices. First, we make two estimations with the full sample for the following two uncertainty indices: 1) the minutes uncertainty index; 2) the 'inflation' and 'the monetary policy decision' topic-uncertainty index. Then, we restrict the sample until the 199th minutes in June 2016 due to a lack of data for the 'general economic conditions' topic-uncertainty index. We again estimate Structural VAR with this reduced sample for all the uncertainty indices constructed from the minutes: 3) the general uncertainty index for the minutes; 4) the 'inflation' and 'the monetary policy decision' topic-uncertainty index; 5) the 'general economic conditions' topic-uncertainty index.

1.6 Results

Figures A.1 and A.2 show the results of the impulse response analysis for the whole sample from 2000 to July 2019. Figure A.1 demonstrates the effects of an increase in a unit shock in the minutes uncertainty index in four Brazilian macroeconomic variables. A rise in one standard shock in the uncertainty index of the minutes depreciates the exchange rate by almost 0.3%. During uncertain times, the Brazilian Real might depreciate to restore the competitiveness of the Brazilian economy. Moreover, an increase in the uncertainty index slightly reduces inflation. However, in two periods after the shock it becomes positive. Lastly, industrial production and the retail trade both decrease by around 0.16% with a unit shock in the general uncertainty index. The results of industrial production and the retail trade are similar to the results of Costa-Filho (2014) after a unit in the uncertainty index. The results of Godeiro and de Oliveira-Lima (2017) also suggest the same negative relationship between macroeconomic uncertainty and industrial production in Brazil. In Figure A.2, the results of the 'inflation' and 'monetary policy decision' topic-uncertainty index are similar to the results of the uncertainty index. The effect on industrial production lasts longer for the 'inflation' and 'monetary policy decision' topic-uncertainty index.

Figures A.3 to A.5 repeat the analysis for all the uncertainty indices constructed from the minutes from 2000 to June 2016. Figure A.3 shows the impulse response functions of the uncertainty index. The results are similar to those computed for the whole sample, as shown in Figure A.1. However, in the reduced sample industrial production decreases drastically in the period following the shock rather than in the same period, as shown in Figure A.1. Figure A.4 shows the results of the impulse response functions for the 'inflation' and 'monetary policy decision' topic-uncertainty index with the reduced sample. Figure A.5 shows the 'general economic conditions' topic-uncertainty index with the reduced sample. A unit shock in the 'inflation' and 'monetary policy decision' topic-uncertainty index leads to a larger fall in the exchange rate than in the results of the 'general economic conditions' topic-uncertainty index. This might be explained by the large depreciation of the Brazilian Real after the world financial crisis of 2008 during the 'world currency war'. This depreciation attempted to make Brazilian exports more competitive. In the five years after the financial crisis of 2008, the 'inflation' and 'monetary policy decision' topic-uncertainty index is relatively high. This might be a proxy of the complex international financial situation facing COPOM board members. The 'general economic conditions' topic-uncertainty index has a low value during the five years after the world's economic crisis of 2008, capturing the growth of the Brazilian economy in that period.

In Figure A.5, we observe that a unit shock in the 'general economic conditions' topic-uncertainty index has a positive impact on inflation. However, the impact of a unit shock in the 'inflation' and 'monetary policy decision' topic-uncertainty index has a negative impact on inflation. This might be explained by the fact that the 'general economic conditions' topic-uncertainty index is higher than the 'inflation' and 'monetary policy uncertainty' topic-uncertainty index during periods of higher inflation and tougher economic conditions (beginning of the decade of 2000s and from 2014 to 2016). It might also be related to the fact that COPOM members express more uncertain views in the paragraphs related to 'inflation' and 'monetary policy decision' during the period after the financial crisis of 2008 characterized by lower inflation.

In addition, the 'inflation' and 'monetary policy decision' topic-uncertainty index has a higher negative effect on industrial production than the 'general economic conditions' topic-uncertainty index. This might be explained by the sharp fall in industrial production after the financial crisis of 2008 which may be correlated with an increase in the 'inflation' and 'monetary policy decision' topic-uncertainty index in the same period. Finally, we observe similar results for a unit shock in retail for both topic-uncertainty indices.

We check the validity of our results by estimating the Structural VAR model with an external uncertainty index such as the EPU index for Brazil. Figure A.6 shows the results of the impulse response analysis for the standardized EPU uncertainty index for the whole sample. The results are similar to those of the uncertainty index of the minutes. Nonetheless, an increase in one standard shock of the EPU index leads to a fall in the exchange rate three times higher than is the case for the uncertainty index of the minutes (Figure A.1). Figure A.7 shows results of the impulse response analysis for the standardized EPU uncertainty index for the period 2000 - June 2016. Again, these results are similar to those of the uncertainty index of the minutes, as shown in Figure A.3. In Figure A.7, in the same period, an increase of one-unit shock in the EPU index has a positive effect on retail and later drop to negative values in the periods after the shock.

1.7 Conclusion

This paper investigates the relationship between the views expressed in the minutes of the meetings of the Monetary Policy Committee (COPOM) of the Central Bank of Brazil and the real economy. For this purpose, we suggest simple measures of communication to identify the topic and tone of the minutes of the Central Bank of Brazil. First, topic or content analysis enables us to understand what the minutes are talking about. Here, we use Latent Dirichlet Allocation to deduce the content or topics of each paragraph of our sample. We identify two main groups of topics, the 'current economic conditions' topics and the 'inflation' and 'monetary policy decision' topics. By tone analysis, we compute the degree of uncertainty in each paragraph of the minutes. We use the Skip-Gram and the K-means algorithms to create a list of words with similar meaning to 'uncertain', 'uncertainty', 'uncertainties' and 'fears' comprising our dictionary of words related to 'uncertainty'. We then compute the relative frequency of the words from the 'uncertainty' dictionary to construct an uncertainty index for the minutes of the Central Bank of Brazil and combine both topic and tone text measures to build two topic-uncertainty indices. The first topic-uncertainty index is constructed from paragraphs that are more likely to include topics related to 'general economic conditions'. We create a second topic-uncertainty index from the paragraphs that are more likely to include topics related to the 'inflation situation and expectations' and the 'monetary policy decision'.

Finally, with a Structural VAR model we estimate the effect on the real economy corresponding to an increase in the uncertainty index of the minutes and the two topic-uncertainty indices. Our results show that higher uncertainty in the minutes of the COPOM leads to a fall in the exchange rate, industrial production, inflation, and retail sales. We also show the differing impacts on the 'general economic conditions' topic-uncertainty and the 'inflation' and 'monetary policy decision' uncertainty index in relation to macroeconomic variables such as the exchange rate, inflation and industrial production.

Future research could further investigate the communications of the Central Bank of Brazil such as the monetary policy statements or study the effect in the financial markets. Future research could also use alternative unsupervised machine learning methods such as Dynamic Topic Modelling.

Bibliography

[1] Apel, M., and Blix Grimaldi, M. (2012). The information content of central bank minutes. *Working Paper Series, Sveriges Riksbank (Central Bank of Sweden)*, 261, Apr.

[2] Baker, S. R., Bloom, N., and Davis, S. J. (2016). Measuring economic policy uncertainty. *The Quarterly Journal of Economics*, 131.4: 1593-1636.

[3] Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning research*, 3, 993-1022.

[4] Bloom, N. (2009). The impact of uncertainty shocks. *Econometrica* 77 (3), 623–685.

[5] Cabral, R., and Guimaraes, B. (2015). O comunicado do banco central. *Revista Brasileira de Economia*, 69, 287-301.

[6] Chague, F., De-Losso, R., Giovannetti, B., and Manoel, P. (2015). Central bank communication affects the term-structure of interest rates. *Revista Brasileira de Economia*, 69, 147-162.

[7] Chakraborty, C., and Joseph, A. (2017). Machine learning at central banks. *Bank of England Staff Working Paper*, 674.

[8] Corrado, D. (2013). Brasile Senza Maschere. Politica, Economia e Società Fuori dai Luoghi Comuni. *Università Bocconi Editore*.

[9] Costa-Filho, A. E., and Rocha, F. (2010). Como o mercado de juros futuros reage à comunicação do banco central?. *Economia Aplicada*, 14, 265-292.

[10] Costa-Filho, A. E. D. (2014). Incerteza e atividade econômica no Brasil. *Economia Aplicada*, 18, 421-453.

[11] Garcia-Herrero, A., Girardin, E., and Dos Santos, E. (2017). Follow what I do, and also what I say: monetary policy impact on Brazil's financial markets. *Economia*, 17, 65-92.

[12] Garcia-Uribe, S. (2018). The effects of tax changes on economic activity: a narrative approach to frequent anticipations. *Documento de Trabajo del Banco de España*, 1828.

[13] Griffiths, T. L., and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101, 5228-5235.

[14] Godeiro, L. L., and de Oliveira-Lima, L. R. R. (2017). Measuring macroeconomic uncertainty to Brasil. *Economia Aplicada*, 21, 311-334.

[15] Hansen, S., and McMahon, M. (2016). Shocking language: understanding the macroeconomic effects of central bank communication. *Journal of International Economics*, 99, S114-S133.

[16] Hansen, S., McMahon, M.,and Prat, A. (2017). Transparency and deliberation within the FOMC: a computational linguistics approach. *The Quarterly Journal of Economics*, 133, 801-870.

[17] Hansen, S., McMahon, M., and Tong, M. (2019). The long-run information effect of central bank communication. *Journal of Monetary Economics*, 108, 185-202.

[18] Hendry, S., and Madeley, A. (2010). Text mining and the information content of Bank of Canada communications. *Staff Working Paper of the Central Bank of Canada*, 31.

[19] Kilian, L., and Lütkepohl, H. (2017). Structural Vector Autoregressive Analysis. *Cambridge University Press.*

[20] Larsen, V. H., and Thorsrud, L. A. (2019). The value of news for economic developments. *Journal of Econometrics*, 210(1), 203-218.

[21] Loughran, T. and McDonald, B. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-Ks. *Journal of Finance*, 66:1, 35-6.

[22] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *Preprint Arxiv*, 1301, 3781.

[23] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pp. 3111-3119.

[24] Nodari, G. (2014). Financial regulation policy uncertainty and credit spreads in the US. *Journal of Macroeconomics*, 41, 122-132.

[25] Ortiz, A., Rodrigo, T., and Turina, J. (2017). How do the central banks talk?: a big data approach to Turkey. *BBVA Research Working Papers*, 17/24.

[26] Rosa, C., and Verga, G. (2007). On the consistency and effectiveness of central bank communication: evidence from the ECB. *European Journal of Political Economy*, 23, 146-175.

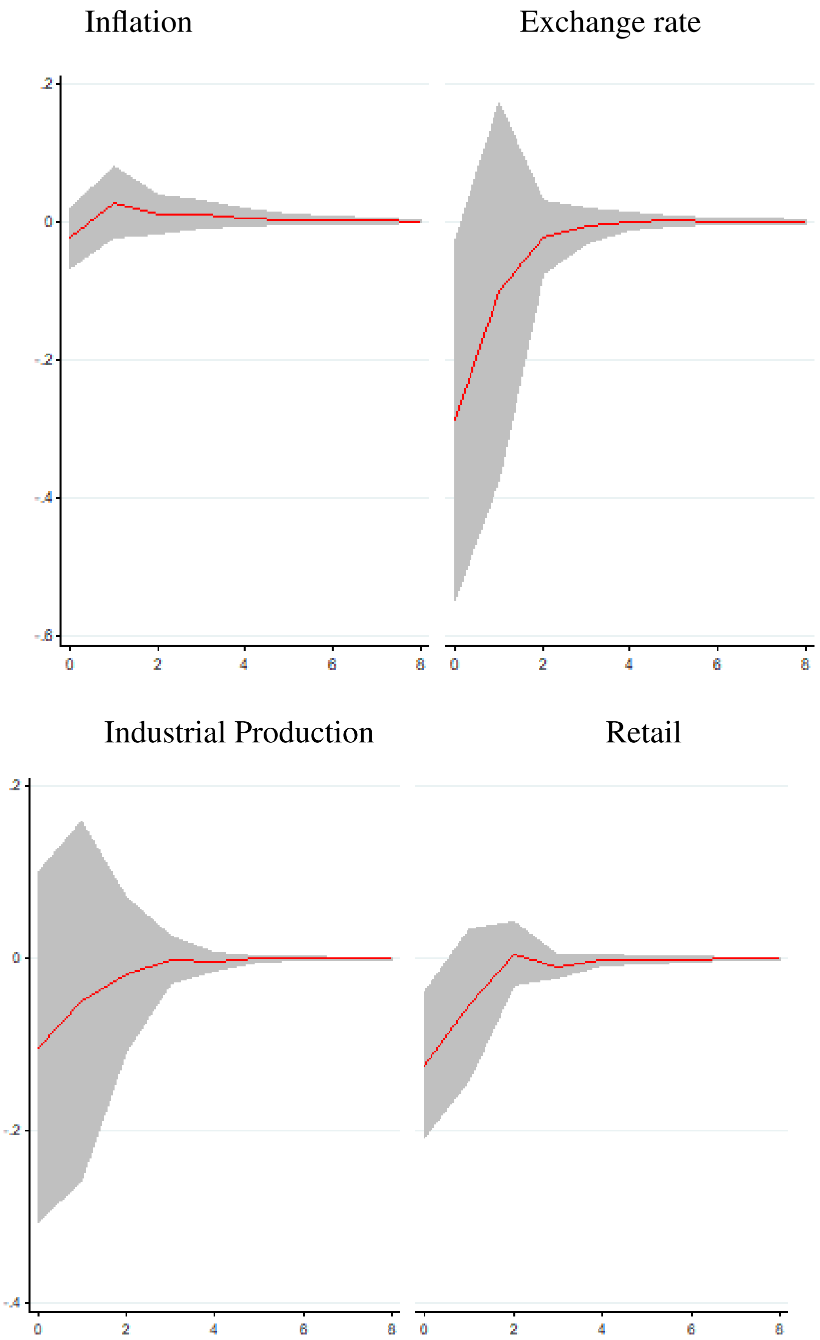
[27] Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, 1-48.

[28] Soto, P. E. (2021). Breaking the word bank: measurement and effects of bank level uncertainty. *Journal of Financial Services Research*, 59(1), 1-45.

[29] Tobback, E., Naudts, H., Daelemans, W., de Fortuny, E. J., and Martens, D. (2018). Belgian economic policy uncertainty index: improvement through text mining. *International Journal of Forecasting*, 34, 355-365.

[30] Zou, H., and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 301-320.

Appendix



Inflation

Exchange rate

Industrial Production

Retail

Figure A.1: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in the uncertainty index of the minutes of the COPOM from 2000 to July 2019. The gray area displays the 90% confidence intervals computed using bootstrapped standard errors (200 replications). The Y-axis is in % points of each of the four macroeconomic variable and the X-axis represents time in months (8 months).

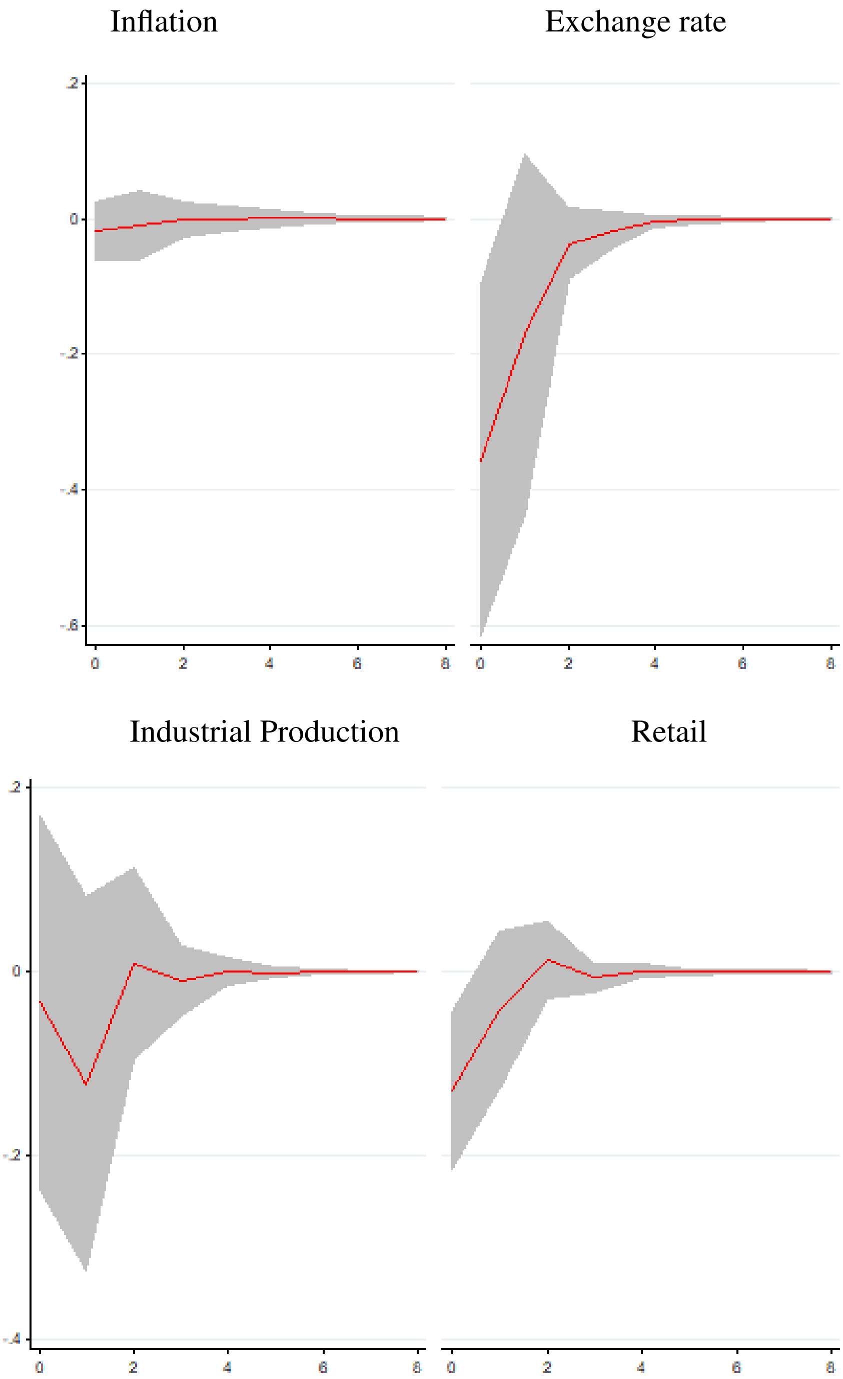


Figure A.2: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in the ‘inflation’ and ‘monetary policy decision’ topic-uncertainty index of the minutes of the COPOM from 2000 to July 2019. The gray area displays the 90% confidence intervals computed using bootstrapped standard errors (200 replications). The Y-axis is in % points of each of the four macroeconomic variable and the X-axis represents time in months (8 months).

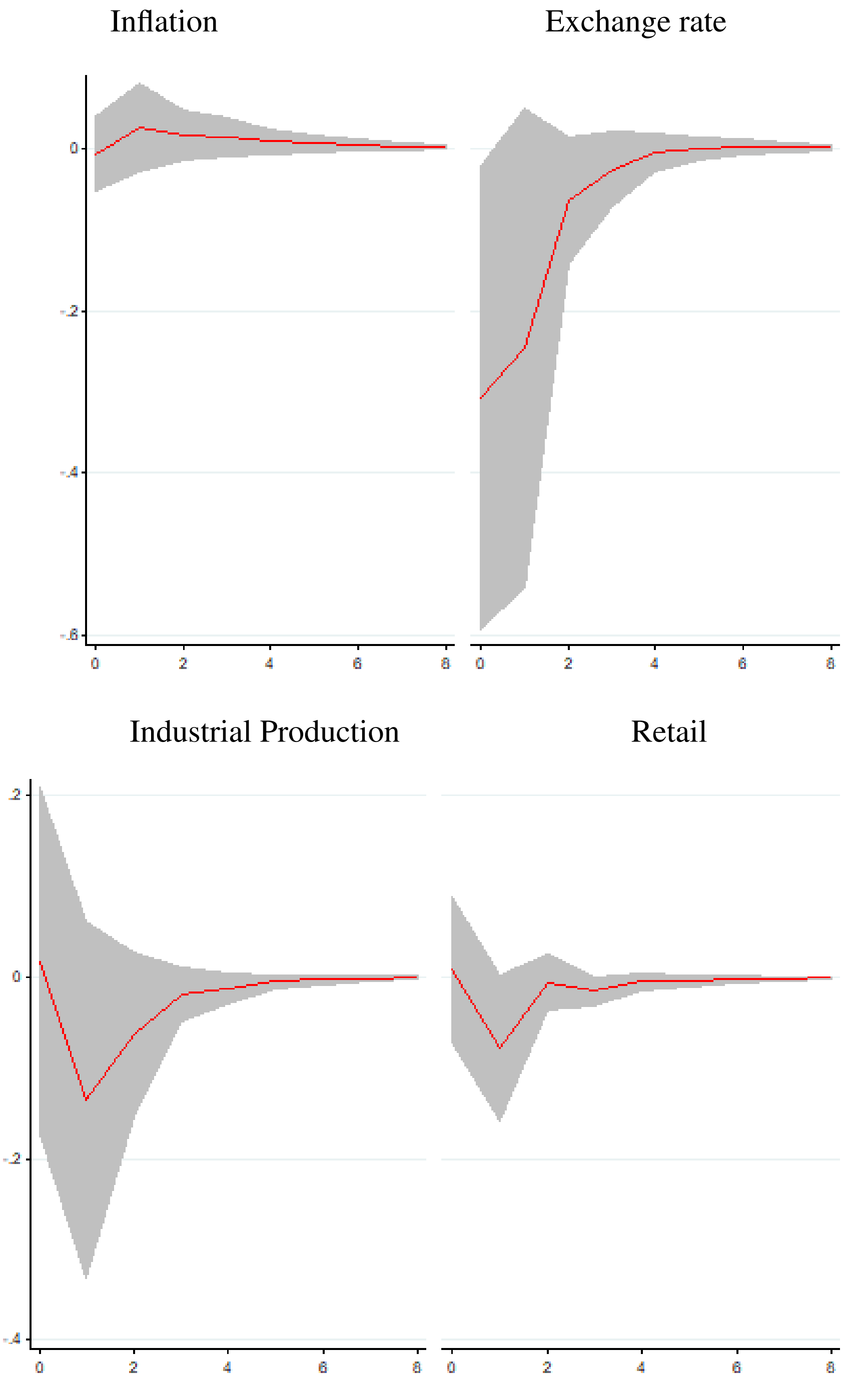


Figure A.3: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in the uncertainty index of the minutes of the COPOM from 2000 to June 2016. The gray area displays the 90% confidence intervals computed using bootstrapped standard errors (200 replications). The Y-axis is in % points change for each one of the four macroeconomic variables and the X-axis represents time in months (8 months).

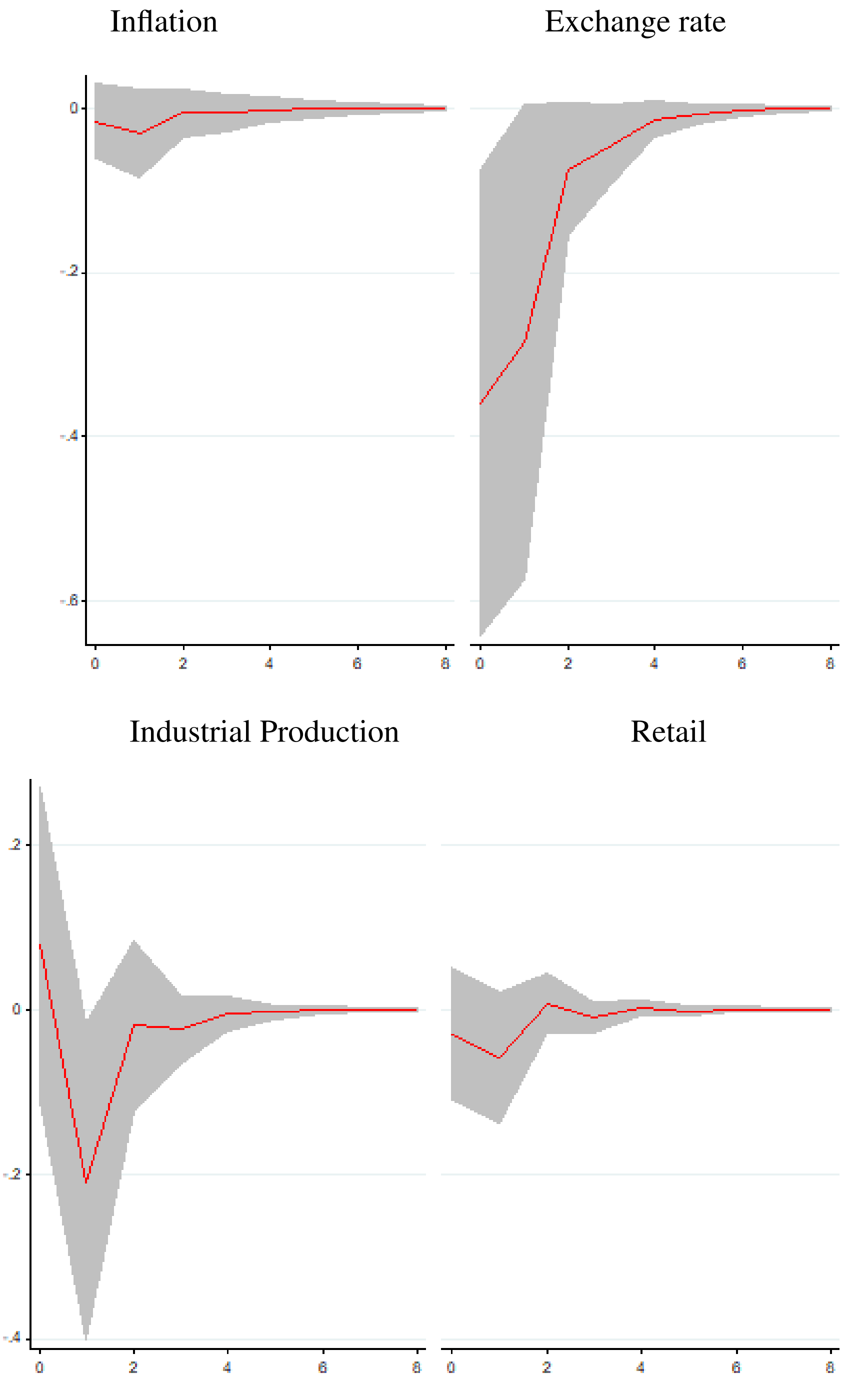


Figure A.4: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in the ‘inflation’ and ‘monetary policy decision’ topic-uncertainty index of the minutes of the COPOM from 2000 to June 2016. The gray area displays the 90% confidence intervals computed using bootstrapped standard errors (200 replications). The Y-axis is in % points change for each one of the four macroeconomic variables and the X-axis represents time in months (8 months).

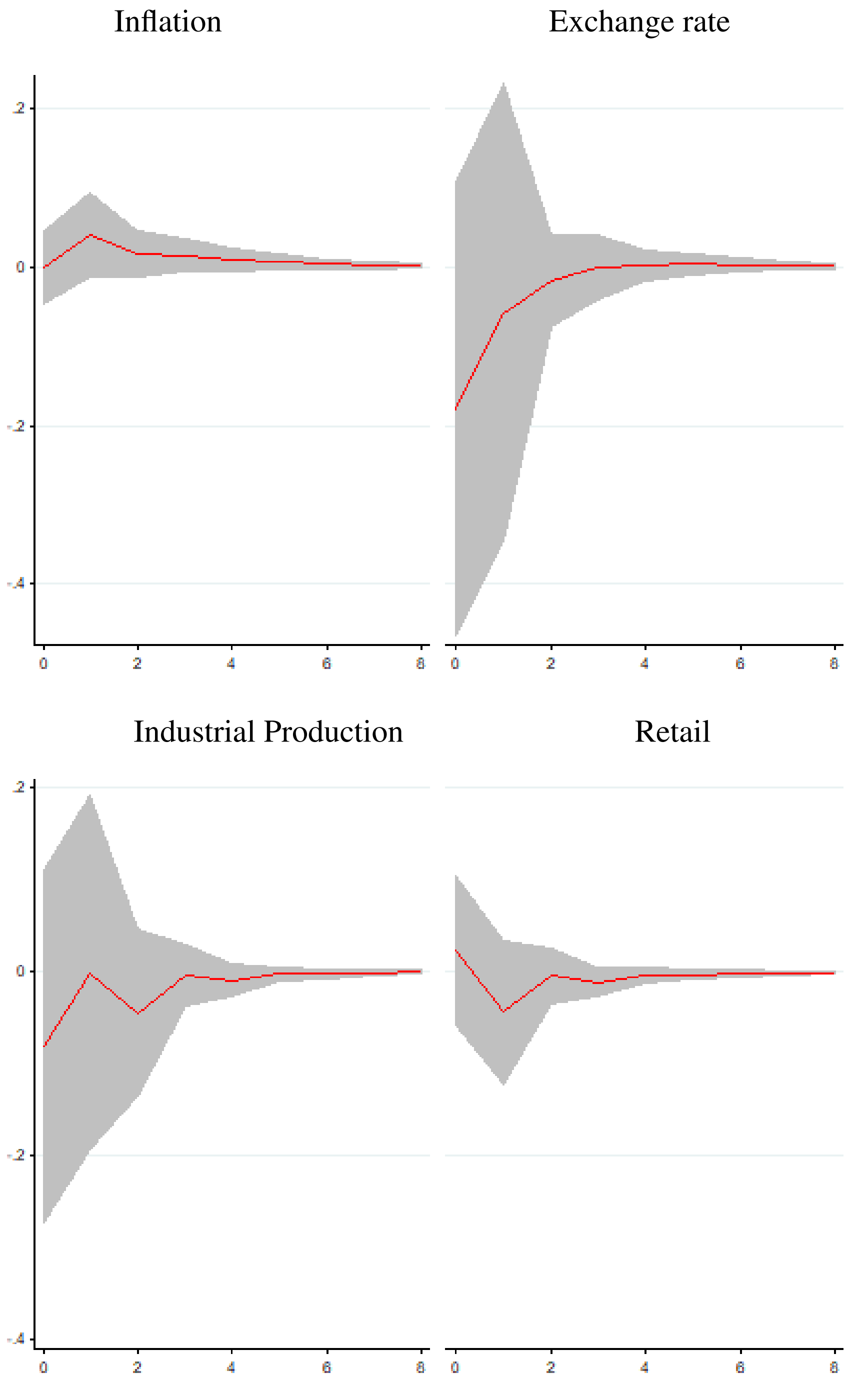
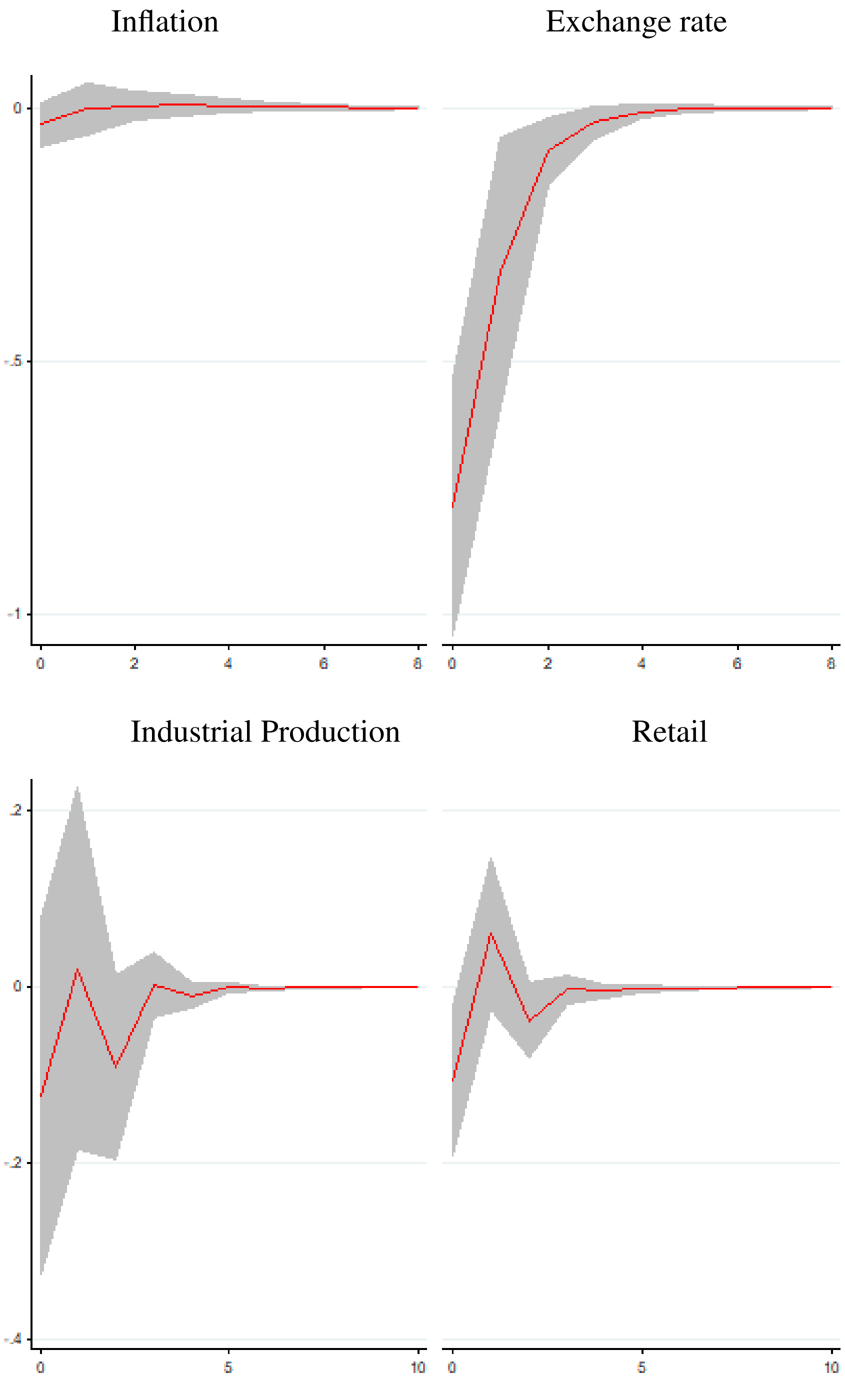


Figure A.5: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in the 'general economic conditions' topic-uncertainty index of the minutes of the COPOM from 2000 to June 2016. The gray area displays the 90% confidence intervals computed using bootstrapped standard errors (200 replications). The Y-axis is in % points change for each one of the four macroeconomic variables and the X-axis represents time in months (8 months).

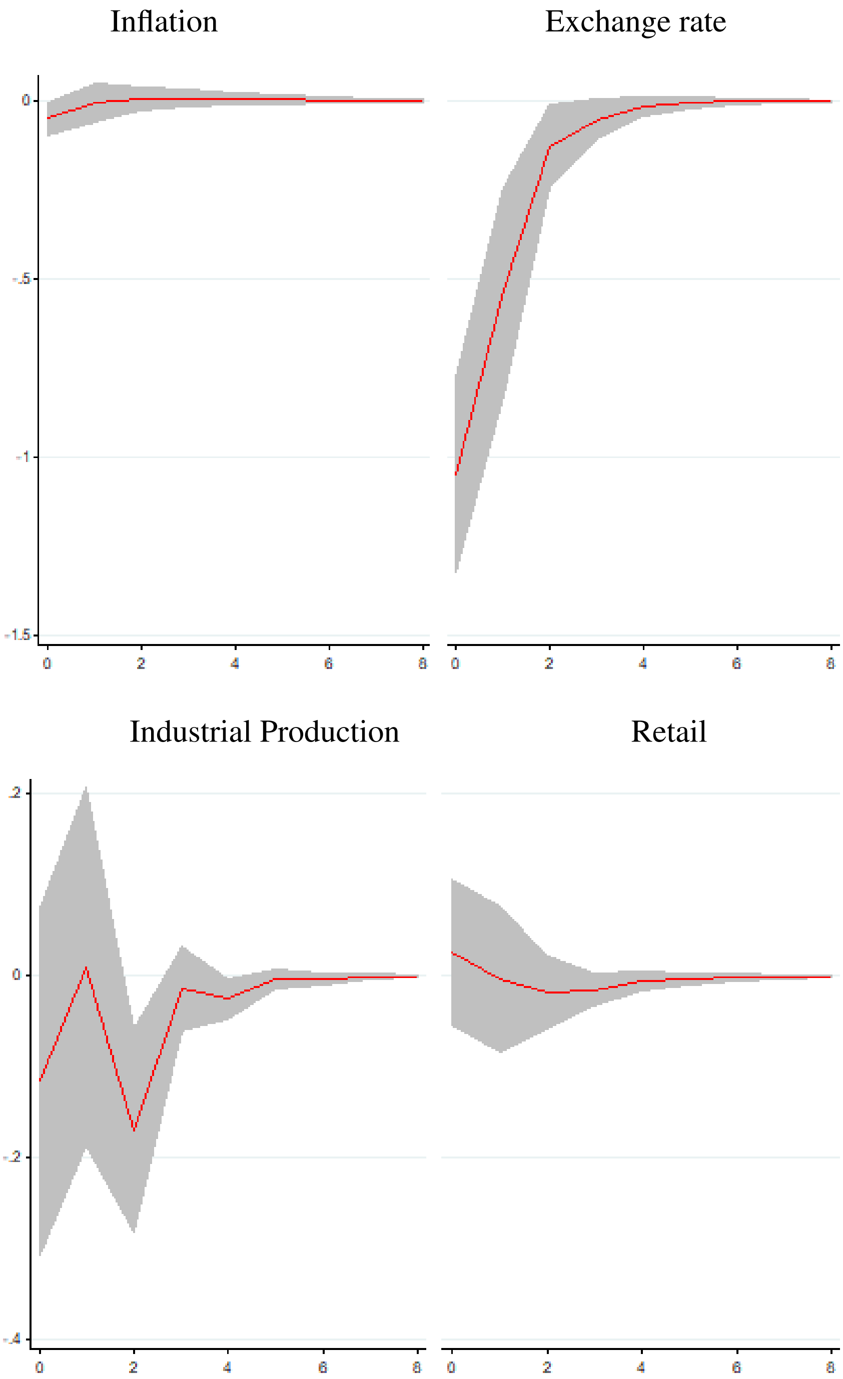

Inflation

Exchange rate

Industrial Production

Retail

Figure A.6: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in the Economic Policy Uncertainty (EPU) index for Brazil created by Baker, Bloom, and Davis (2016) from 2000 to July 2019. The gray area displays the 90% confidence intervals computed using bootstrapped standard errors (200 replications). The Y-axis is in % points change for each one of the four macroeconomic variables and the X-axis represents time in months (8 months).



Inflation

Exchange rate

Industrial Production

Retail

Figure A.7: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in the Economic Policy Uncertainty (EPU) index for Brazil created by Baker, Bloom, and Davis (2016) from 2000 to June 2016. The gray area displays the 90% confidence intervals computed using bootstrapped standard errors (200 replications). The Y-axis is in % points change for each one of the four macroeconomic variables and the X-axis represents time in months (8 months).

Chapter 2

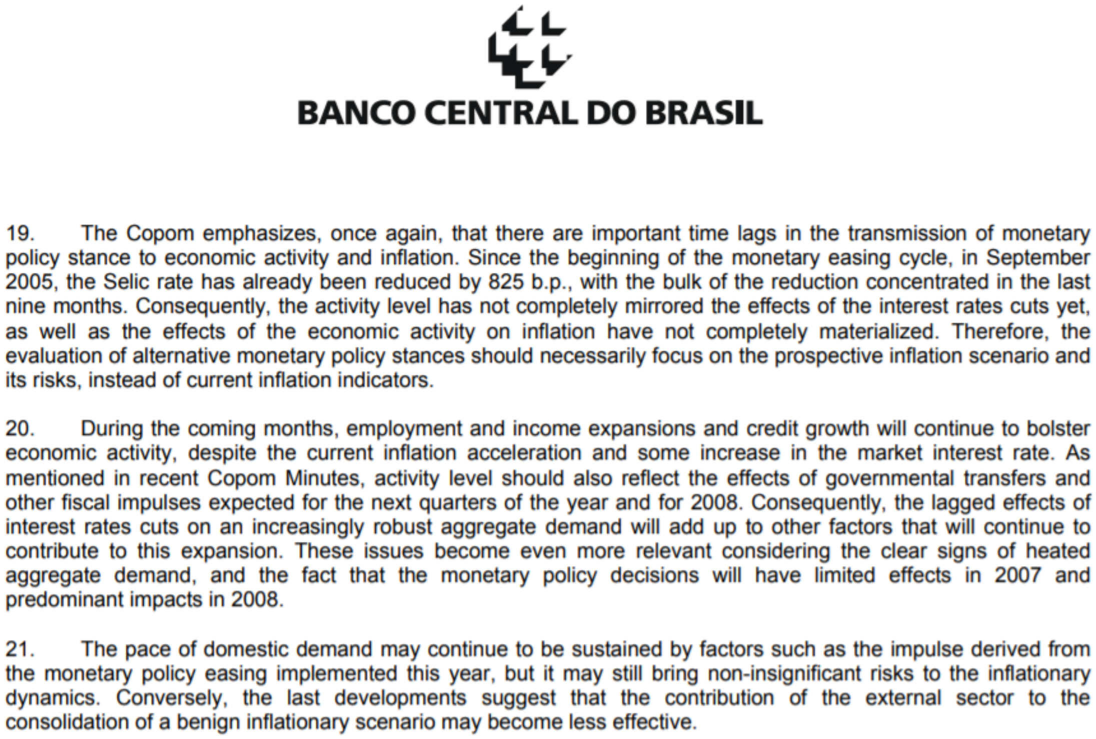
Supplementary Material - Making Text Talk: The Minutes of the Central Bank of Brazil and the Real Economy

2.1 Text Database: The Minutes of the Central Bank of Brazil

This paper investigates the relationship between the views expressed in the minutes of the meetings of the Central Bank of Brazil's Monetary Policy Committee (COPOM) and the real economy. We use the English version of the minutes of the COPOM as a proxy of the Portuguese version. We extract the minutes from the Central Bank of Brazil's web page in PDF format.[1] Figure 1 shows an example of three paragraphs of the 'monetary policy decision' section of the 129th minute in 2007.

This paper applies various computational linguistic machine learning algorithms to construct measures of the minutes of the COPOM. To apply these algorithms, we manually transform the PDF of each set of minutes into text files with unicode UTF-8 format. We remove from the minutes the parts that are not relevant for the LDA and the Skip-Gram models such as the cover, the introduction, the footnotes and acronyms. We also assign tags to each paragraph to identify the date, the number and section of the minutes. Figure 2 shows one of the paragraphs of Figure 1 with the tags and without the irrelevant parts. All the words are changed to lower case such as in Figure 3. Finally, we attach a copy of the text database of the minutes in the complementary material folder with the name 'Text_database_COPOM_2019.txt'.

[1] https://www.bcb.gov.br/en/publications/copomminutes



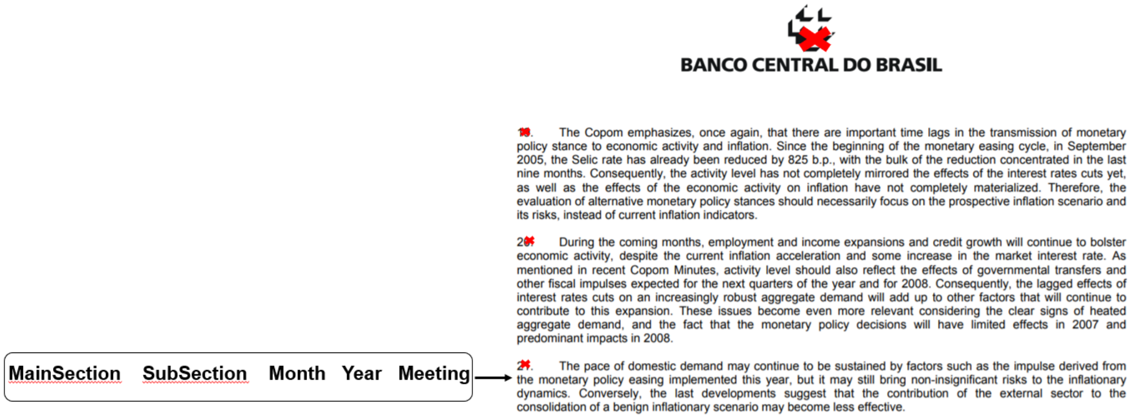
BANCO CENTRAL DO BRASIL

19. The Copom emphasizes, once again, that there are important time lags in the transmission of monetary policy stance to economic activity and inflation. Since the beginning of the monetary easing cycle, in September 2005, the Selic rate has already been reduced by 825 b.p., with the bulk of the reduction concentrated in the last nine months. Consequently, the activity level has not completely mirrored the effects of the interest rates cuts yet, as well as the effects of the economic activity on inflation have not completely materialized. Therefore, the evaluation of alternative monetary policy stances should necessarily focus on the prospective inflation scenario and its risks, instead of current inflation indicators.

20. During the coming months, employment and income expansions and credit growth will continue to bolster economic activity, despite the current inflation acceleration and some increase in the market interest rate. As mentioned in recent Copom Minutes, activity level should also reflect the effects of governmental transfers and other fiscal impulses expected for the next quarters of the year and for 2008. Consequently, the lagged effects of interest rates cuts on an increasingly robust aggregate demand will add up to other factors that will continue to contribute to this expansion. These issues become even more relevant considering the clear signs of heated aggregate demand, and the fact that the monetary policy decisions will have limited effects in 2007 and predominant impacts in 2008.

21. The pace of domestic demand may continue to be sustained by factors such as the impulse derived from the monetary policy easing implemented this year, but it may still bring non-insignificant risks to the inflationary dynamics. Conversely, the last developments suggest that the contribution of the external sector to the consolidation of a benign inflationary scenario may become less effective.

Figure 1: Paragraphs of the ‘monetary policy decision’ section of the 129th minute in 2007.



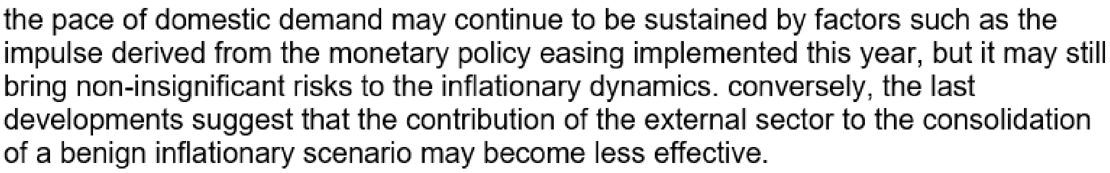
BANCO CENTRAL DO BRASIL

~~19~~. The Copom emphasizes, once again, that there are important time lags in the transmission of monetary policy stance to economic activity and inflation. Since the beginning of the monetary easing cycle, in September 2005, the Selic rate has already been reduced by 825 b.p., with the bulk of the reduction concentrated in the last nine months. Consequently, the activity level has not completely mirrored the effects of the interest rates cuts yet, as well as the effects of the economic activity on inflation have not completely materialized. Therefore, the evaluation of alternative monetary policy stances should necessarily focus on the prospective inflation scenario and its risks, instead of current inflation indicators.

~~20~~. During the coming months, employment and income expansions and credit growth will continue to bolster economic activity, despite the current inflation acceleration and some increase in the market interest rate. As mentioned in recent Copom Minutes, activity level should also reflect the effects of governmental transfers and other fiscal impulses expected for the next quarters of the year and for 2008. Consequently, the lagged effects of interest rates cuts on an increasingly robust aggregate demand will add up to other factors that will continue to contribute to this expansion. These issues become even more relevant considering the clear signs of heated aggregate demand, and the fact that the monetary policy decisions will have limited effects in 2007 and predominant impacts in 2008.

~~21~~. The pace of domestic demand may continue to be sustained by factors such as the impulse derived from the monetary policy easing implemented this year, but it may still bring non-insignificant risks to the inflationary dynamics. Conversely, the last developments suggest that the contribution of the external sector to the consolidation of a benign inflationary scenario may become less effective.

Figure 2: Paragraph tagging and elimination of non-relevant parts.



the pace of domestic demand may continue to be sustained by factors such as the impulse derived from the monetary policy easing implemented this year, but it may still bring non-insignificant risks to the inflationary dynamics. conversely, the last developments suggest that the contribution of the external sector to the consolidation of a benign inflationary scenario may become less effective.

Figure 3: Lower case transformation of the text.

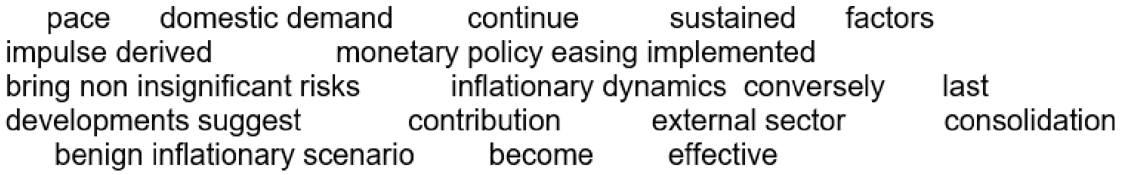
2.2 Latent Dirichlet Allocation

This sections explains the application of Latent Dirichlet Allocation (LDA). First, the data are ‘cleaned’ before applying LDA. The ‘cleaning’ data process for LDA requires three steps eliminating non-relevant information from the text. The second section shows a figure for further understanting of the LDA theory. We then show the python code to estimate LDA. Finally, we include the python code to estimate Figures 1 and 2 of the paper that show the weights of the LDA topics.

2.2.1 Latent Dirichlet Allocation: text pre-processing

The ‘cleaning’ data process for LDA requires three steps eliminating non-relevant information from the text. The first step is to remove the punctuation and stop words such as ‘the’, ‘all’, ‘because’, ‘this’, not relevant since they provide no information about the theme of the paragraph which is shown in Figure 4.[2] The second step is to stem the remaining words. Stemming is a process that consists in reducing words into their word stem or base root. For instance, the words ‘inflationary’, ‘inflation’, ‘consolidate’ and ‘consolidating’ are transformed into their stem ‘inflat’ and ‘consolid’, respectively. Figure 5 shows the stems of the words in Figure 4. Finally, we rank these stems according to the term frequency-inverse document frequency (tf-idf). This index grows proportionally with the number of times a stem appears in a document. However, it decreases by the number of documents that contain that stem. This index serves to eliminate common and unusual words. We disregard all stems that have a value of 3,000 or lower.

After the pre-processing, our corpus comprises 9,484 paragraphs of all the minutes from the end of 1999 to September 2019. Our corpus also comprises 2,900 unique stems and the total number of stems is 450,174.



pace domestic demand continue sustained factors
impulse derived monetary policy easing implemented
bring non insignificant risks inflationary dynamics conversely last
developments suggest contribution external sector consolidation
benign inflationary scenario become effective

Figure 4: Removal of the punctuation signs and the stop words.

[2] We include the words of the different months of the year and the word ‘year’ as stop words in order to eliminate seasonality or topics referring to a particular quarter.

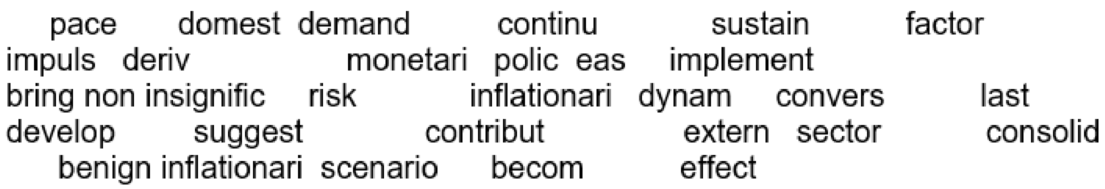

pace domest demand continu sustain factor
impuls deriv monetari polic eas implement
bring non insignific risk inflationari dynam convers last
develop suggest contribut extern sector consolid
benign inflationari scenario becom effect

Figure 5: Stemming of words.

2.2.2 Latent Dirichlet Allocation: theory

We display and extra figure to understand the LDA topic assignment and word-topic assignment that is described in the paper.

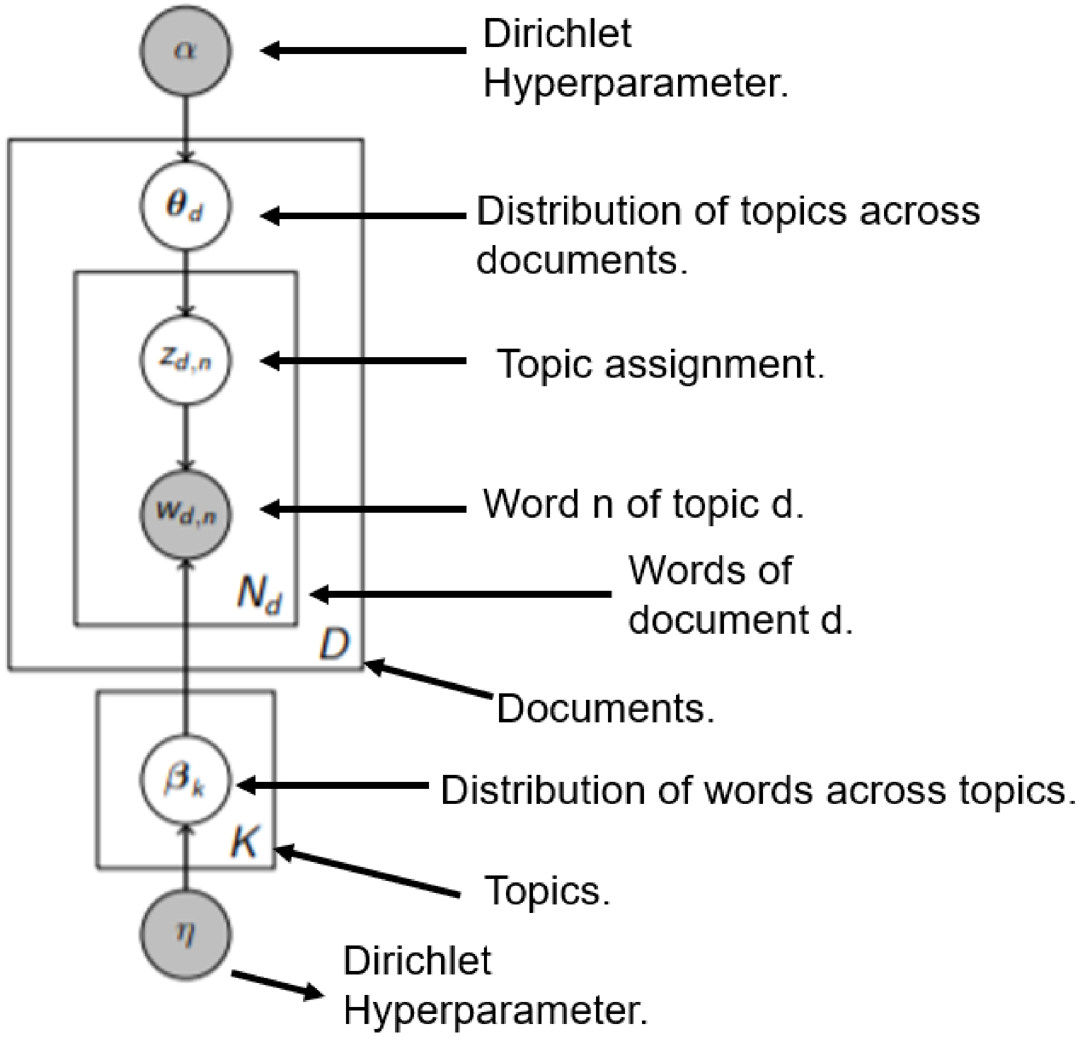


Figure 6: LDA plate diagram (Hansen, McMahon and Prat; 2017).

2.2.3 Latent Dirichlet Allocation: estimation

To apply Latent Dirichlet allocation, we use most of the python code provided by the Professor Stephen Hansen of the Imperial College Business School.[3] The python code used is shown in the following lines:

```python
import pandas as pd
import topicmodels
import matplotlib.pyplot as plt
import matplotlib
import numpy as np
import re
from gensim.utils import simple_preprocess
import pyLDAvis

#Opening the dataset of the minutes of the COPOM.
data = pd.read_table("Text_database_COPOM_2019.txt", encoding="utf-8")
data = data[data.year >= 2000]

#Replacing the paragraphs section tag errors (re, recc) in the dataset for the correct tag (rec).
data.main = data.main.str.strip().str.lower().str.replace('re','rec')
data.main = data.main.str.strip().str.lower().str.replace('recc','rec')

#Changing the paragraphs section tags to numerical values.
changemain = {'rec': 0,'ait': 1,'mpd': 2}
data.main = [changemain[item] for item in data.main]
print(data)

#Using long list of the English stopwords and including the months of the year and the word 'year'  in the stopwords.
docsobj = topicmodels.RawDocs(data.speech, "long")
docsobj.stopwords.add(unicode('january'))
docsobj.stopwords.add(unicode('february'))
docsobj.stopwords.add(unicode('march'))
```

[3] https://github.com/sekhansen

```python
docsobj.stopwords.add(unicode('april'))
docsobj.stopwords.add(unicode('may'))
docsobj.stopwords.add(unicode('june'))
docsobj.stopwords.add(unicode('july'))
docsobj.stopwords.add(unicode('june'))
docsobj.stopwords.add(unicode('august'))
docsobj.stopwords.add(unicode('september'))
docsobj.stopwords.add(unicode('october'))
docsobj.stopwords.add(unicode('november'))
docsobj.stopwords.add(unicode('december'))
docsobj.stopwords.add(unicode('year'))

#Cleaning the dataset.
docsobj.token_clean(1)

#We remove stopwords.
docsobj.stopword_remove("tokens")

#We stem the corpus.
docsobj.stem()
docsobj.stopword_remove("stems")

#We rank these stems  according to the term frequency-inverse document frequency (tf-idf).
docsobj.term_rank("stems")

#We disregard all stems that have a value of the tfidf ranking of 3,000 or lower.
docsobj.rank_remove("tfidf", "stems", docsobj.tfidf_ranking[3000][1])

#Plotting the tfidf ranking.
plt.plot([x[1] for x in docsobj.tfidf_ranking])

#Printing number of unique and total stems in the database.
all_stems = [s for d in docsobj.stems for s in d]
print("number of unique stems = %d" % len(set(all_stems)))
print("number of total stems = %d" % len(all_stems))

# Estimatation of LDA where 9 is the number of topics.
```

```
ldaobj = topicmodels.LDA.LDAGibbs(docsobj.stems, 9)

# We run 20 samples from points in the chain that are
 ↪  thinned with a thinning interval of 50.
ldaobj.sample(1000, 50, 20)
print ldaobj.perplexity()
ldaobj.sample(1000, 50, 20)
print ldaobj.perplexity()

ldaobj.samples_keep(4)
ldaobj.topic_content(20)

dt = ldaobj.dt_avg()
tt = ldaobj.tt_avg()
ldaobj.dict_print()

#LDA estimation.
data = data.drop('speech', 1)
for i in range(ldaobj.K):
    data['T' + str(i)] = dt[:, i]
data.to_csv("topics_document_COPOM.csv", index=False)

#We query the output by topics per minutes.
data['speech'] = [' '.join(s) for s in docsobj.stems]
aggspeeches = data.groupby(['year', 'meeting'])['speech'].\
    apply(lambda x: ' '.join(x))
aggdocs = topicmodels.RawDocs(aggspeeches)

queryobj = topicmodels.LDA.QueryGibbs(aggdocs.tokens,
 ↪  ldaobj.token_key,
                                      ldaobj.tt)
queryobj.query(10)
queryobj.perplexity()
queryobj.query(30)
queryobj.perplexity()

dt_query = queryobj.dt_avg()
aggdata = pd.DataFrame(dt_query, index=aggspeeches.index,
                       columns=['T' + str(i) for i in
                        ↪  range(queryobj.K)])
```

```
aggdata.to_csv("final_output_agg_brazil_3000_1000_10
 ↪  _WithoutM.csv")

#We query the output by topics per sections.
data['speech'] = [' '.join(s) for s in docsobj.stems]
aggspeeches1 = data.groupby(['year','meeting',
 ↪  'main'])['speech'].\
    apply(lambda x: ' '.join(x))
aggdocs1 = topicmodels.RawDocs(aggspeeches1)

queryobj1 = topicmodels.LDA.QueryGibbs(aggdocs1.tokens,
 ↪  ldaobj.token_key,
                                      ldaobj.tt)
queryobj1.query(10)
queryobj1.perplexity()
queryobj1.query(30)
queryobj1.perplexity()

dt_query1 = queryobj1.dt_avg()
aggdata1 = pd.DataFrame(dt_query1,
 ↪  index=aggspeeches1.index,
                       columns=['T' + str(i) for i in
                        ↪  range(queryobj.K)])
aggdata1.to_csv("final_output_agg_sections_3000_1000_10
 ↪  _WithoutM.csv")
```

The results are not reproducible. However, the results tend always to be similar after several trials. The following list shows the name of the python code and the different outputs included in the supplementary material folder. An explanation of each document is given within brackets.

1. ‘LDA_Brazil.py’ (Python code to estimate LDA);
2. ‘Topic description.csv’ (LDA output: words per topic);
3. ‘final_output_brazil2.csv’ (LDA output: topics per document);
4. ‘final_output_agg_brazil_3000_1000_10_WithoutM.csv’ (LDA output: topics per minute);
5. ‘final_output_agg_sections_3000_1000_10_WithoutM.csv’ (LDA output: topics per section);
6. ‘df_ranking.csv’ (LDA output: ranking of stems by document frequency);

7. ‘tfidf_ranking.csv’ (LDA output: ranking of stems by tf-idf measure).

2.2.4 Latent Dirichlet Allocation: graphs

This section shows the python the code to construct the graphs of the weights of the topics. First, we construct Figure 1 of the paper that shows the weights of the topics related to the ‘general economic conditions’. We then show the code to construct Figure 2 of the paper which shows the weights of the topics related to ‘inflation’ and the ‘monetary policy decision’. The date of the meeting is used in the graph. The excel file that includes the date of the meetings is ‘minutes_date.csv’. To assign the date to each meeting, we merge the latter file with the file ‘topics per minutes.csv’. The python code to construct the graphs is included in the supplementary material folder with the name ‘graph_lda_brasil.py’. The python code is shown in the following lines:

```python
from pylab import *
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.patches as mpatches
from matplotlib import pyplot
import Pyro4
import seaborn as sns

#Loading 'topics per minutes' output in python as a
 ↪  DataFrame.
minutes = pd.read_csv("final_output_agg_brazil_3000_1000_10
_WithoutM.csv", encoding="utf-8")

#Loading 'dates of the minutes' excel file as a DataFrame.
date = pd.read_csv("minutes_date.csv", sep = ';', encoding
 ↪  = "utf-8")

#Merging 'minutes' DataFrame with with 'date' DataFrame in
 ↪  a new DataFrame.
minutes_date = pd.merge(minutes, date,  how='left',
 ↪  left_on=['meeting'], right_on = ['meeting'])

#Changing format of the 'date' column from object to
 ↪  datetime64[ns].
```

```
minutes_date['date'] =
 ↪  pd.to_datetime(minutes_date['date'],infer_datetime_formatv
 ↪  =True,dayfirst=True)

#Checking  if the format of the 'minute_date' DataFrame is
 ↪  the correct one.
minutes_date.dtypes

#Setting 'date' column of the 'minutes_date' DataFrame as
 ↪  index.
minutes_date = minutes_date.set_index('date')

minutes_date.head(3)

# Use seaborn style defaults and set the default figure
 ↪  size
sns.set(rc={'figure.figsize':(11, 8)})

#Graph of the weights of the topics related to 'monetary
 ↪  policy decision' and 'inflation'.
minutes_date['T0'].plot(color='orange')
minutes_date['T1'].plot(color='red')
minutes_date['T3'].plot(color='blue')
minutes_date['T5'].plot(color='green')
plt.ylabel("Probability of the topic in each COPOM's
 ↪  minute")
plt.xlabel("Minutes across time")
axvline('2001-05-23', color='red', ls="dotted")
axvline('2003-01-22', color='blue', ls="dotted")
axvline('2003-04-23', color='red', ls="dotted")
axvline('2005-09-14', color='red', ls="dotted")
axvline('2011-01-19', color='blue', ls="dotted")
axvline('2014-02-26', color='red', ls="dotted")
axvline('2016-07-20', color='black', ls="dotted")
axvline('2019-03-20', color='blue', ls="dotted")
orange_patch = mpatches.Patch(color='orange', label='T0
 ↪  (Inflation)')
red_patch = mpatches.Patch(color='red', label='T1
 ↪  (Inflation)')
```

```
blue_patch = mpatches.Patch(color='blue', label='T3 (COPOM decision)')
green_patch = mpatches.Patch(color='green', label='T5 (COPOM decision)')
plt.legend(handles=[orange_patch, red_patch,blue_patch,green_patch],loc='center left', bbox_to_anchor=(0, 0.9))

lll

#Graph of the weights of the topics related to 'general economic conditions'.
minutes_date['T2'].plot(color='red')
minutes_date['T4'].plot(color='lime')
minutes_date['T6'].plot(color='yellow')
minutes_date['T7'].plot(color='blue')
minutes_date['T8'].plot(color='pink')
plt.ylabel("Probability of the topic in each COPOM's minute")
plt.xlabel("Minutes across time")
axvline('2001-05-23', color='red', ls="dotted")
axvline('2003-01-22', color='blue', ls="dotted")
axvline('2003-04-23', color='red', ls="dotted")
axvline('2005-09-14', color='red', ls="dotted")
axvline('2011-01-19', color='blue', ls="dotted")
axvline('2014-02-26', color='red', ls="dotted")
axvline('2016-07-20', color='black', ls="dotted")
axvline('2019-03-20', color='blue', ls="dotted")
red_patch = mpatches.Patch(color='red', label='T2 (Economic activity)')
lime_patch = mpatches.Patch(color='lime', label='T4 (Trade / credit Operations)')
yellow_patch = mpatches.Patch(color='yellow', label='T6 (Sales / retail)')
blue_patch = mpatches.Patch(color='blue', label='T7 (Employment)')
pink_patch = mpatches.Patch(color='pink', label='T8 (Industrial production)')
```

```
plt.legend(handles=[red_patch,lime_patch, yellow_patch, blue_patch, pink_patch],loc='center left', bbox_to_anchor=(0.22, 0.9))
```

2.3 Skip-Gram and K-Means

This section shows the python codes to construct the uncertainty index. First, the Skip-Gram model is applied to the same corpus of minutes of the Central Bank of Brazil. Nonetheless, there are some differences in the preprocessing of the corpus. After applying the Skip-Gram and K-Means models, we select all the words in the same clusters as 'uncertainty', 'uncertain', 'uncertainties' and 'fears' to construct a dictionary or list of words related to uncertainty. We assume that the words in the same clusters share a similar semantic meaning. We also construct topic-uncertainty indices. Finally, we make graphs of the evolution of the uncertainty and the topic-uncertainty indices.

2.3.1 Skip-Gram and K-Means: text pre-processing

This section explains the python code of the 'cleaning' process before we apply the Skip-Gram model. Most of the pre-processing python code is obtained from the web page machinelearningplus.com.[4] This python code is included in the supplementary folder with the name 'brazil_skipgram_preprocessing.py'. The final output is saved like 'COPOM_minutes_word2vec_disordered.txt' and it is also saved without format as 'COPOM_minutes_word2vec_ordered'.

```
import nltk; nltk.download('stopwords')
import re
import numpy as np
import pandas as pd
from pprint import pprint

#Gensim.
import gensim
import gensim.corpora as corpora
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel
import pickle

#NLTK stop words.
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
```

[4] https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/

```python
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])

#Opening database minutes of the COPOM as DataFrame 'df'.
df = pd.read_table("Text_database_COPOM_2019.txt", encoding="utf-8")
df= df[df.year >= 2000]

#Converting the 'speech' column of the 'df' DataFrame to list.
data = df.speech.values.tolist()

#Removing symbols of the list 'data'.
data = [re.sub('\S*@\S*\s?', '', sent) for sent in data]

#Removing new line characters of the list 'data'.
data = [re.sub('\s+', ' ', sent) for sent in data]

#Removing the distracting single quotes of the list 'data'.
data = [re.sub("\'", "", sent) for sent in data]
pprint(data[:1])

#Defining function to pass format from list of strings to list of lists.
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))  # deacc=True removes punctuations

#Passing format of 'data' from list of strings to list of lists.
data_words = list(sent_to_words(data))
print(data_words[:1])

#Constructing the bigram model.
bigram = gensim.models.Phrases(data_words, min_count=5, threshold=10)
bigram_mod = gensim.models.phrases.Phraser(bigram)

#Definition of the functions for stopwords and bigrams.
def remove_stopwords(texts):
    return [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for doc in texts]

def make_bigrams(texts):
    return [bigram_mod[doc] for doc in texts]

#We remove the stop words.
data_words_nostops = remove_stopwords(data_words)

#We constuct the bigrams.
data_words_bigrams = make_bigrams(data_words_nostops)

#Passing format of 'data_words_bigrams' from a list of lists to a list of strings.
implodeList = []

for item in data_words_bigrams :
    implodeList.append(' '.join(item))

#Adding as a column the pre-processed minutes in the 'df' dataframe as 'data_words_bigrams'.
df['data_words_bigrams'] = implodeList

#Saving the pre-processed data in txt file.
with open('COPOM_minutes_word2vec_disordered.txt', 'w', encoding = 'utf-8') as f:
    for item in df.data_words_bigrams:
        f.write("%s " % item)

#Saving the pre-processed data without format.
with open('COPOM_minutes_word2vec_ordered', 'wb') as fp:
    pickle.dump(df.data_words_bigrams, fp, protocol =2)

with open('COPOM_minutes_word2vec_ordered', 'rb') as fp:
    df['database'] = pickle.load(fp)
```

2.3.2 Skip-Gram and K-Means: estimation

We construct an ‘uncertainty’ dictionary with the Skip-Gram model and K-Means. These algorithms are estimated with Python 2.7. Most of the python code to apply the Skip-

Gram model is obtained from the github web page of professor Florian Leitner.[5] Word2Vec of the gensim package is used to estimate Word Embeddings with the Skip-Gram model. K-Means is implemented with the code provided by the webpage ‘https://ai.intelligentonlinetools.com/’.[6]

The python code to estimate the Skip-Gram model and K-Means is available in the supplementary material folder with the name ‘Skip-Gram - K-Means estimation.py’. The python code is shown in the following lines:

```python
import gensim # for Word2Vec
import nltk
from IPython.display import HTML
import re
import string
import pandas as pd
from gensim.models import Word2Vec
from nltk.cluster import KMeansClusterer
from sklearn import cluster
from sklearn import metrics

#We prepare the dataset for Word2Vec.
#We open the text database of the minutes of the Central Bank of Brazil.
with open('COPOM_minutes_word2vec_disordered.txt') as f:
    tokens_bigrams = f.read().split()

print("raw n. tokens =", len(tokens_bigrams))

with open('text9_collocations', 'wt') as f:
    f.write(" ".join(tokens_bigrams ))

with open('text9_collocations') as f:
    phrases = f.read().split()

HTML(" ".join(tokens_bigrams [:100]))

def text8_to_sentences(tokens):
```

[5] https://github.com/fnl/asdm-tm-class, Florian Leitner teaches the ‘text mining’ course of the Madrid UPM Machine Learning and Advanced Statistics Summer School

[6] The article is titled ‘K Means Clustering Example with Word2Vec in Data Mining or Machine Learning’

```
    """The models insist on sentences; Let's build some."""
    index = 0
    inc = 200

    while index + inc < len(tokens):
        yield tokens[index:index+inc]
        index += inc

    yield tokens[index:]

sentences = list(text8_to_sentences(tokens_bigrams))

#Constuction of Word Embeddings with Word2Vec.
PYTHONHASHSEED=999 #Computed with Python 2.7
#In Python 3, to make the results reproducible we should
 ↪  set the seed as `set PYTHONASHSEED=0' in the terminal
 ↪  before opening Python. Then, we should open Python from
 ↪  the terminal.

#Size indicates the window size of the Skip-Gram model, and
 ↪  window is the size of the context words. Set sg = 1 and
 ↪  workers = 1 to be able to reproduce the results.
model =
 ↪  gensim.models.Word2Vec(list(text8_to_sentences(phrases)),
 ↪  sg=1, size=200, window=10, seed=999, workers=1)
print(model==0)

print (list(model.wv.vocab))
print (len(list(model.wv.vocab)))
print(model)

X = model[model.wv.vocab]

print (model.similarity('uncertainty', 'uncertainties'))

print (list(model.wv.vocab))

print (len(list(model.wv.vocab)))

model.most_similar('uncertainty', topn=40)
```

```

 #Estimation of clusters of the Word Embeddings with K-Means Clustering.
#Number of clusters.
NUM_CLUSTERS=140

import random

#Setting seed for reproducibility.
rng = random.Random()
rng.seed(123)

#Estimation of K-Means.
kclusterer = KMeansClusterer(NUM_CLUSTERS, distance=nltk.cluster.util.cosine_distance, repeats=25, rng = rng)
assigned_clusters = kclusterer.cluster(X, assign_clusters=True)
print (assigned_clusters)

words = list(model.wv.vocab)
for i, word in enumerate(words):
    print (word + ":" + str(assigned_clusters[i]))

kmeans = cluster.KMeans(n_clusters=NUM_CLUSTERS)
kmeans.fit(X)

labels = kmeans.labels_
centroids = kmeans.cluster_centers_

print ("Cluster id labels for inputted data")
print (labels)
print ("Centroids data")
print (centroids)

print ("Score (Opposite of the value of X on the K-means objective which is Sum of distances of samples to their closest cluster center):")
print (kmeans.score(X))

```

```
silhouette_score = metrics.silhouette_score(X, labels,
 ↪  metric='euclidean')

print ("Silhouette_score: ")
print (silhouette_score)

cluster_list = pd.DataFrame(
    {'assigned_clusters': assigned_clusters,
     'words': words
    })

#Clusters of the words 'uncertain', 'uncertainty',
 ↪  'uncertainties' and 'fears'.
uncertain =
 ↪  cluster_list.loc[cluster_list['assigned_clusters'] ==
 ↪  115]

uncertainty =
 ↪  cluster_list.loc[cluster_list['assigned_clusters'] ==
 ↪  94]

uncertainties =
 ↪  cluster_list.loc[cluster_list['assigned_clusters'] ==
 ↪  115]

fears = cluster_list.loc[cluster_list['assigned_clusters']
 ↪  == 58]

#Saving in excel the clusters of the words 'uncertain',
 ↪  'uncertainty', 'uncertainties' and 'fears'.
uncertain.to_excel('uncertain_list_words_k140_s200_w10.xlsx'
                   )
uncertainty.to_excel('uncertainty_list_words_k140_s200
_w10.xlsx')

uncertainties.to_excel('uncertainties_list_words_k140_s200
_w10.xlsx')

fears.to_excel('fears_list_words_k140_s200_w10.xlsx')
```

The complementary material folder includes the list of words of the clusters of ‘uncertain', 'uncertainty', 'uncertainties' and 'fears'. One migth be aware that the clusters of the words 'uncertain' and 'uncertainties' are the same. Moreover, we manually include the words of the clusters of 'uncertain', 'uncertainty', 'uncertainties' and 'fears' into one excel file. The documents attached in the complementary material are listed in the following list:

1. 'uncertain_list_words_k140_s200_w10.xlsx' (List of words of the cluster of the word 'uncertain');
2. 'uncertainty_list_words_k140_s200_w10.xlsx' (List of words of the cluster of the word 'uncertainty');
3. 'uncertainties_list_words_k140_s200_w10.xlsx' (List of words of the cluster of the word 'uncertainties');
4. 'fears_list_words_k140_s200_w10.xlsx' (List of words of the cluster of the word 'fears');
5. 'Brazil_uncertainty-fears_wordslist_k140_s200_w10.xlsx' (Combination of the words of the lists of the words 'uncertain', 'uncertainty', 'uncertainties', 'fears').

2.3.3 Skip-Gram and K-Means: construction of uncertainty and topic-uncertainty indices

This sections shows the python code ('Brazil_count-words-uncertainty.py') to count the frequency of the 'uncertainty' dictionary and the total number of words of each paragraph. The output is saved in a csv file as 'Brazil_CountWords_uncertainty_2019.csv'.

```python
import pandas as pd
import matplotlib.pyplot as plt
import pickle

#Loading COPOM database as DataFrame 'df'.
df = pd.read_table("text_database_COPOM_2019.txt", encoding="utf-8")
df= df[df.year >= 2000]

#Loading pre-processed COPOM's minutes database for Skip-Gram as a column of the DataFrame 'df'.
with open ('COPOM_minutes_word2vec_ordered', 'rb') as fp:
    df['database_skipgram'] = pickle.load(fp)

#Loading the 'uncertainty' dictionary  as the DataFrame
  ↪  'data'.
data = pd.read_csv
  ↪  ("Brazil_uncertainty-fears_wordslist_k140_s200_w10.csv",
  ↪  sep = ",", encoding="utf-8")

#Passing the 'uncertainty' dictionary from a column of the
  ↪  'data' DataFrame to a list.
uncer_index = data['words']
implodeList = list(uncer_index)

#Passing the 'uncertainty' dictionary from low to upper
  ↪  capital letters.
uncertainty = []
for word in implodeList:
   uncertainty.append(word.upper())
print(uncertainty)

#We create two new columns in the 'df' DataFrame with the
  ↪  names 'UncerScore' and 'TotalWordCount'.
df = pd.concat([df, pd.DataFrame(columns = ['UncerScore']),
                  pd.DataFrame(columns =
                   ↪  ['TotalWordCount'])])

#Computing the frequency of the 'uncertainty' words and the
  ↪  total number of words.
bow_uncer = []

for i,article in enumerate(df.database_skipgram):
    if str(article) != 'nan':
        m = 0
        for word in article.split(' '):
               if word.upper() in uncertainty:
                    m+= 1
                    bow_uncer.append(word)

        df.UncerScore[i]     = m
        df.TotalWordCount[i] = len(article.split(' '))
```

```

#Creating the DataFrame 'df_min' with some variables of the
 ↪  'df' DataFrame.
df_min = df[['year','UncerScore','meeting',
 ↪  'TotalWordCount','main','sub']].copy()

#Saving the DataFrame 'df_min' in an excel file.
df_min.to_csv("Brazil_CountWords_uncertainty_2019.csv")
```

2.3.4 Skip-Gram and K-Means: graphs

This section constructs an uncertainty index for the minutes of the COPOM. We construct two topic-uncertainty indices, creating the first topic-uncertainty index for the paragraphs more likely to include topics related to ‘general economic conditions’. Another topic-uncertainty index is created for the paragraphs more likely to include topics related to ‘inflation’ and the ‘monetary policy decision’. To build the two topic-uncertainty indices, we follow the same procedure as described for the general uncertainty index. We then create Figures 4 and 5 of the paper. Figure 4 shows the evolution of the uncertainty index. We compare it with the Economic Policy Uncertainty (EPU) index for Brazil created by Baker, Bloom, and Davis (2016) from the Brazilian newspaper ‘Folha de Sao Paulo’. Figure 5 shows the evolution of the two topic-uncertainty indices and we compare them again to the EPU index of Baker, Bloom, and Davis (2016) for Brazil. We extract the EPU index for Brazil from the web page of Baker, Bloom and David (2016).[7] We save the EPU index for Brazil in the supplementary material folder in the excel file ‘baker.csv’. Finally, we save the output in a dataset with the name ‘brazil_database_structuralvar_2019.csv’ including the values of the normalized uncertainty index, the topic-uncertainty indices and the normalized EPU index. The python code is included in the supplementary folder such as ‘brazil_construction-uncertainty-index_and_graphs.py’. The python code is show in the following lines:

```
from pylab import *
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.patches as mpatches
from datetime import datetime
import Pyro4
import seaborn as sns
```

[7] https://www.policyuncertainty.com/

```python

#Importing the uncertainty and total count words database as 'df' DataFrame.
df = pd.read_csv("Brazil_CountWords_uncertainty_2019.csv", sep = ',', encoding = "utf-8")

#Importing the LDA  output 'topics per documents' as 'data' DataFrame.
data = pd.read_csv("final_output_brazil2.csv", sep = ',', encoding = "utf-8")

#Importing the minutes database as 'brazil' DataFrame.
brazil = pd.read_table("Text_database_COPOM_2019.txt", encoding="utf-8")

#Adding as a column to the 'data' DataFrame the column'speech' of the 'brazil' DataFrame.
data['brasil'] = brazil['speech'].copy()

#We assign each paragraph to one of the two group of topics of LDA (0 - General economic conditions; 1 - Inflation and monetary policy decision).

#We sum the probabilities of the  topics related to 'inflation'.
data['inflation'] = data['T0'] + data['T1']
#We sum the probabilities of the  topics related to the 'monetary policy decision'.
data['copom'] = data['T3'] + data['T5']
#We sum the probabilities of the  topics related to the 'general economic conditions'.
data['gec'] = data['T2'] + data['T4'] + data['T6'] + data['T7']  + data['T8']

#We create a dummy variable for paragraph-topic assignment.
#We assign the value 0 if the paragraph is assigned to the 'general economic conditions' group of topics.
#We assign the value 1 if the paragraph is assigned to the 'inflation and monetary policy decision' group of topics.
```

```
data.loc[data.gec >= 0.555 , 'topic'] = 0
data.loc[data.gec < 0.555 , 'topic'] = 1

#We copy the column of topic assignment from 'data'
  ↪  DataFrame to 'df' DataFrame.
df['topic'] = data['topic'].copy()

#####################################################
##### Construction of minutes uncertainty index #####
#####################################################

#Grouping by the number of 'uncertainty' words and the
  ↪  total number of words per meeting in the DataFrame
  ↪  'temp_total'.
temp_total = df.groupby(['year',
  ↪  'meeting'])['TotalWordCount','UncerScore'
  ↪  ].sum().reset_index().rename(columns={'CombScore':
  ↪  'combsum'})

#The meeting 76 and 77 occured in the same month. Thus, we
  ↪  join them in the same observation.
temp_76 = temp_total.copy()

#We add the values of the minute 77 to the row of the
  ↪  meetings' minute 76.
temp_76.loc[34] += temp_76.loc[35]

#We drop the row 35 which corresponds to the minute of the
  ↪  meeting 77.
temp_76.drop([35], inplace=True)

#We change the values of the row  of the minute number 76
  ↪  that we did not want to change as year or meeting.
temp_76.at[34, 'year'] = 2002
temp_76.at[34, 'meeting'] = 76

#We load the 'minutes_date.csv' data set that contains the
  ↪  dates in which each meeting took place.
date = pd.read_csv("minutes_date.csv", sep = ';', encoding
  ↪  = "utf-8")
```

```

#We merge the 'minutes' DataFrame with the 'date'
 ↪  DataFrame.
minutes_date = pd.merge(temp_76, date,  how='left',
 ↪  left_on=['meeting'], right_on = ['meeting'])

#We change the format the of the 'date' column from object
 ↪  to datetime64[ns].
minutes_date['date'] = pd.to_datetime(minutes_date['date'],
 ↪  infer_datetime_format=True,dayfirst=True)

#Checking data format of 'minutes_date' DataFrame.
minutes_date.dtypes

#We create the uncertainty score variable ('score') by
 ↪  dividing the total number of uncertain words
 ↪  (minutes_date['UncerScore'] ) by the total number of
 ↪  words per minute (minutes_date['TotalWordCount']).
minutes_date['score'] = minutes_date['UncerScore'] /
 ↪  minutes_date['TotalWordCount']

#We construct the normalized uncertainty index with mean
 ↪  100.
minutes_date['uncertainty_normalized'] = (100 *
 ↪  minutes_date['score']) / minutes_date["score"].mean()

#Creating copy of the 'minutes_date' DataFrame as
 ↪  'df_general' DataFrame.
df_general = minutes_date.copy()

#We create new columns in the 'df_general' DataFrame with
 ↪  the values of the year, the month and the day of the
 ↪  column 'date'.
df_general['year'] = df_general['date'].dt.year
df_general['month'] = df_general['date'].dt.month
df_general['day'] = df_general['date'].dt.day

#We change the 'day' column values to 1 since we are merely
 ↪  interested in monthly observations to compare it with
 ↪  the EPU index.
```

```
df_general['day'] = 1

#We create  column 'date' with the values of the columns
 ↪  'year', 'month' and 'day'.
df_general['date'] = pd.to_datetime(df_general[["year",
 ↪  "month", "day"]])

#We load the EPU index of Brazil of  Baker, Bloom and David
 ↪  (2016).
baker = pd.read_csv("baker.csv", sep = ';', encoding =
 ↪  "utf-8")

#We create a new column in the 'baker' DataFrame with the
 ↪  same values of the column 'Brazil News-Based EPU' but
 ↪  with a simplier name.
baker['epu'] =  baker['Brazil News-Based EPU'].copy()

#We change the format of the 'date' column from object to
 ↪  datetime64[ns].
baker['date'] = pd.to_datetime(baker['date'],
 ↪  infer_datetime_format=True,dayfirst=True)

#We check the format of the 'baker' DataFrame.
baker.dtypes

#We merge the 'df_general' DataFrame to the 'baker'
 ↪  DataFrame in a new DataFrame named 'graph_general'.
graph_general = pd.merge(df_general, baker,  how='outer',
 ↪  left_on=['date'], right_on = ['date'])

#We sort the values of the 'graph_general' DataFrame  by
 ↪  date.
graph_general = graph_general.sort_values(by=['date'])

#We delete all the observations that took place before
 ↪  December 1999.
graph_general =
 ↪  graph_general[~graph_general['date'].isin(pd.date_range(
 ↪  start ='1991-01-01', end='1999-11-01'))]

```

```
#We delete all the observations that took place after
  ↪  September 2019.
graph_general =
  ↪  graph_general[~graph_general['date'].isin(pd.date_range(
  ↪  start ='2019-10-01', end='2019-10-01'))]

#Filling empty values with the previous value of the
  ↪  uncertainty index column (['uncertainty_normalized']).
graph_general['uncertainty_normalized'] =
  ↪  graph_general['uncertainty_normalized'].fillna(method=
  ↪  'ffill')

#Normalizing the EPU index for Brazil with mean 100.
graph_general['epu_normalized'] = (100 *
  ↪  graph_general['epu']) / graph_general["epu"].mean()

#Setting the 'date' column as index of the 'graph_general'
  ↪  DataFrame.
graph_general = graph_general.set_index('date')
graph_general.head(3)

#########################################################
##### Construction of the topic-uncertainty indexes #####
#########################################################

#Grouping the number of uncertainty words and the  total
  ↪  number of words by minutes and LDA group of topics.
temp_topic = df.groupby(['year', 'meeting','topic'])
  ↪  ['TotalWordCount','UncerScore'
  ↪  ].sum().reset_index().rename(columns={
  ↪  'CombScore':'combsum'})

#Creating a copy of the 'temp_topic' DataFrame with the
  ↪  name 'temp_top'.
temp_top = temp_topic.copy()

#Creating the 'general economic conditions' DataFrame  as
  ↪  'topic_gec' with the paragraphs related to its topics.
topic_gec = temp_top[temp_top.topic == 0]

```

```
#Creating the 'inflation and monetary policy decision'
↪  DataFrame as 'topic_copom' with the paragraphs related
↪  to its topics.
topic_copom = temp_top[temp_top.topic == 1]

#Resetting index of the 'general economic conditions'
↪  DataFrame.
topic_gec = topic_gec.reset_index()

#Resetting index of the 'inflation and monetary policy
↪  decision' DataFrame.
topic_copom = topic_copom.reset_index()

#########################################
## Construction of the 'general economic conditions'
↪  topic-uncertainty index ##
########################################
#The meeting 76 and 77 occur in the same month. Thus, we
↪  join them in the same observation.
topic_gec.loc[34] += topic_gec.loc[35]

#We drop the row 35 which corresponds to the minute of the
↪  meeting 77.
topic_gec.drop([35], inplace=True)

#We change the values of the row of the minute number 76
↪  that should not change as year or meeting.
topic_gec.at[34, 'year'] = 2002
topic_gec.at[34, 'meeting'] = 76

#We merge the 'general economic conditions' DataFrame with
↪  the 'date' DataFrame in a new DataFrame called
↪  'minutes_gec'.
minutes_gec = pd.merge(topic_gec, date,  how='left',
↪  left_on=['meeting'], right_on = ['meeting'])

#We change the format of the 'date' column from object to
↪  datetime64[ns].
```

```
minutes_gec['date'] =
 ↪  pd.to_datetime(minutes_gec['date'],infer_datetime_format
 ↪  = True, dayfirst=True)

#Checking the data format of the 'minutes_gec' DataFrame.
minutes_gec.dtypes

#We create the uncertainty score variable
 ↪  (minutes_gec['score']) by dividing the total number of
 ↪  uncertain words (minutes_gec['UncerScore']) by the
 ↪  total number of words per minute
 ↪  (minutes_gec['TotalWordCount']).
minutes_gec['score'] = minutes_gec['UncerScore'] /
 ↪  minutes_gec['TotalWordCount']

#We create the normalized  'general economic conditions'
 ↪  topic-uncertainty index with mean 100.
minutes_gec['uncertainty_normalized'] = (100 *
 ↪  minutes_gec['score']) / minutes_gec["score"].mean()

#We create a copy of the DataFrame 'minutes_gec' with the
 ↪  name 'df_gec'.
df_gec = minutes_gec.copy()

#We create new columns in the DataFrame 'df_gec' with the
 ↪  values of the year, the month and the day of the column
 ↪  'date'.
df_gec['year'] = df_gec['date'].dt.year
df_gec['month'] = df_gec['date'].dt.month
df_gec['day'] = df_gec['date'].dt.day

#We change the day column values to  1 since we are merely
 ↪  interested in monthly observations in order to be able
 ↪  to compare it with the EPU index.
df_gec['day'] = 1

#We create the column 'date' with the values of the columns
 ↪  'year', 'month' and 'day'.
df_gec['date'] = pd.to_datetime(df_gec[["year", "month",
 ↪  "day"]])
```

```

#We merge the 'df_gec' DataFrame to the 'baker' DataFrame
 ↪  in a new DataFrame named 'graph_gec'.
graph_gec = pd.merge(df_gec, baker,  how='outer',
 ↪  left_on=['date'], right_on = ['date'])

#We sort the values of the 'graph_gec' DataFrame  by date.
graph_gec = graph_gec.sort_values(by=['date'])

#We delete all the observations that took place before
 ↪  December 1999.
graph_gec =
 ↪  graph_gec[˜graph_gec['date'].isin(pd.date_range(
 ↪  start='1991-01-01', end='1999-11-01'))]

#We delete all the observations that took place after
 ↪  September 2019.
graph_gec = graph_gec[˜graph_gec['date'].isin(
 ↪  pd.date_range( start='2019-10-01', end='2019-10-01'))]

#Filling empty values with the previous value of the
 ↪  'general economic conditions' topic-uncertainty index
 ↪  column (graph_gec['uncertainty_normalized']).
graph_gec['uncertainty_normalized'] =
 ↪  graph_gec['uncertainty_normalized'].fillna(method='ffill')

#Setting the 'date' column as index of the 'graph_gec'
 ↪  DataFrame.
graph_gec = graph_gec.set_index('date')
graph_gec.head(3)


########################################
# Construction of the 'inflation and monetary policy
 ↪  decision' topic-uncertainty index #
########################################
#The meeting 76 and 77 occurred in the same month. Thus, we
 ↪  join them in the same observation.
topic_copom.loc[34] += topic_copom.loc[35]

```

```python
#We drop the row 35 which corresponds to the minute of the meeting 77.
topic_copom.drop([35], inplace=True)

#We change the values of the row of the minute number 76 that should not change as year or meeting.
topic_copom.at[34, 'year'] = 2002
topic_copom.at[34, 'meeting'] = 76
topic_copom.at[34, 'topic'] = 1

#We merge the 'inflation and monetary policy decision' DataFrame with the 'date' DataFrame in a new DataFrame called 'minutes_copom'.
minutes_copom = pd.merge(topic_copom, date,  how='left', left_on=['meeting'], right_on = ['meeting'])

#We change the format of the 'date' column from object to datetime64[ns].
minutes_copom['date'] = pd.to_datetime(minutes_copom['date'], infer_datetime_format=True,dayfirst=True)

#Checking the data format of the 'minutes_copom' DataFrame.
minutes_copom.dtypes

#We create the uncertainty score variable (minutes_copom['score']) by dividing the total number of uncertain words (minutes_copom['UncerScore']) by the total number of words per minute (minutes_copom['TotalWordCount']).
minutes_copom['score'] = minutes_copom['UncerScore'] / minutes_copom['TotalWordCount']

#We create the normalized  'general economic conditions' topic-uncertainty index with mean 100.
minutes_copom['uncertainty_normalized'] = (100 * minutes_copom['score']) / minutes_copom["score"].mean()

#We create a copy of the DataFrame 'minutes_copom' with the name 'df_copom'.
```

```
df_copom = minutes_copom.copy()

#We create new columns in the DataFrame 'df_copom' with the
 ↪  values of the year, the month and the day of the column
 ↪  'date'.
df_copom['year'] = df_copom['date'].dt.year
df_copom['month'] = df_copom['date'].dt.month
df_copom['day'] = df_copom['date'].dt.day

#We change the 'day' column values to  1 since we are
 ↪  merely interested in monthly observations in order to
 ↪  be able to compare it with the EPU index.
df_copom['day'] = 1

#We create the column 'date' with the values of the columns
 ↪  'year', 'month' and 'day'.
df_copom['date'] = pd.to_datetime(df_copom[["year",
 ↪  "month", "day"]])

#We merge the 'df_copom' DataFrame to the 'baker' DataFrame
 ↪  in a new DataFrame named 'graph_copom'.
graph_copom = pd.merge(df_copom, baker,  how='outer',
 ↪  left_on=['date'], right_on = ['date'])

#We sort the values of the 'graph_copom' DataFrame by date.
graph_copom = graph_copom.sort_values(by=['date'])

#We delete all the observations that took place before
 ↪  December 1999.
graph_copom = graph_copom[~graph_copom['date'].isin(
 ↪  pd.date_range( start='1991-01-01', end='1999-11-01'))]

#We delete all the observations that took place after
 ↪  September 2019.
graph_copom = graph_copom[~graph_copom['date'].isin(
 ↪  pd.date_range( start='2019-10-01', end='2019-10-01'))]

```

```
#Filling empty values with the previous value of the
  ↪  'inflation and monetary policy decision'
  ↪  topic-uncertainty index column
  ↪  (graph_copom['uncertainty_normalized']).
graph_copom['uncertainty_normalized'] =
  ↪  graph_copom['uncertainty_normalized'].fillna(
  ↪  method='ffill')

#Setting the 'date' column as index of the 'graph_copom'
  ↪  DataFrame.
graph_copom = graph_copom.set_index('date')
graph_copom.head(3)


##########################################
# Graph uncertainty index and EPU index #
##########################################

graph_general['epu_normalized'].plot(color='orange')
graph_general['uncertainty_normalized'].plot(color='green')
plt.ylabel("Uncertainty index (Mean = 100)")
plt.xlabel("Minutes across time")
axvline('2001-05-23', color='red', ls="dotted")
axvline('2003-01-22', color='blue', ls="dotted")
axvline('2003-04-23', color='red', ls="dotted")
axvline('2005-09-14', color='red', ls="dotted")
axvline('2011-01-19', color='blue', ls="dotted")
axvline('2014-02-26', color='red', ls="dotted")
axvline('2016-07-20', color='black', ls="dotted")
axvline('2019-03-20', color='blue', ls="dotted")
orange_patch = mpatches.Patch(color='orange', label='EPU
  ↪  uncertainty index')
green_patch = mpatches.Patch(color='green', label='Minutes
  ↪  uncertainty index')
plt.legend(handles=[orange_patch, green_patch],loc='center
  ↪  left', bbox_to_anchor=(0, 0.95))

##################################################
# Graph topic-uncertainty indexes and EPU index #
##################################################
```

```

graph_general['epu_normalized'].plot(color='orange')
graph_gec['uncertainty_normalized'].plot(color='red')
graph_copom['uncertainty_normalized'].plot(color='blue')
plt.ylabel("Uncertainty index (Mean = 100)")
plt.xlabel("Minutes across time")
axvline('2001-05-23', color='red', ls="dotted")
axvline('2003-01-22', color='blue', ls="dotted")
axvline('2003-04-23', color='red', ls="dotted")
axvline('2005-09-14', color='red', ls="dotted")
axvline('2011-01-19', color='blue', ls="dotted")
axvline('2014-02-26', color='red', ls="dotted")
axvline('2016-07-20', color='black', ls="dotted")
axvline('2019-03-20', color='blue', ls="dotted")
blue_patch = mpatches.Patch(color='orange', label='EPU
 ↪  uncertainty index')
red_patch = mpatches.Patch(color='red', label='General
 ↪  economic conditions topic-uncertainty index')
green_patch = mpatches.Patch(color='blue', label='Inflation
 ↪  and monetary policy decision topic-uncertainty index')
plt.legend(handles=[blue_patch, red_patch,
 ↪  green_patch],loc='center left', bbox_to_anchor=(0,
 ↪  0.93))

#####################################################
# Construction of excel database for Structural VAR #
#####################################################

#Creating new columns for the uncertainty and topic
 ↪  uncertainty indexes variables with new names.
graph_general['uncertainty_general'] =
 ↪  graph_general['uncertainty_normalized'].copy()
graph_gec['uncertainty_gec'] =
 ↪  graph_gec['uncertainty_normalized'].copy()
graph_copom['uncertainty_copom'] =
 ↪  graph_copom['uncertainty_normalized'].copy()

#Merging DataFrames  'graph_general' and 'graph_gec'.
svar1 = pd.merge(graph_general, graph_gec,  how='left',
 ↪  left_on=['date'], right_on = ['date'])
```

```python

#Creating new columns for the year and month variables with new names.
svar1['yeear'] = svar1['year_y_x']
svar1['moonth'] = svar1['month_y_x']

#Merging DataFrames 'svar1' and 'graph_copom' in a new DataFrame named 'svar2'.
svar2 = pd.merge(svar1, graph_copom,  how='left', left_on=['date'], right_on = ['date'])

#Creating new columns for the year, month, day and EPU variables with new names.
svar2['meeting'] = svar2['meeting_x']
svar2['year'] = svar2['year_x']
svar2['epu'] = svar2['epu_x']
svar2['day'] = svar2['day_x']
svar2['day'] = 1

#Creating DataFrame 'svar_min' only with the relevant variables of the DataFrame 'svar2'.
svar_min = svar2[['meeting','yeear','moonth','day', 'uncertainty_general','epu','uncertainty_gec', 'uncertainty_copom']].copy()

#Saving in csv the DataFrame 'svar_min'.
svar_min.to_csv("brazil_database_structuralvar_2019.csv")
```

2.4 Structural VAR Model

2.4.1 Description of the macroeconomic database

To analyze the relationship between the uncertainty indices and the real economy, we download a group of macroeconomic variables from the Federal Reserve Bank of St. Louis aka FRED database. We download four variables which are saved in the excel file 'brasil_macro_2019.csv' and they are described in the following list:

1. Industrial production (Series ID: BRAPROINDMISMEI; Title: production of total industry in Brazil; Units: index 2015 = 100; Frequency = monthly; Seasonal adjustment = seasonally adjusted; Excel tag = indpro).

2. Retail (Series ID: BRASARTMISMEI; Title: total retail trade in Brazil; Units: index 2015 = 100; Frequency = monthly; Seasonal adjustment = seasonally adjusted; Excel tag = retail).

3. CPI (Series ID: CPALTT01BRM659N; Title: consumer price index: total all items for Brazil; Units: growth rate same period previous year; Frequency = monthly; Seasonal adjustment = not seasonally adjusted; Excel tag = cpi).

4. Exchange rate (Series ID: RBBRBIS; Title: real broad effective exchange rate for Brazil; Units: index 2010=100; Frequency = monthly; Seasonal adjustment = not seasonally adjusted; Excel tag = Real_broad _exch_rate).

2.4.2 Merging of macroeconomic database and uncertainty indices database

We merge the macroeconomic database (brasil_macro_2019.csv) with the uncertainty indices database (‘brazil_database_structuralvar_2019.csv’) to create an unified database for stata (‘brazil_sva_macro _ui.xlsx’). The python code (‘merging_database _brazil_svar.py’) to create the merged database is the following:

```python
import pandas as pd

#We import the 'uncertainty' database as 'unc' DataFrame.
unc = pd.read_csv("brazil_database_structuralvar_2019.csv", sep = ',', encoding = "utf-8")

#We normalize the EPU index with mean = 100.
unc['epu_normalized'] = (100 * unc['epu']) / unc["epu"].mean()

#We create new variables to rename the varaibles 'year' and 'month'.
unc['year'] = unc['yeear']
unc['month'] = unc['moonth']

#Creating 'date' column with the values of the columns 'year', 'month' and 'day'.
unc['date'] = pd.to_datetime(unc[["year", "month", "day"]])

#We change the format of the 'date' column from object to datetime64[ns].
```

```
unc['date'] = pd.to_datetime(unc['date'],
 ↪  infer_datetime_format=True, dayfirst=True)

#Loading the macroeconomic database as 'macro' DataFrame.
macro = pd.read_csv("brasil_macro_2019.csv", sep = ';',
 ↪  encoding = "utf-8")

#We change the format the column 'date' from object to
 ↪  datetime64[ns].
macro['date'] = pd.to_datetime(macro['date'],
 ↪  infer_datetime_format=True, dayfirst=True)

#Merging the 'unc' and 'macro' DataFrames.
svar = pd.merge(unc, macro,  how='left', left_on=['date'],
 ↪  right_on = ['date'])

#Saving 'svar' DataFrame as excel.
svar.to_excel("brazil_svar_macro_ui.xlsx")
```

2.4.3 Structural VAR: estimation

We estimate several Structural VAR models to understand the relationship between the real economy and the uncertainty indices. The stata code for these estimations is included in the complementary material folder with the name ‘Brazil_SVAR_impulse-response.do’. The database with the macroeconomic and uncertainty data is passed from excel format to dta format with the name ‘brazil_svar_macro_ui.dta’. Below, we show the stata code to estimate Structural VAR. We show the stata code to construct the impulse response functions of a rise in one standard shock in the uncertainty index.

```
*Setting date index from December 1999.
gen daate = m(1999m12) + _n - 1
format %tm daate
tsset daate

*Descriptive statistics between Decemeber 1999 and June
 ↪  2019.
summarize uncertainty_general epu uncertainty_gec
 ↪  uncertainty_copom oecd_gdp retail cpi
 ↪  real_broad_exch_rate if daate>=tm(1999m12) &
 ↪  daate<=tm(2019m6)
```

```
*Creating differentiated variables.
gen d_uncgen = uncertainty_general - uncertainty_general[_n-1]
gen d_epu = epu_normalized - epu_normalized[_n-1]
gen d_uncgec = uncertainty_gec - L.uncertainty_gec
gen d_unccopom   = uncertainty_copom - L.uncertainty_copom
gen d_indpro = indpro - L.indpro
gen d_retail = retail - L.retail
gen d_cpi = cpi - L.cpi
gen d_exch = real_broad_exch_rate - L.real_broad_exch_rate

*We drop observations before December 1999.
drop if daate <= tm(1999m12)

*We drop observations after June 2019.
drop if daate > tm(2019m6)

*We check if our variables pass the Dickey Fuller.
dfuller d_uncgen
dfuller d_cpi
dfuller d_exch
dfuller d_indpro
dfuller d_retail

*Then, we define the Cholesky restrictions.
matrix A = (1,0,0,0,0\.,1,0,0,0\.,.,1,0,0\.,.,.,1,0\.,.,.,.,1)
matrix B = (.,0,0,0,0\0,.,0,0,0\0,0,.,0,0\0,0,0,.,0\0,0,0,0,.)

****************************************
*Estimation of SVAR with minutes uncertainty index from 2000 until June 2019 *
****************************************

```

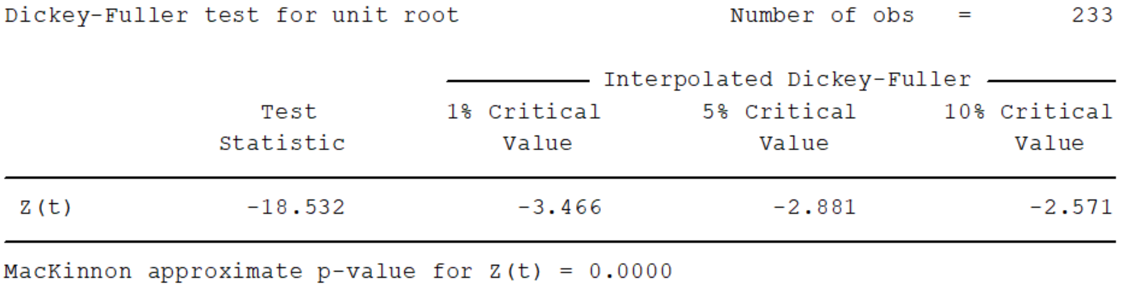
```
*The varsoc test reports the final prediction error (FPE),
  ↪ Akaike's information criterion (AIC), Schwarz's
  ↪ Bayesian information criterion (SBIC), and the Hannan
  ↪ and Quinn information criterion (HQIC) lag order
  ↪ selection statistics.
varsoc d_uncgen d_exch d_cpi d_indpro d_retail if
  ↪ daate>=tm(2000m1), lutstats

*Estimation of the SVAR model for the minutes uncertainty
  ↪ index from 2000 until June 2019.
svar d_uncgen d_exch d_cpi  d_indpro d_retail if
  ↪ daate>=tm(2000m1), dfk aeq(A) beq(B) lags(1)
matrix Aest = e(A)
matrix Best = e(B)
matrix chol_est = inv(Aest)*Best
matrix list chol_est
matrix sig_var = e(Sigma)
matrix chol_var = cholesky(sig_var)
matrix list chol_var

*varnorm reports the Jarque-Bera statistic.
varnorm

*varlmar reports the Lagranger-Multiplier test for residual
  ↪ autocorrelation after SVAR.
varlmar, mlag(5)

*varstable indicates the eigenvalue stability conditions.
varstable

*Impulse response functions from the Structural VAR model
  ↪ corresponding to one standard-deviation in the minutes
  ↪ uncertainty index in exchange rate and inflation for
  ↪ the period 2000 - June 2019.
irf create order1, step(8) set(myirf1)
irf graph oirf, impulse(d_uncgen) response(d_exch d_cpi)
  ↪ subtitle("") plot1opts(lcolor(red))
  ↪ byopts(legend(off)) byopts(graphregion(color(white)))
  ↪ byopts(bgcolor(white))    byopts(note("")) xtitle("")

```

```
*Impulse response functions from the Structural VAR model corresponding to one standard-deviation in the minutes uncertainty index in industrial production and  retail for the period 2000 - June 2019.
irf create order1, step(8) set(myirf2)
irf graph oirf, impulse(d_uncgen) response(d_indpro d_retail) subtitle("") plot1opts(lcolor(red)) byopts(legend(off)) byopts(graphregion(color(white))) byopts(bgcolor(white))  byopts(note("")) xtitle("")
```

2.4.4 Structural VAR: measures of goodness of fit

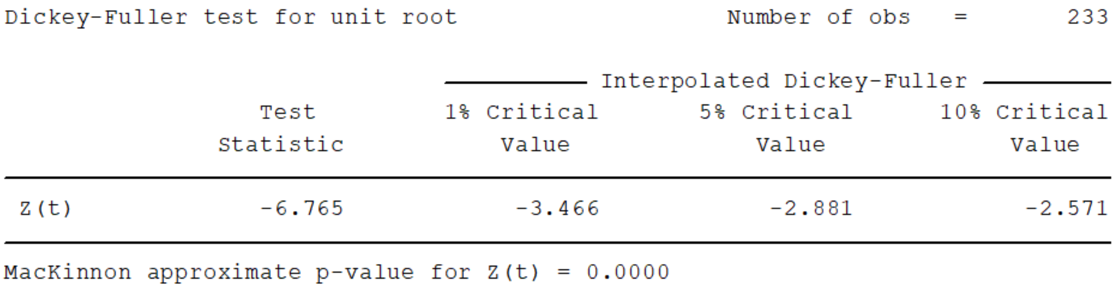
This section shows the results of the measures of goodness of fit that are not included in the paper.

All variables are differentiated to overcome the non-stationary problem in light of the augmented Dickey-Fuller test indicating I(1). From Figure 7 to Figure 11, we check if the difference variables pass the Dickey Fuller test. All the difference variables are stationary or I(1).



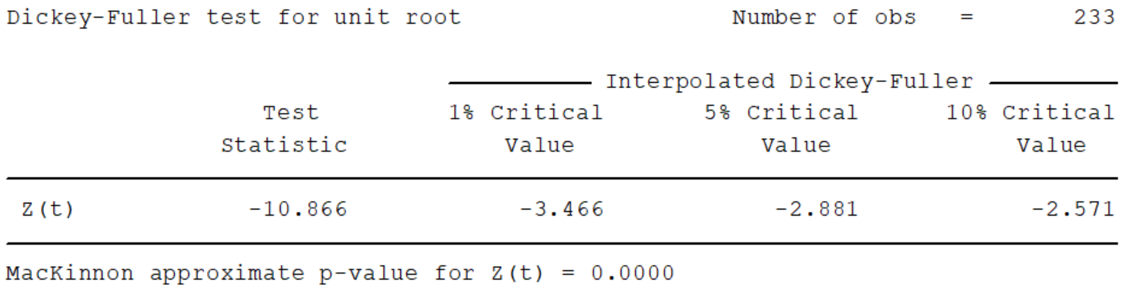
```
Dickey-Fuller test for unit root                   Number of obs   =       233

                               ———— Interpolated Dickey-Fuller ————
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
──────────────────────────────────────────────────────────────────────────────
 Z(t)           -18.532            -3.466            -2.881            -2.571
──────────────────────────────────────────────────────────────────────────────
MacKinnon approximate p-value for Z(t) = 0.0000
```

Figure 7: Dickey-Fuller test for unit root for the difference of the minutes uncertainty index.



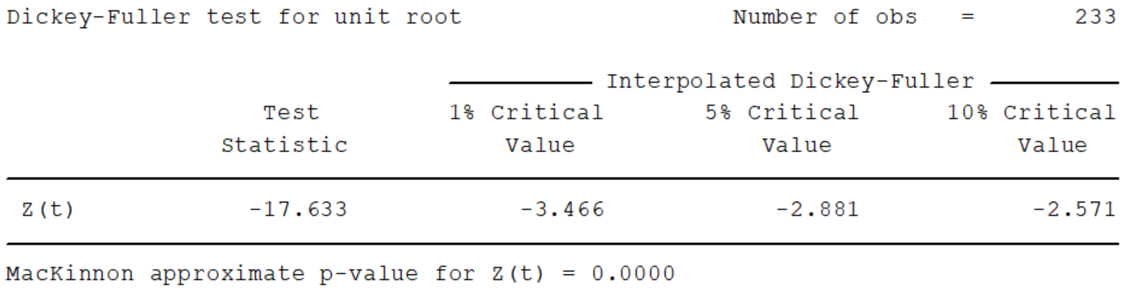
```
Dickey-Fuller test for unit root                   Number of obs   =       233

                               ———— Interpolated Dickey-Fuller ————
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
──────────────────────────────────────────────────────────────────────────────
 Z(t)            -6.765            -3.466            -2.881            -2.571
──────────────────────────────────────────────────────────────────────────────
MacKinnon approximate p-value for Z(t) = 0.0000
```

Figure 8: Dickey-Fuller test for unit root for the difference of the consumer price index.



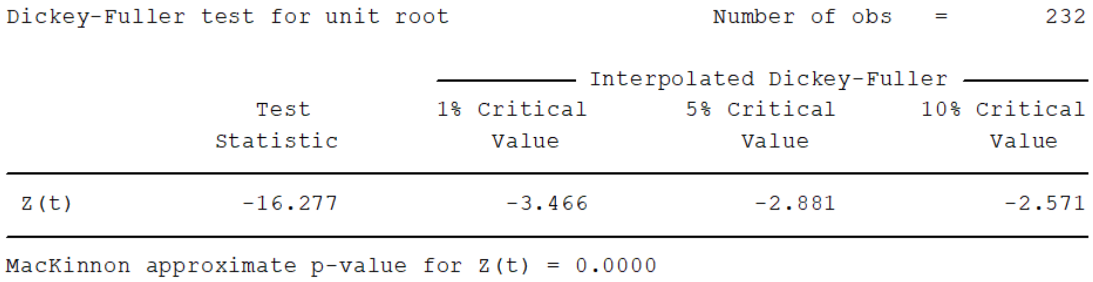
```
Dickey-Fuller test for unit root                   Number of obs   =       233

                           ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)           -10.866            -3.466            -2.881            -2.571
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0000
```

Figure 9: Dickey-Fuller test for unit root for the difference of the exchange rate.



```
Dickey-Fuller test for unit root                   Number of obs   =       233

                           ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)           -17.633            -3.466            -2.881            -2.571
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0000
```

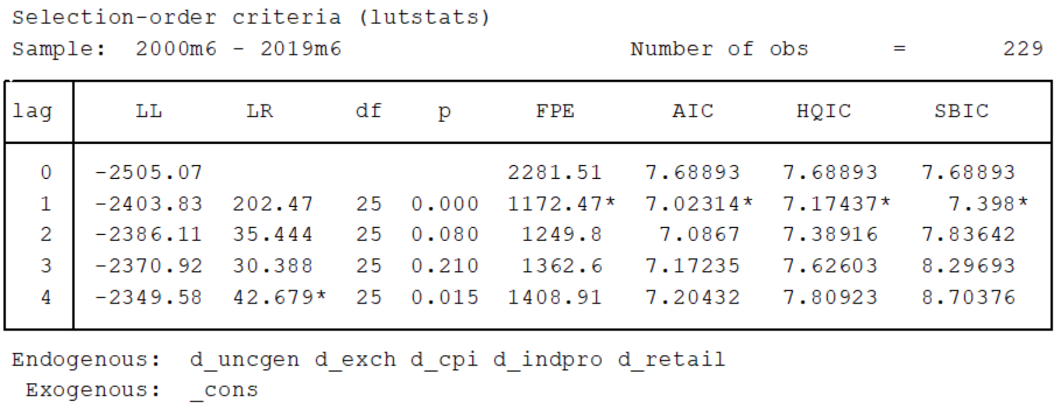
Figure 10: Dickey-Fuller test for unit root for the difference of industrial production.



```
Dickey-Fuller test for unit root                   Number of obs   =       232

                           ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)           -16.277            -3.466            -2.881            -2.571
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0000
```

Figure 11: Dickey-Fuller test for unit root for the difference of retail.

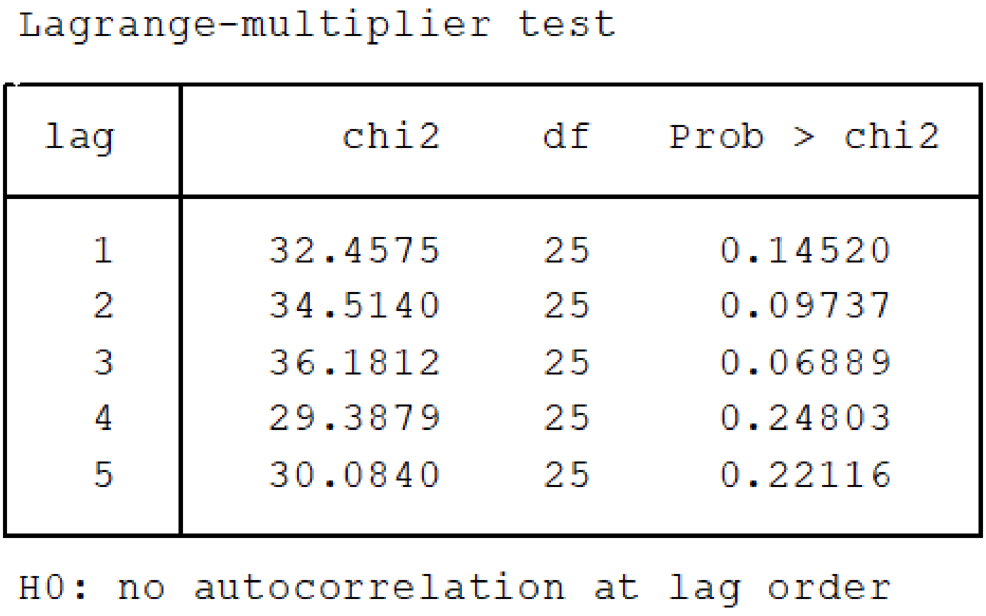
Figure 12 shows the results of the varsoc test that reports the final prediction error (FPE), Akaike's information criterion (AIC), Schwarz's Bayesian information criterion (SBIC) and the Hannan and Quinn information criterion (HQIC) lag order selection statistics. The optimal number of lags is one according to AIC, SBIC, HQIC and FPE.



```
Selection-order criteria (lutstats)
Sample:  2000m6 - 2019m6                         Number of obs      =       229

lag      LL         LR       df     p       FPE        AIC        HQIC       SBIC

 0   -2505.07                             2281.51   7.68893    7.68893    7.68893
 1   -2403.83   202.47     25   0.000   1172.47*  7.02314*   7.17437*     7.398*
 2   -2386.11   35.444     25   0.080    1249.8     7.0867    7.38916    7.83642
 3   -2370.92   30.388     25   0.210    1362.6    7.17235    7.62603    8.29693
 4   -2349.58   42.679*    25   0.015   1408.91    7.20432    7.80923    8.70376

Endogenous:  d_uncgen d_exch d_cpi d_indpro d_retail
 Exogenous:  _cons
```

Figure 12: Final prediction error (FPE), Akaike's information criterion (AIC), Schwarz's Bayesian information criterion (SBIC), and the Hannan and Quinn information criterion (HQIC) lagorder selection statistics.

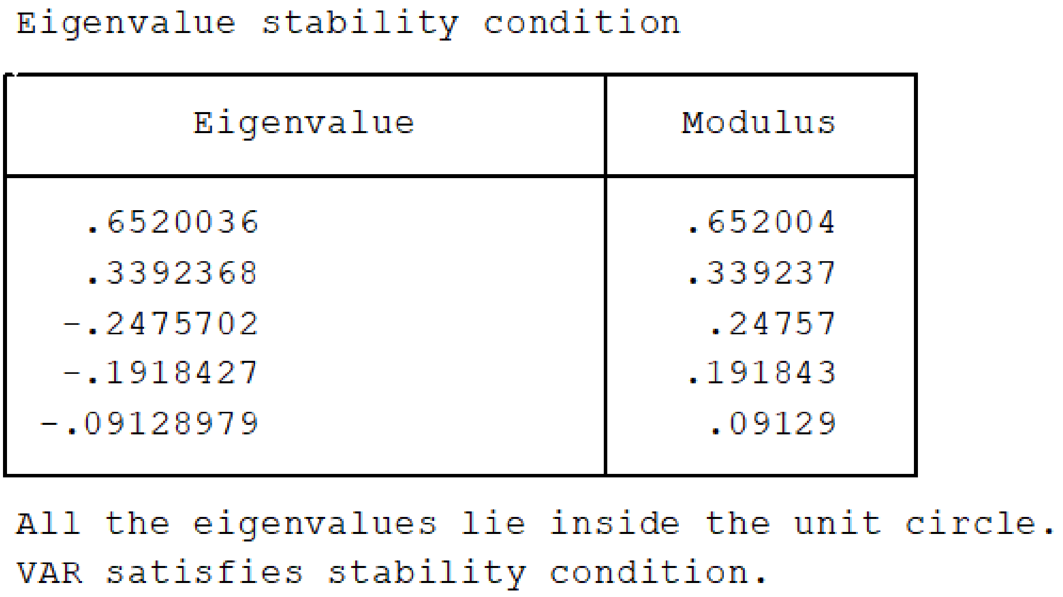
The following two figures show the tests of the Structural VAR model corresponding to one standard-deviation in the uncertainty index. Figure 12 shows the output of the Lagrange multiplier test. We do not reject the null hypothesis, meaning there is not autocorrelation in the residuals for four of the lags tested. However, it is rejected for the third lag.



```
Lagrange-multiplier test

 lag          chi2      df    Prob > chi2

   1       32.4575      25      0.14520
   2       34.5140      25      0.09737
   3       36.1812      25      0.06889
   4       29.3879      25      0.24803
   5       30.0840      25      0.22116

H0: no autocorrelation at lag order
```

Figure 13: Lagrange multipier test

Our Structural VAR results comply with the stability condition since all roots of the characteristic polynomial are outside of the unit circle.



```
Eigenvalue stability condition

  Eigenvalue        Modulus

   .6520036         .652004
   .3392368         .339237
  -.2475702          .24757
  -.1918427         .191843
 -.09128979          .09129

All the eigenvalues lie inside the unit circle.
VAR satisfies stability condition.
```

Figure 14: Eigen value stability condition

Chapter 3

Monetary Policy Uncertainty in Mexico: An Unsupervised Approach

3.1 Introduction

Nowadays, to prevent monetary policy serving political interests, in particular in order to finance the public deficit (as in part of the 70s and the 80s when the Central Bank of Mexico printed money to finance the Mexican public debt, leading to high inflation), most central banks are independent and their communications are an important part of their policy. Independent central banks are asked to maintain a high level of transparency in their communications to guarantee the accountability of their decisions. In particular, central bank communications help markets to take action in advance of future changes in key monetary policy variables such as interest rates or money supply.

During the 90s and early 2000s, several Latin American central banks - in Brazil, Colombia, Chile, Mexico and Peru - adopted an inflation targeting system with the aim of reducing and controlling inflation. The inflation targeted monetary approach in these Latin American countries included the publication of inflation reports, the creation of mid-term inflation targets and improved communications with the markets (Taborda, 2015). Since then, several authors have investigated the communications of Latin American central banks and their effect on the markets. For instance, Costa-Fiho and Rocha (2010), Cabral and Guimaraes (2015), Garcia-Herrero, Girandin and Dos Santos (2017) study how the communication of the Central Bank of Brazil changes interest rate expectations. In all these works, the authors manually process Central Bank of Brazil communication to infer if the communication is dovish or hawkish. Other authors have investigated the communication of the Central Banks of Chile and Colombia. They include Garcia-Herrero, Girardin and Gonzalez (2017) and Ciro, Camilo and Anzoátegui-Zapata (2019). The communication of the Central Bank of Mexico has been investigated, by Herrerias and Gurrola (2012) among others.

This paper investigates and creates text uncertainty measures for the minutes of the meetings of the board of governors of the Central Bank of Mexico. The board of governors of the Central Bank of Mexico (aka Bank of Mexico or Banxico) meets eight times a year to set the interest rate. Since 2011, the minutes have been published two weeks after the meetings. The minutes provide in-depth information on the meetings of the board of governors that is not provided by the initial statements regarding the monetary policy decision.

In the literature, investigations take different approaches to obtain measures from text. Some authors use dictionary methods, i.e. predefined lists of words related to a sentiment such as uncertainty. They count the relative frequency of the words in the dictionary in the text to create a sentiment index, such as an uncertainty index. Some of the most common English language dictionaries used in economic research are the Loughran and McDonald (2011) and Harvard IV-4 Psychological dictionaries. For instance, Shapiro et al. (2019) apply the Loughran and McDonald (2011) dictionary to the transcripts of the meetings of the Federal Open Market Committee (FOMC) to investigate its loss function. Nonetheless, dictionary methods can include some bias since the words of the dictionary may not fit the words of the text. Some authors such as Bernal and Pedraz (2020) try to overcome this issue by constructing their own dictionaries. These authors manually created the first positive, negative and neutral word dictionary in Spanish for financial stability from Financial Stability Reports of the Bank of Spain from 2002 to 2019. Other authors such as Ghirelli, Pérez and Urtasun (2019) have built an economic policy uncertainty index for Spain from Spanish newspapers, improving the methodology of Baker, Bloom and David (2016). With a VAR model, Ghirelli, Pérez and Urtasun (2019) estimate the effect of their uncertainty index on GDP, consumption and investment.

Machine learning techniques attempt to improve on the construction of text measures. We distinguish between supervised and unsupervised machine learning techniques. Supervised machine learning techniques use a set of input variables (X) to predict an output variable (Y). For instance, Manela and Moreira (2017) use Support Vector Machines, a supervised machine learning algorithm, to create a news-based measure of implicit volatility from news in the Wall Street Journal from 1890 to 2009.

Unsupervised machine learning tries to find meaningful relationships among the input data (X) without relying on any output (Y). Some investigations use unsupervised machine learning techniques for topic analysis. They included Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) and Dynamic Topic Model (DTM). These techniques consist in joining words in groups of similar themes or topics. For instance, if we apply these techniques to a newspaper, we obtain topics that are related to the different sections of the newspaper such as politics, economics, fashion, cooking, or sports. Some authors such as Arango, Pantoja, and Velasquez (2017) apply Latent Semantic Analysis to analyze the communications of the Central Bank of Colombia. They use a Structural VAR to measure the effect of the weights of the topics in break-even-inflation expectations, the economic situation indicator, and the inter-bank interest rate. Additionally, Ortiz et al. (2017) use Dynamic Topic Model together with dictionary methods to analyze the effect of the communications of the Central Bank of Turkey on the financial market and the real economy. Finally, Latent Dirichlet Allocation (LDA) is an unsupervised machine learning algorithm for topic analysis which consists in a generative probabilistic model of a body of text. The basis of LDA is that documents are depicted as random combinations of latent topics, where each topic is represented by a distribution over words (Blei et al, 2003). Some authors such as Azqueta-Gavaldon (2017) apply LDA to create an uncertainty index by counting the number of articles in which one of the topics related to uncertainty have the highest probability. Other authors like Bybee et al. (2020) apply LDA to 800,000 Wall Street Journal articles from 1984 to 2017. These authors apply a Structural VAR model to explore how higher attention to the topic related to recession is linked to a decrease in industrial production and unemployment. Additionally, other papers such as Thorsrud (2016) use topics from newspaper data to increase macroeconomic forecasting. Other investigations such as Hansen, McMahon, and Prat (2017) use LDA and dictionary methods to study the effect of transparency on the decisions of the Federal Open Market Committee (FOMC). Moreover, several papers in the literature use LDA to study central bank communications. They include Hansen, McMahon and Tong (2019). Following Zou and Hastie (2005), these authors use Elastic Net to identify the topics in the Bank of England inflation report with the strongest predictive power.

Some papers also use various unsupervised machine learning algorithms such as the Skip-Gram model, introduced by Mikolov et al. (2013a), and Mikolov et al. (2013b). The main output of the Skip-Gram model comprises Word Embeddings, continuous vector representations of words that preserve the syntactical and semantic similarities between words in a Euclidean Space. In economics, the Word Embeddings are used for sentiment analysis since they reveal the most similar words to a given word. Thus, researchers can create their own dictionaries related to a sentiment with their own corpus in an automatic way instead of depending on predetermined dictionaries that might not be suitable. The Skip-Gram model also provides cheap and fast text classification compared to manual classification, which is time consuming and normally quite expensive, requiring researchers to be hired to classify the text. There is a shortage of economics literature on the Skip-Gram method. Soto (2021) investigates how commercial banks communicate in their quarterly conference calls. After computing the Skip-gram model, Soto (2021) uses K-Means to find the nearest word vectors to the vector representations of ‘uncertainty’ and ‘uncertain’ and constructs a list of uncertain words. He then uses the frequency of these words in the different documents to create an uncertainty index, later applying LDA and combining the topic weight results of LDA with the uncertainty index to create topic-uncertainty indices.

To the best of our knowledge, this is the first paper to apply unsupervised machine learning techniques to construct text measures from the Spanish version of the communications of the Central Bank of Mexico. To understand the content or theme, we apply Latent Dirichlet Allocation to the minutes of the meetings of the Bank of Mexico board of governors from 2011 to 2018. The first LDA output shows the probability of words across topics. Our results show that the words in topic 5 have a similar meaning to the words 'uncertainty' and 'risk'. We use the probability of topic 5 in the minutes to build an uncertainty index and call it the LDA uncertainty index. The second contribution of this paper is to process another uncertainty index for the minutes applying the Skip-Gram model and K-Means, following Soto (2021). The Skip-Gram and K-Means results provide a list of words (dictionary) related to 'uncertainty'. We use the frequency of these words in the different minutes to create an uncertainty index and call it the Skip-Gram uncertainty index. We then create the mean uncertainty index as the average mean of the LDA uncertainty index and the Skip-Gram uncertainty index. The third contribution of the paper is the construction of uncertainty measures for the different sections of the minutes.

In the literature, some papers such as Garcia-Herrero, Girardin and Lopez-Marmolejo (2019) try to find a connection between the communications of the Central Bank of Mexico and the financial markets. They manually classify the text as hawkish, neutral, or dovish to understand the sign of the written and oral statements of the Banxico. Then, with a GARCH model they study how the communications of the Central Bank of Mexico influence the most liquid segment of the REPO market, the one-day maturity from early 2005 to the summer of 2013. Other investigations look at the relationship between central bank communications and different variables such as market and real variables. For instance, with LDA and by classifying manually each paragraph, Hansen and McMahon (2016) identify the parts of the FOMC statements that discuss either the 'current economic conditions' or the 'monetary policy decision'. For the parts of the FOMC statements related to the 'current economic conditions', they create a positive-negative index by counting the relative frequency of the words associated with expansion and recession in the dictionary lists of Apel and Blix Grimaldi (2012). And for the 'monetary policy decision' parts of the FOMC, they build a topic-uncertainty index by counting the relative frequency of the words in the uncertainty dictionary of Loughran and McDonald (2011). They then estimate a Factor-Augmented Vector Autogression (FAVAR) to find the effect of topic-uncertainty indeces shocks on market and real variables. They find that shocks in the 'current economic conditions' index are less relevant than shocks in the 'monetary policy decision' index aka the 'forward guidance' index. Lastly, some articles such as

Azqueta-Gavaldon et al. (2020) investigate the effect of uncertainty measures from newspapers on macroeconomic variables. These authors use Word Embeddings and LDA to construct several country uncertainty indices from newspapers in Italy, Spain, Germany and France. They then evaluate the impact of the various country uncertainty indices on investment in machinery and equipment using a Structural VAR for each country.

Finally, via a Structural VAR model, we investigate how shocks in uncertainty during the meetings of the Banxico boards of governors lead to changes in key monetary and financial variables. Our results show also that a unit shock in uncertainty leads to changes of the same sign but of different magnitude in the inter-bank rate and the target interest rate. Moreover, a unit shock in the mean uncertainty index increases the money supply and the consumer price index. Finally, the effect on the exchange rate goes both sides, with a depreciation of the Mexican currency against the US dollar in the same period of the uncertainty shock and appreciation in the period afterwards.

The rest of the paper is organized as follows. Section 2 reviews the minutes of the Central Bank of Mexico. Section 3 describes how we construct the uncertainty index with Latent Dirichlet Allocation (LDA). Section 4 explains how a Skip-Gram uncertainty index is built with the Skip-Gram model and K-Means. Section 5 contains the Structural VAR analysis. Finally, Section 6 presents our conclusions.

3.2 Minutes of the Central Bank of Mexico

The main mission of the Central Bank of Mexico is to preserve the value of the national currency (the ‘peso’) in the long-term to maintain the economic welfare of the Mexican people. In 1994, the Bank of Mexico obtained autonomy to minimize the political influence in its monetary policy decisions aimed at maintaining the value of the ‘peso’ without interference from government. The monetary policy decision is taken by the Bank of Mexico board of governors, comprising the governor and four deputy governors. The governor and the rest of the board members are elected by the President of Mexico and ratified by the senate or the permanent Commission of Congress. The governor of Banxico is elected for six years. The deputy governors are elected for eight years, staggered every two years. This measure aims to guarantee the independence of the members of the board. The monetary policy decision is taken by majority decision of members of the board.

To guarantee the independence of the decisions and to fight against high inflation after 1995, Banxico became more transparent in their decisions and published more economic and financial information. Another guarantee of the independence of Banxico was allowing the peso to float in financial markets. An inflation targeting system was adopted. In 1996, Banxico started setting an annual inflation target and a long-term target, which stood at 3% in 2002. From 1995 to 2007, the Bank of Mexico adopted a monetary policy mechanism called the ‘short’ (‘corto’ in Spanish) or ‘operational target on cumulative balances’. On January 21, 2008 it began a new system for monetary policy based on a target rate for overnight inter-bank transactions.

All the public speeches of members are published on the Banxico website to increase transparency. Furthermore, the Banxico publishes quarterly reports analyzing the economic situation and inflation. These quarterly reports also analyze the implementation of monetary policy. Moreover, a monetary policy statement is released after each monetary policy decision of the board of governors. Since 2011, Banxico has usually published the Spanish version of the minutes two weeks after the meeting and eight times a year. There has also been an English version of the minutes since 2018.

This paper studies the Spanish version of the minutes of the board governors published in the period 2011-2018. The minutes are divided into several parts, illustrating what was presented, discussed and decided during the meeting. Most of the minutes of the Central Bank of Mexico are divided into four sections as follows:

1. Description of the international economic and financial situation;
2. Description of the Mexican economic, financial and inflation situation;
3. Analysis and rationale behind the governing board’s vote;
4. The monetary policy decision.

We process this division manually by assigning to each paragraph a tag identifying the corresponding section and subsection. First, the section ‘description of the international economic and financial situation’ presents mostly the economic and financial situation in important economies such as the United States, Europe, Japan and China. The section combines two subsections, one describing international economic activity and the other international financial activity.

The next section describes the economic, financial and inflation situation in Mexico. It is also a combination of three subsections, describing Mexican economic activity, Mexican financial activity and the situation of inflation in Mexico.

The third section illustrates the discussion of the board members concerning the economic, financial and inflation situation abroad and in Mexico. This section also includes the discussion of board members leading to the monetary policy decision.

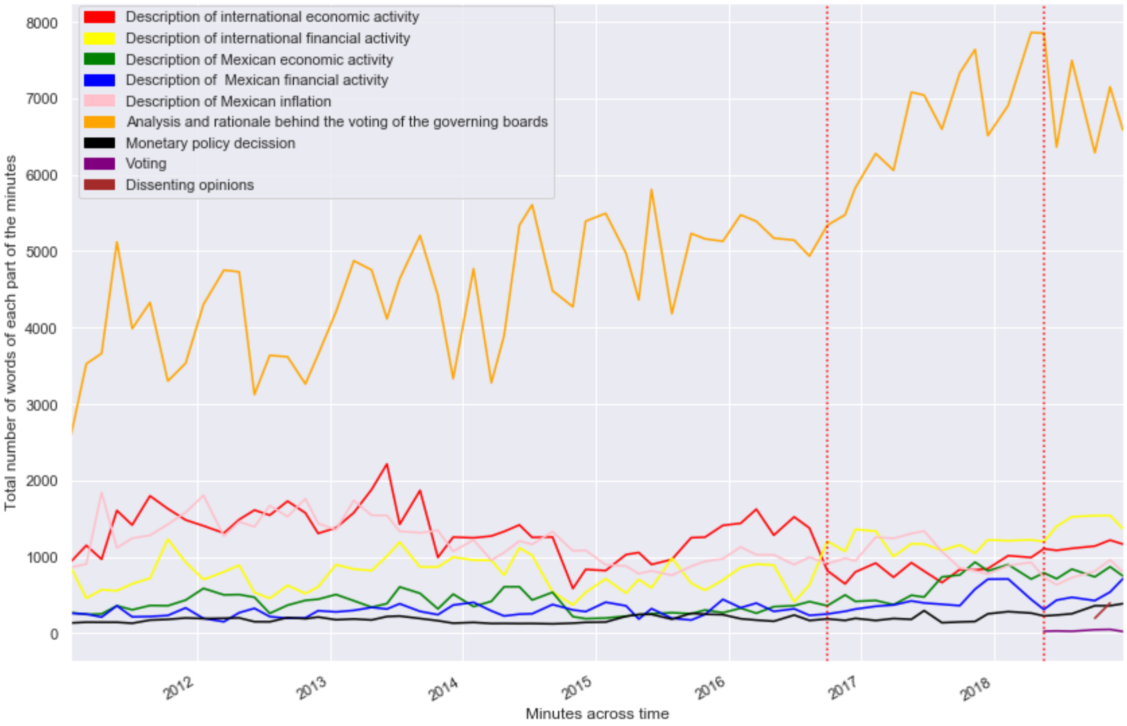


Figure 1: Total number of words in the different sections of Bank of Mexico minutes. We exclude paragraphs repeated over time in the same section. The dotted red lines represent a change in the format of the minutes. After the second dotted red line which corresponds to the 59th minutes, the minutes include two new sections, 'voting' and 'dissenting opinions'.

The final section briefly explains the final decision of the board of governors. Since the minutes numbered 59 (in 2018), the minutes of the Bank of Mexico have included a new section titled 'voting' which publishes the vote of each member of the board. Also, since then, the minutes have included a new section titled 'dissenting opinions' in which board members who voted against the majority explain their reasons.

Figure 1 shows the attention given to each section and subsection of the minutes by counting the total number of words. Most sections are stable over time. However, the 'analysis and rationale behind the governing board vote' section increases after the first change of format. Additionally, there is a slight decline in the size of the 'international economic activity' section over time.

3.3 Latent Dirichlet Allocation

In this and the following section, we investigate the degree of uncertainty in the minutes of Banxico. For that purpose, we construct two uncertainty indices for the minutes of Banxico with different unsupervised machine learning methodologies later combined to obtain one sole index. First, we apply Latent Dirichlet Allocation (LDA) to identify the probability of twenty topics occurring in all the paragraphs of the corpus. We use the probability in the minutes of topic 5 related to 'uncertainty', as the LDA uncertainty index. In the next section, we construct the Skip-Gram uncertainty index with the Skip-Gram and K-Means models. We then build the mean uncertainty index as the average mean of the LDA uncertainty index and the Skip-Gram uncertainty index. Finally, we construct different uncertainty indices for the various sections to understand the main sources of uncertainty in the minutes.

Latent Dirichlet Allocation (LDA) is an unsupervised machine learning technique introduced by Blei, Ng and Jordan (2003) that can be used for textual analysis. LDA aims to identify the topics (combinations of words representing a similar theme) in the documents (here, a document is a paragraph in the minutes) of a corpus (in our paper the corpus is the combination of all the minutes from 2011 to 2018) without a person needing to read the text. The ability of LDA to produce easily interpretable topics is one of its advantages. For that purpose, we assign a name to each topic. For instance, we could choose inflation as a topic since the words with the highest probability for the topic are inflation, price, index, increase and inflationary. However, this labelling does not a affect the results.

3.3.1 LDA uncertainty index

To estimate Latent Dirichlet Allocation (LDA), we manually convert the PDF files of the Spanish version of the minutes into text files. During this process, we delete unnecessary parts for the analysis such as the cover, the graphs, the footnotes and the paragraphs in the minutes that do not provide any relevant information. We then assign a tag to each paragraph to identify the number of the minutes, the sections and subsections. Finally, we convert the entire corpus into lower case.

Before applying LDA we need to 'clean' the text. First, we remove the stop words, i.e. common words that do not provide any information such as 'a', 'we' or 'herself'. We eliminate months and the word 'month' to exclude seasonality topics comprising months of the year. Second, we remove numbers and punctuation marks. Third, we stem the remaining words to their base root. For instance, the words 'inflationary', 'inflation', 'consolidate' and 'consolidating' are transformed into their stem 'inflat' and 'consolid', respectively. Finally, we order the stems following term frequency-inverse document frequency (tf-idf). This index grows in proportion to the number of times a stem appears in a document. However, it decreases by the number of documents that contain that stem. This index serves to exclude common and unusual words. We disregard all stems that have a value of 2,600 or lower.

After identifying 20 topics, we apply Latent Dirichlet Allocation to the 'cleaned' corpus of the minutes of the meetings of the board of governors of Bank of Mexico from 2011 to 2018. There are a total of 264,968 stems in the corpus, with 2,532 unique stems. We set the hyperparameters of the Dirichlet priors following the suggestions of Griffiths and Steyvers (2004). In the estimation, we run 500 iterations before running the sample. We then run 20 samples from points in the chain thinned with a thinning interval of 50.

Table A.1 shows the word-topic matrix, which is the first output from LDA. It shows the first fifteen words with the highest probability for each of the twenty topics. In other words, word 1 is the word or stem with the highest probability in that topic, word 2 is the word with the second highest probability and so on. Since the results are in Spanish, we assign tags to each topic in English. For instance, we assign the tag 'monetary policy' to topic 3 since the stems with the highest probability are 'monetari' (monetary) with a probability of 0.133, 'polit' (policy) with a probability of 0.111, 'banc' (bank) with a probability of 0.092 and 'central' (central) with a probability of 0.054. The topics cover the different sections of the minutes. For instance, the sections that discuss the economic and financial situation are represented by topics 0, 4, 6, 10, 16 and 17. Topics 3, 12, 13, 14 and 19 are related to the sections that discuss expectations and the monetary policy discussion. Several topics, for example 11 and 18, are linked to inflation. Other topics, for example 2, 8 and 9, are related to the international economic and financial conditions.

The second output of LDA is the distribution of topic probabilities per document. In our paper, each paragraph corresponds to a document. We estimate the distribution of topics in each set of minutes since our goal is to construct an LDA uncertainty index for the minutes with one of the topics. In particular, we are interested in topic 5 since it comprises words related to 'risk' and 'uncertainty'. Following Bybee et al. (2020), we use the weighting of this 'uncertainty' and 'risk' topic to construct an uncertainty index for the minutes. These authors use a Structural VAR model to investigate how higher attention to a topic, formed by words related to recession, is linked with a decrease in industrial production and unemployment. In our research, we assume that the probability of topic 5 is a proxy of the level of uncertainty during the meetings of the Banxico board of governors. To construct the LDA uncertainty index, we multiply the probability per set of minutes of topic 5 by 100 and then divide it by the mean probability of topic 5 for all the minutes as shown in the following equation:

$$R_s = 100 \, \frac{U_s}{\frac{1}{M}\sum_{m=1}^{M} U_m}, \tag{1}$$

where U_s is the probability of topic 5 in minutes s and the denominator of Equation (1) is the mean probability of topic 5 for all the minutes. Furthermore, R_s is the standardized topic 5 uncertainty index or LDA uncertainty index.

We compute the LDA uncertainty index for each one of the following sections of the minutes:

1. Description of the international economic and financial situation;
2. Description of the Mexican economic, financial and inflation situation;
3. Analysis and rationale behind the governing board vote;
4. Monetary policy decision.

Figure A.1 shows the time series of the LDA uncertainty index for the first section ('description of the international economic and financial situation') aka the LDA 'international' uncertainty index. Figure A.1 also shows the evolution of the LDA uncertainty index for the second section, aka the LDA 'Mexican' uncertainty index. In 2012, the LDA 'international' uncertainty index is higher than the LDA 'Mexican' uncertainty index due to the Eurozone crisis. After 2014, the LDA 'international' section uncertainty index is higher than the LDA 'Mexican' section uncertainty index until the peak in the LDA 'Mexican' uncertainty index due to the NAFTA negotiations and Mexican elections in May 2018.

Figure A.2 shows the time series of the LDA uncertainty index for the third section ('analysis of and rationale behind the governing board vote') aka the LDA 'analysis' uncertainty index. Values are above the mean (100) in the LDA 'analysis' uncertainty index after 2016 due to higher uncertainty in Mexico and elsewhere. Furthermore, the LDA 'analysis' uncertainty index increases substantially at the end of 2017. Figure A.2 also shows the LDA uncertainty index for the 'monetary policy decision' section. However, this section is not used in the following analysis because it is too small to provide consistent results over time.

Figure A.3 shows the evolution of the LDA uncertainty index for all the minutes. We compare the LDA uncertainty index with the Economic Policy Uncertainty (EPU) index for Mexico created by Baker, Bloom, and Davis (2016) from the Mexican newspapers 'El Norte' and 'Reforma'. The Mexican EPU index is standardized following the same formula as in Equation (1). Moreover, the LDA uncertainty index for all the minutes shows a similar trend to the LDA 'analysis' uncertainty index because the 'analysis' section is the largest.

3.4 Word Embedding and Skip-Gram Model

Word Embeddings were introduced by Mikolov et al. (2013a). Word Embeddings are continuous vector representations of words that preserve syntactical and semantic similarities between words in a Euclidean Space, having a limited number of dimensions. The main idea of Word Embeddings is that a lot of meaning can be obtained from a word by representing this word by the words around it. For instance, in the following documents:

1. the economy experienced growing *uncertainty* about the growth capacity,
2. the economy experienced growing *concerns* about the growth capacity,

the words *uncertainty* and *concerns* have similar meanings related to doubt and worry. The words *uncertainty* and *concerns* are preceded by the 'the economy experienced growing' and followed by 'about the growth capacity'. The basic idea of Word Embeddings is to create a dense vector for each word type that is good at predicting the words appearing in a given context, also represented by a vector. In this case, we prefer a machine learning method that puts the vectors of words with similar meanings, such as *uncertainty* and *concerns*, into the same part of the vector space since they appear in the same context. To create the Word Embeddings in this way, the Skip-Gram model is used as introduced by Mikolov et al. (2013a). The Skip-Gram model is a neural network method that tries to predict context words given a center word. This process is repeated for all the unique terms in the corpus, and for each term a vector of probabilities is created and placed in the vector space. For instance, in the first sentence above, *uncertainty* is the input or center word. The rest of the words are output or context words:

$$\underbrace{\text{economy experienced growing}}_{\text{Output}}\ \underbrace{\textit{uncertainty}}_{\text{Input}}\ \underbrace{\text{about the growth capacity}}_{\text{Output}}$$

In the previous example, the Skip-Gram model gives the probability distribution of each of the context words depending on uncertainty, the center word in this example. For instance, $P(\text{ growing } \mid \text{ uncertainty })$ or $P(\text{ about } \mid \text{ uncertainty })$. For each word ($t = 1, ..., T$), the number of the words in the context is given by the size of the window, m, that determines the number of context words before and after each center word. A window size of five means that we compute the probabilities of the five output words before the input word and the five output words that follow.

3.4.1 K-Means

K-Means Clustering is a technique that tries to cluster observations close to each other in the input space. In this paper, we use K-Means to cluster the the vectors from Word Embeddings into C disjoint groups (clusters). We then identify the cluster that encompasses the words related to 'uncertainty' as in Soto (2021).

K-Means is a centroid-base algorithm. This algorithm aims to find the cluster assignments of all m observations to C clusters that minimize within cluster distances (normally measured by the Euclidean distance) between each point x_i and its cluster centre μ_c (Chakraborty and Joseph, 2017). The corresponding cost function is:

$$ERR(X, C) = \frac{1}{m} \sum_{c=1}^{C} \sum_{x_i \in C_c} || \, x_i - \mu_c \, ||^2. \tag{2}$$

Here, the sum of squares is normalized by the number of observations, which is required to compare clusters of different size. In order to establish a fixed number of clusters C, we alternate cluster assignment steps with centroid shifting. During the clustering assignment, we assign each observation x_i to its closest centroid C_i. For each centroid we calculate its new position. Moreover, highly-correlated features must be avoided since the might cause spurious clustering. Finally, the number of clusters has to be decided. They can be evaluated in various ways such as the 'silhouette coefficient' or the 'elbow-method' (Chakraborty and Joseph, 2017).

3.4.2 Skip-Gram uncertainty index

We estimate Word Embeddings with the Skip-Gram model using the minutes of meetings of the Bank of Mexico board of governors. To apply the Skip-Gram model, the corpus is processed differently than in LDA. First, the words are not stemmed since we could lose the semantic differences between words. Secondly, we identify pairs of words or bigrams appear with a frequency higher than 10, this helps to identify couples of words that represent the same term or idea.

When the Skip-Gram model is applied, a hidden-layer (H) of 200 is used as well as a context window size (m) of 10. Furthermore, we estimate K-Means with 145 clusters, selecting these parameters because they provided more logical results after several trials with different combinations.

Words in the same clusters have similar meanings. We put all the words in the clusters containing 'incertidumbre' (uncertainty), 'incierto' (uncertain), 'inquietud' (unease or concern) and 'riesgo' (risk) in the same list of words. We use this list as our dictionary related to the sentiments 'uncertain' and 'risk'. Tables 1, 2, 3 and 4 show the words in the clusters of 'incertidumbre' (uncertainty), 'incierto' (uncertain), 'inquietud' (unease or concern) and 'riesgo' (risk), respectively. The results include words related to the economic cycle (‘burbujas’, ‘volatilidad_financiera’), catastrophic natural events (‘tornado’) or political events (‘electoral’, ‘proceso_electoral’, ‘tclan’). In addition, some words indicate the possibility that an event taking place (‘futuro_proximo’, ‘podría_conducir’, ‘podría_traer’, ‘probabilidad’).

Our ‘uncertainty’ dictionary better captures the ‘uncertainty’ sentiment of the minutes than other pre-established dictionaries because our dictionary is built from the minutes themselves. The Skip-Gram and K-Means models allow dictionaries to be created for languages not common in economic dictionaries such as Spanish, without the need for human intervention and in less time. Our results shed some light on the application of these algorithms in economics. However, the results would be more accurate with larger databases.

Table 1: List of words in the cluster containing the word ‘incertidumbre’ (uncertainty).

américa, electoral, entorno_externo, eventos, evolución_desfavorable, factores_externos, incertidumbre, incertidumbre_asociada, incertidumbre_relacionada, interés_externas, libre_comercio, moneda_nacional, negociación, negociaciones, norte_tlcan, nuevo_episodio, nuevos_episodios, presionada, proceso_electoral, puede_descartarse, reacción_adversa, recrudecimiento, renegociación, tlcan, tratado, turbulencia, volatilidad_financiera.

Table 2: List of words in cluster containing the word ‘incierto’ (uncertain).

advirtieron, alto_grado, aún, carácter_estructural, cíclicos, compleja, deflacionarias, desaparecido, disipado, enfrenta, enfrentando, existe, existen, existencia, expresaron, externas, extremos, futuro_próximo, incierto, lejos, marcadamente, materialicen, materializado, naturaleza_cíclica, opinó, parecen, perciben, podría_conducir, podría_traer, pone, prevalece, prevalecen, probabilidad, razones, tornado.

Table 3: List of words in the cluster containing the word ‘inquietud’ (unrest or concern).

abruptos, abundante, acentuar, adelante, agencias_calificadoras, alta_frecuencia, alternativas, amplios, astringencia, aunada, burbuja, burbujas, competitivas, conocido, constituyen, deberse, deteriorar, diferenciación, dificultar, elemento, factor, fuente, generando, inquietud, intensidad, internas, interpretar, invertir, libera, negativos, normalidad, noticias, percepción, principio, propiciando, resultando, seguramente, significativos, tecnológico, traducirse, vulnerable.

Table 4: List of words in the cluster containing the word 'riesgo' (risk).

abruptas, abrupto, acentuarse, acrecentado, agotamiento, agravamiento, ajuste_desordenado, altamente, aminorar, apreciarse, conflicto, conflictos_geopolíticos, correcciones, dependencia, descartan, específicos, exacerbar, exacerbarse, factor_adicional, generado, geopolíticas, geopolítico, idiosincráticos, inestabilidad_financiera, influenciados, internacional, materia_comercial, materialización, naturaleza_geopolítica, nerviosismo, nuevos_periodos, optimismo, oriente_medio, podría_ocasionar, podría_representar, podrían_generar, políticos_geopolíticos, posibles_consecuencias, potenciales, prevalecido, propiciado, provocar, pudieran_tener, ratificación, reciben, regreso, restricciones, restringido, resurgimiento, revaluación, riesgo, severos, sistémica, sobrevaluación, sujetos, suman, temas, tensión.

We construct an uncertainty index for the minutes of the Central Bank of Mexico using the 'uncertainty' dictionary. To construct this uncertainty index, we count the number of times any word in the clusters of 'uncertainty', 'uncertain', 'unrest' and 'risk' appear in each set of minutes T_s. In Equation (3), we divide T_s by the total number of words in each set of minutes, (N_s), to compute an uncertainty score for each set, S_s. In Equation (4), we estimate the Skip-Gram uncertainty index or standardized score, represented by the term D_s. To compute D_s, we multiply S_s by 100 and divide it by the mean of the uncertainty score for all the minutes:

$$S_s = T_s / N_s, \tag{3}$$

$$D_s = 100 \, \frac{S_s}{\frac{1}{M} \sum_{m=1}^{M} S_m}. \tag{4}$$

Figure A.3 shows the evolution of the Skip-Gram uncertainty index for all the minutes. The Skip-Gram uncertainty index shows a similar pattern to the LDA uncertainty index. We follow the same procedure to create the Skip-Gram uncertainty indices for the main sections of the minutes as we did for LDA. Specifically, we create Skip-Gram uncertainty indices for the following sections:

1. Description of the international economic and financial situation;

2. Description of the Mexican economic, financial and inflation situation;

3. Analysis of and rationale behind the governing board vote.

Figure A.4 shows the three Skip-Gram section uncertainty indices created. We observe similar patterns to the LDA section uncertainty indices described above.

Finally, we create the mean uncertainty index as the mean of the Skip-Gram uncertainty index and the LDA uncertainty index. Figure A.5 shows the mean uncertainty index jointly with the EPU index of Mexico. There is a high peak in the EPU index in 2017 not captured by the mean uncertainty index.

3.5 Structural VAR: Relating Uncertainty to Monetary and Financial Variables

We investigate how uncertainty in the minutes of the meetings of the Bank of Mexico board of governors affects the key financial variables for monetary policy such as the inter-bank rate. For this purpose, we estimate a Structural VAR model as follows:

$$B_0 Y_t = \sum_{i=1}^{p} B_i Y_{t-i} + \omega_t, \tag{5}$$

where ω_t refers to a structural innovation or structural shock, but also represents a mean zero serially uncorrelated error term. The term Y_t is a K-dimensional time series, $t = 1, \ldots, T$, which is approximated by a vector autoregression of finite order p. The matrix B_0 represents the simultaneous associations of the variables in the model (Kilian and Lütkepohl; 2017). The model can be expressed in reduced form as:

$$Y_t = \underbrace{B_0^{-1} B_1}_{A_1} Y_{t-1} + \cdots + \underbrace{B_0^{-1} B_p}_{A_p} Y_{t-p} + \underbrace{B_0^{-1} \omega_t}_{u_t}, \tag{6}$$

where the new error vector, u_t, is a linear transformation of the old error vector, ω_t. Once we estimate the reduced form, the problem is to recover the structural representation of the VAR model, as represented by Equation (5). In particular, the main issue is how to obtain B_0 since it is able to estimate ω_t due to $\omega_t = u_t B_0$ and also to estimate B_i since $B_i = A_i B_0$, for $i = 1, \ldots, p$. To obtain ω_t, we 'orthogonalize' the reduced form error which consists in making the errors mutually uncorrelated. This can be achieved by defining the lower-triangular KxK matrix P with positive main diagonal such as $PP' = \sum_u$, where $\sum_u$ is the variance-covariance matrix of u_t. We know that the matrix P is the lower-triangular Cholesky decomposition of $\sum_u^2$. Therefore, one of the solutions to obtain ω_t is the condition $\sum_u = B_0^{-1} B_0^{-1'}$ in which $B_0^{-1} = P$ (Kilian and Lütkepohl; 2017).

In this model, the vector $Y_t = [\Delta f_t, \Delta i_t, \Delta m_t, \Delta e_t, \Delta \pi_t]$ where, Δi_t is the logarithmic difference of the average monthly value of the inter-bank rate for less than 24 hours, Δm_t is the logarithmic difference of the M3 money supply in Mexico, Δe_t stands for the logarithmic difference of the exchange rate of the Mexican peso against the US dollar, and $\Delta \pi_t$ indicates the logarithmic difference of the consumer price index in Mexico. Finally,

Δf_t stands for the logarithmic difference in the uncertainty index. The value of the previous observation of the uncertainty index is assigned to the months when meetings did not occur. All the financial variables are from the Federal Reserve Bank of St. Louis and all variables are in logs and differences to make them stationary since augmented Dicky-Fuller tests indicate that they are all I(1). However, the variables cannot be checked for joint stationarity because of the limited database.

According to Akaike Information Criteria (AIC) and the Hannan and Quinn information criterion (HQIC), one is the optimum number of lags. The SVAR model complies with the stability condition since all roots of the characteristic polynomial are outside the unit circle. Identification of the structural shock is obtained by appealing to the usually estimated Cholesky decomposition proposed by Sims (1980). The Cholesky decomposition involves the so-called recursiveness assumption. Specifically, the recursiveness assumption is an economic assumption in the timing of the reaction to the shocks of the variables. In other words, the recursiveness assumption imposes order between the variables. In this paper, the uncertainty index (Δf_t) simultaneously affects the other variables but is not itself simultaneously affected by the remaining variables, as in Bloom (2009) and Nodari (2014). Therefore, Δi_t simultaneously affects Δm_t , Δe_t and $\Delta \pi_t$. Δm_t simultaneously impacts Δe_t and $\Delta \pi_t$. Subsequently, it continues in the same way for the last two variables. In our specification, we assume that the uncertainty index simultaneously affects all the financial variables. Moreover, a shock in the inter-bank interest rate has a simultaneous effect on the money supply. For instance, a higher interest rate might reduce the money supply since banks would likely borrow less. However, a shock in the money supply does not have a simultaneous effect on the interest rate. The money supply directly affects the exchange rate. The greater the money supply, the lower the value of the currency, all else being equal. According to our specification, inflation is affected simultaneously by all the variables, but inflation does not simultaneously affect the remaining variables. An increase in money supply could lead to higher prices in the same period.

We estimate a Structural VAR model for each one of the uncertainty indices. First, we estimate a Structural VAR model with the mean uncertainty index. We then estimate a Structural VAR for each of the four uncertainty indices computed with LDA and Skip-Gram, respectively. The uncertainty indices included in the different Structural VAR estimations include: 1) the mean uncertainty index for all the minutes; 2) the LDA uncertainty index for all the minutes; 3) the LDA uncertainty index of the ‘description of the international economic and financial situation’ section; 4) the LDA uncertainty index of the ‘description of the Mexican economic, financial and inflation situation’ section; 5) the LDA uncertainty index of the ‘analysis of and rationale behind the governing board vote’ section; 6) the Skip-Gram uncertainty index for all the minutes; 7) the Skip-Gram uncertainty index of the ‘description of the international economic and financial situation’ section; 8) the Skip-Gram uncertainty index of the ‘description of the Mexican economic, financial and inflation situation’ section; 9) the Skip-Gram uncertainty index of the ‘analysis of and rationale behind the governing board vote’ section.

3.5.1 Impulse response functions

To the best of our knowledge, this paper is one of the first attempts to disentangle the sources of uncertainty in the meetings of the board of governors of the Bank of Mexico. In particular, our aim is to create different section uncertainty indices to understand the degree of uncertainty in the various sections of the minutes of the meetings of the board of governors. However, the limited length of the sections might skew the robustness of the ‘international’ and ‘Mexican’ section indices because unsupervised machine learning techniques provide more accurate results with larger databases.

Figures A.6 to A.14 show the results of the impulse response functions of the Structural VAR estimations and the effect of a unit shock on the uncertainty index for the financial variables at time t, then on $t + 1$, and so on.

Figure A.6 shows the effect of an increase in a unit shock in each one of the uncertainty indices for the inter-bank interest rate. One standard-deviation shock in the mean uncertainty index leads to an increase in the inter-bank rate during the same period. Nonetheless, this effect disappears in the periods after the shock. The results of the impulse response function of the LDA ‘international’ uncertainty index are similar to those of the mean uncertainty index. On the contrary, unit shocks in the LDA and Skip-Gram ‘Mexican’ uncertainty indices lead to a decrease in the inter-bank rate in the same period.

Figure A.7 shows the impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices in the money supply. In particular, a unit shock in the mean uncertainty index leads to an increase in the money supply (M3) in the same period, suggesting that Banxico might increase the money supply and hence liquidity in response to uncertain circumstances. However, this effect tends to disappear in the following period, and even turns negative for some of the section uncertainty indices such as the LDA ‘analysis’ uncertainty index.

Figure A.8 shows the impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices in the exchange rate. An increase in the mean uncertainty index leads to the depreciation of the peso against the US dollar in the same period. This depreciation is followed by an appreciation in the subsequent period. A unit shock in the LDA and Skip-Gram ‘international’ section uncertainty indices leads to the appreciation of the Mexican peso against the US dollar in the same period of the shock. These results might suggest that uncertainty abroad increases the value of the Mexican peso.

Figure A.9 demonstrates that a unit shock in the mean uncertainty index boosts the consumer price index in the period after the shock but not in the same period as the shock. Moreover, an increase in the LDA and Skip-Gram 'Mexican' section uncertainty indices leads in the same period to an increase of the consumer price index. We should highlight that the 'Mexican' section of the minutes illustrates the inflation situation and expectations in Mexico. Thus, our results confirm that there is a positive relationship between the LDA and Skip-Gram 'Mexican' section uncertainty indices and inflation.

3.5.2 Alternative interest rate specification

In this alternative SVAR specification, we substitute the logarithmic difference of the inter-bank rate with the logarithmic difference of the target interest rate as decided in the meeting in SVAR model Equation (5). We estimate the Structural VAR model with the three uncertainty indices built from the entire corpus of minutes, as follows: 1) the mean uncertainty index; 2) the LDA uncertainty index; 3) the Skip-Gram uncertainty index.

Figure A.10 shows the results of the impulse response functions of the Structural VAR estimations of a unit shock in each of the three uncertainty indices on the target interest rates. Our results show that a unit shock in uncertainty leads to a small increase of the target interest rate in the same period as the shock followed by a decrease in the target interest rate in the period after the shock. The increase in the target interest rate in the same period of the shock is smaller in absolute terms than the decrease in the target interest rate in the period after the shock.

The results of the impulse response functions in Figure A.10 are similar to those in Figure A.6 corresponding to one standard-deviation in each of the uncertainty indices in the inter-bank interest rate. The results of both SVAR estimations tend to be similar to an increase of the inter-bank and target interest rates in the same period, followed by a decrease in the period after the shock. However, the increase in the inter-bank interest rate is higher than the increase in the target interest rate in the same period as the shock. On the other hand, the decline in the inter-bank interest rate is lower in absolute terms than the decline in the target interest rate in the period after the shock. This might indicate a partial failure of the financial transmission mechanism since lower target interest rates by the Banxico might not be fully passed on to the inter-bank rate negotiated by the financial sector. However, we leave this question open for future investigations.

Finally, we estimate the SVAR model replacing the minutes uncertainty indices with the global EPU index and the Mexican EPU index constructed by Baker, Bloom, and Davis (2016). There are two main differences in the construction of the minutes uncertainty indices and the EPU indices that could affect the results. First, the minutes uncertainty indices are constructed with the corpus of the minutes in which the 'Mexican' and international economic, financial and inflation conditions are discussed in due proportion. However, the global EPU index and the Mexico EPU index are built from newspaper articles that might not always provide information similar to the minutes. For instance, the global EPU index is built with newspapers in different countries Second, the mean uncertainty index is constructed with unsupervised machine learning techniques such as Latent Dirichlet Allocation and the Skip-Gram model. On the contrary, the EPU indices are built by counting the number of articles that contain at least one word from each of three groups of words pre-established by the researches. The first group of words contains words related to policy terms such as 'regulation' or 'deficit', the second group comprises the words 'uncertain' and 'uncertainty' and the third group of words comprises the words 'economic' and 'economy'.

Figure A.11 shows the impulse response functions corresponding to a shock in each of the uncertainty indices in the inter-bank rate. An increase in the global and Mexico EPU indices leads to an increase in the inter-bank rate in the same period as the shock. The same is true of the mean uncertainty index.

Figure A.12 shows the impulse response functions corresponding to a shock in uncertainty in money supply. The impulse response functions of the EPU indices show an increase in money supply in the same period and the period after the uncertainty shock. However, the effect becomes negative after two periods, whereas the effect of a unit shock in the mean uncertainty index seems to be positive in most time periods.

Figure A.13 shows the impulse response functions corresponding to a shock in uncertainty in the exchange rate of the Mexican peso against the US dollar. A unit shock in the EPU indices and the mean uncertainty leads to a depreciation of the peso during the same period as a shock. This initial depreciation is followed by an appreciation in the case of the Global EPU index and the mean uncertainty index in the period after the shock. In the case of the Mexico EPU index, the appreciation of the Mexican peso occurs two periods after the shock.

Figure A.14 shows the impulse response function of the Structural VAR model corresponding to the effect of a unit shock in uncertainty in the consumer price index. The results are different for the three uncertainty indices. However, there is an increase in the consumer price index in the same period as the shock in the impulse response functions of the global EPU and mean uncertainty indices.

3.6 Conclusion

This paper creates text uncertainty measures of the minutes of the meetings of the Bank of Mexico board of governors. In particular, we construct two uncertainty measures with unsupervised machine learning techniques from the Spanish version of the minutes. The first uncertainty index is constructed with LDA. Then, a second uncertainty index is created for the minutes with Skip-Gram and K-Means. We combine the LDA uncertainty index with the Skip-Gram uncertainty index to construct a mean uncertainty index. We also create the LDA and the Skip-Gram uncertainty indices for each of the three main sections of the minutes.

Furthermore, with Structural VAR we estimate the effect of one standard deviation in uncertainty on some monetary and financial variables. A unit shock in the mean uncertainty index leads to changes of the same sign but different magnitude in the inter-bank rate and the target interest rate of the Central Bank of Mexico. Moreover, an increase in the mean uncertainty index leads to an increase in the money supply (M3) and inflation in the same period as the shock. Finally, a unit shock in the mean uncertainty index leads to depreciation of the Mexican peso against the US dollar in the same period as the shock.

Future research could use supervised machine learning techniques to create sentiment indices for the Banxico minutes. For instance, researches might study the effect of the communication of Banxico on financial markets with text measures constructed using machine learning techniques such as Random Forest.

Bibliography

[1] Apel, M., and Blix Grimaldi, M. (2012). The information content of central bank minutes. *Working Paper Series, Sveriges Riksbank (Central Bank of Sweden)*, 261, Apr.

[2] Arango, L. E., Pantoja, J., and Velasquez, C. (2017). Effects of the central bank's communications in Colombia. A content analysis. *Borradores de Economía Banco de la Republica de Colombia*, 1024.

[3] Azqueta-Gavaldón, A. (2017). Developing news-based economic policy uncertainty index with unsupervised machine learning. *Economics Letters*, 158, 47-50.

[4] Azqueta-Gavaldon, A., Hirschbühl, D., Onorante, L., and Saiz, L. (2020). Economic policy uncertainty in the Euro area: an unsupervised machine learning approach. *European Central Bank Working Paper Series*, 2359.

[5] Baker, S. R., Bloom, N., and Davis, S. J. (2016). Measuring economic policy uncertainty. *The Quarterly Journal of Economics*, 131, 1593-1636.

[6] Bernal, Á. I. M., and Pedraz, C. G. (2020). Sentiment analysis of the Spanish financial stability report. *Documentos de Trabajo, Banco de España*, N. 2011.

[7] Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning research*, 3, 993-1022.

[8] Bloom, N. (2009). The impact of uncertainty shocks. *Econometrica*, 77, 623–685.

[9] Bybee, L., Kelly, B. T., Manela, A., and Xiu, D. (2020). The structure of economic news. *National Bureau of Economic Research*, w26648.

[10] Cabral, R., and Guimaraes, B. (2015). O comunicado do banco central. *Revista Brasileira de Economia*, 69, 287-301.

[11] Ciro, G., Camilo, J., and Anzoátegui-Zapata, J. C. (2019). Efectos de los anuncios de política monetaria y la credibilidad sobre las expectativas de inflación: evidencia para Colombia. *Apuntes del CENES*, 38(67), 73-94.

[12] Costa-Filho, A. E., and Rocha, F. (2010). Como o mercado de juros futuros reage à comunicação do banco central?. *Economia Aplicada*, 14, 265-292.

[13] Garcia-Herrero, A., Girardin, E., and Dos Santos, E. (2017). Follow what I do, and also what I say: monetary policy impact on Brazil's financial markets. *Economia*, 17, 65-92.

[14] Garcia-Herrero, A., Girardin, E., and Gonzalez, H. (2017). Analyzing the impact of monetary policy on financial markets in Chile. *Revista de Analisis Economico*, 32, 2, 3-21.

[15] Garcia-Herrero, A., Girardin, E., and Lopez-Marmolejo, A. (2019). Mexico's monetary policy communication and money markets. *International Journal of Economics and Finance*, 11, 81.

[16] Ghirelli, C., Pérez, J. J., and Urtasun, A. (2019). A new economic policy uncertainty index for Spain. *Economics Letters*, 182, 64-67.

[17] Hansen, S., McMahon, M., and Prat, A. (2017). Transparency and deliberation within the FOMC: a computational linguistics approach. *The Quarterly Journal of Economics*, 133, 801-870.

[18] Hansen, S., McMahon, M., and Tong, M. (2019). The long-run information effect of central bank communication. *Journal of Monetary Economics*, 108, 185-202.

[19] Herrerias, R., and Gurrola, P. (2012). Monetary policy announcements and short-term interest rate futures volatility: evidence from the Mexican market. *International Finance*, 15, 225-250.

[20] Kilian, L., and Lütkepohl, H. (2017). Structural Vector Autoregressive Analysis. *Cambridge University Press.*

[21] Loughran, T., and McDonald, B. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-Ks. *Journal of Finance*, 66:1, 35-6.

[22] Manela, A., and Moreira, A. (2017). News implied volatility and disaster concerns. *Journal of Financial Economics*, 123, 137-162.

[23] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *Preprint Arxiv*, 1301, 3781.

[24] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 3111-3119.

[25] Nodari, G. (2014). Financial regulation policy uncertainty and credit spreads in the US. *Journal of Macroeconomics*, 41, 122-132.

[26] Ortiz, A., Rodrigo, T., and Turina, J. (2017). How do the central banks talk?: a big data approach to Turkey. *BBVA Research Working Paper*, 17/24.

[27] Shapiro, A. H., and Wilson, D. (2019). Taking the Fed at its word: direct estimation of central bank objectives using text analytics. *Federal Reserve Bank of San Francisco Working Paper*, 2019-02.

[28] Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, 1-48.

[29] Soto, P. E. (2021). Breaking the word bank: measurement and effects of bank level uncertainty. *Journal of Financial Services Research*, 59(1), 1-45.

[30] Taborda, R. (2015). Procedural transparency in Latin American central banks under inflation targeting schemes. A text analysis of the minutes of the Boards of Directors. *Ensayos sobre Política Económica*, 33, 76-92.

[31] Thorsrud, L. A. (2016). Nowcasting using news topics. Big data versus big bank. *Norges Bank Working Paper*, 20/2016.

Appendix

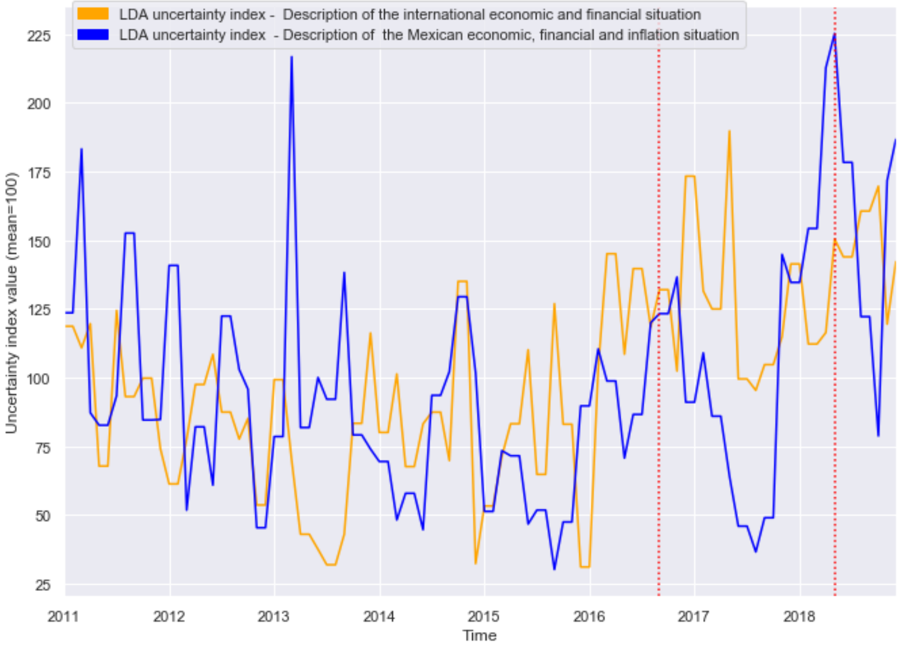


Figure A.1: LDA uncertainty indices for the 'description of the international economic and financial situation' section and the 'description of the Mexican economic, financial and inflation situation' section in the minutes from 2011 to 2018. The dotted red lines represent a change in the format of the minutes.

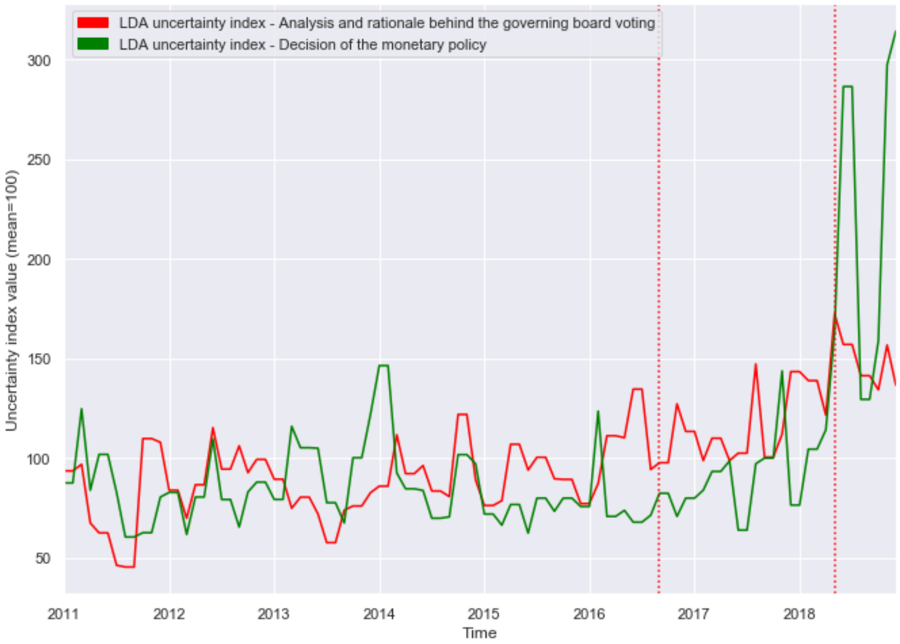


Figure A.2: LDA uncertainty indices for the 'analysis of and rationale behind the governing board vote' section and the 'monetary policy vote' section in the minutes from 2011 to 2018. The dotted red lines represent a change in the format of the minutes.

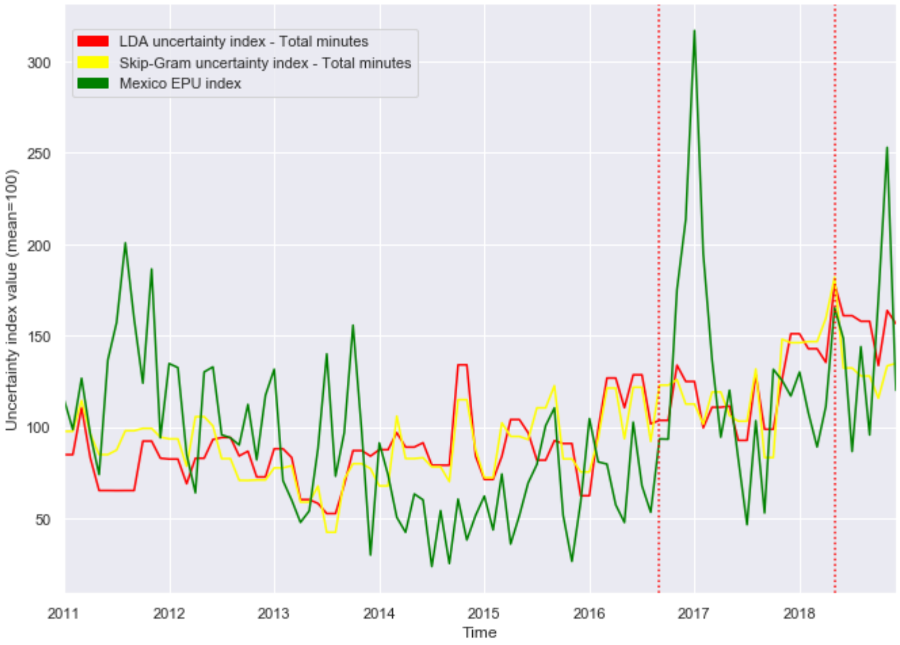


Figure A.3: Mexico EPU monthly uncertainty index, Skip-Gram uncertainty index and LDA uncertainty index in the minutes from 2011 to 2018. The dotted red lines represent a change in the format of the minutes.

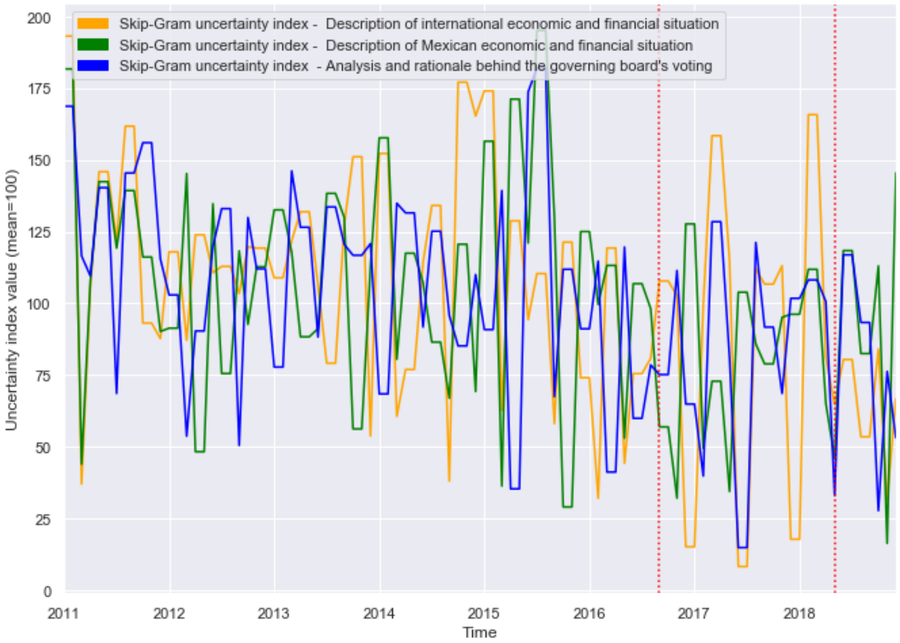


Figure A.4: Skip-Gram uncertainty indices for the ‘description of the international economic and financial situation’ the ‘description of the Mexican economic, financial and inflation situation’ sections and the ‘analysis of and rationale behind the governing board vote’ sections in the minutes from 2011 to 2018. The dotted red lines represent a change in the format of the minutes.

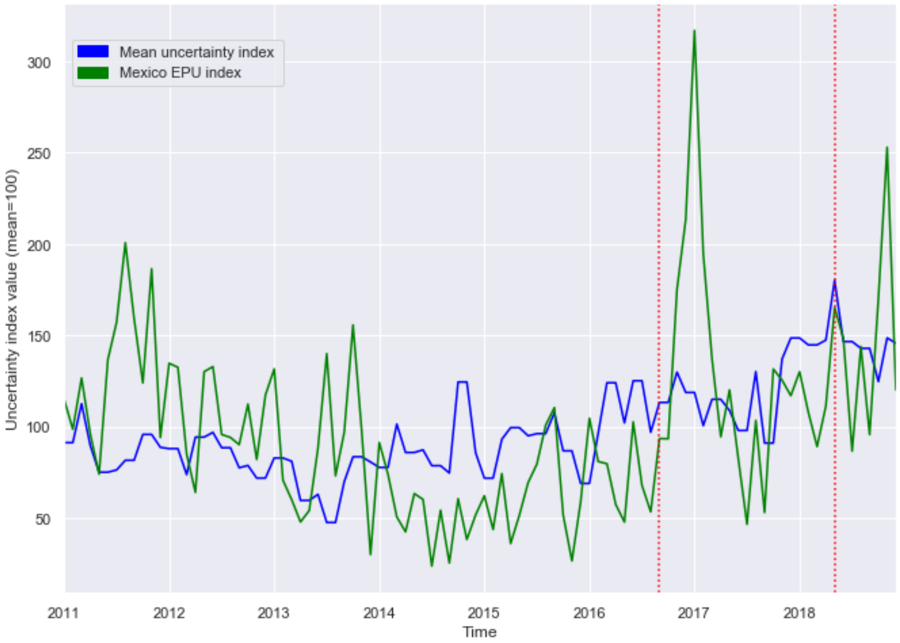
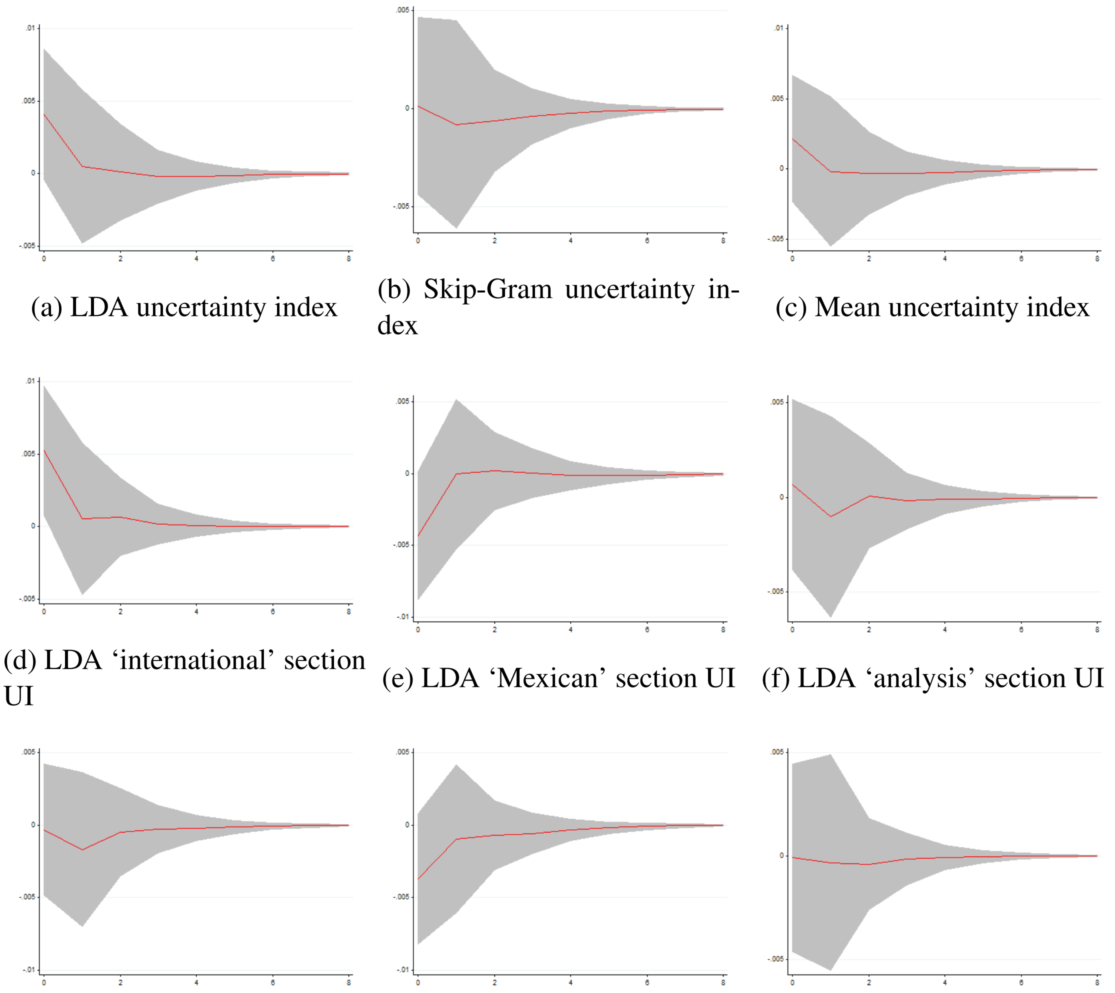


Figure A.5: Mexico EPU monthly uncertainty index and mean uncertainty index in the minutes from 2011 to 2018. The dotted red lines represent a change in the format of the minutes.



(a) LDA uncertainty index

(b) Skip-Gram uncertainty index

(c) Mean uncertainty index

(d) LDA 'international' section UI

(e) LDA 'Mexican' section UI

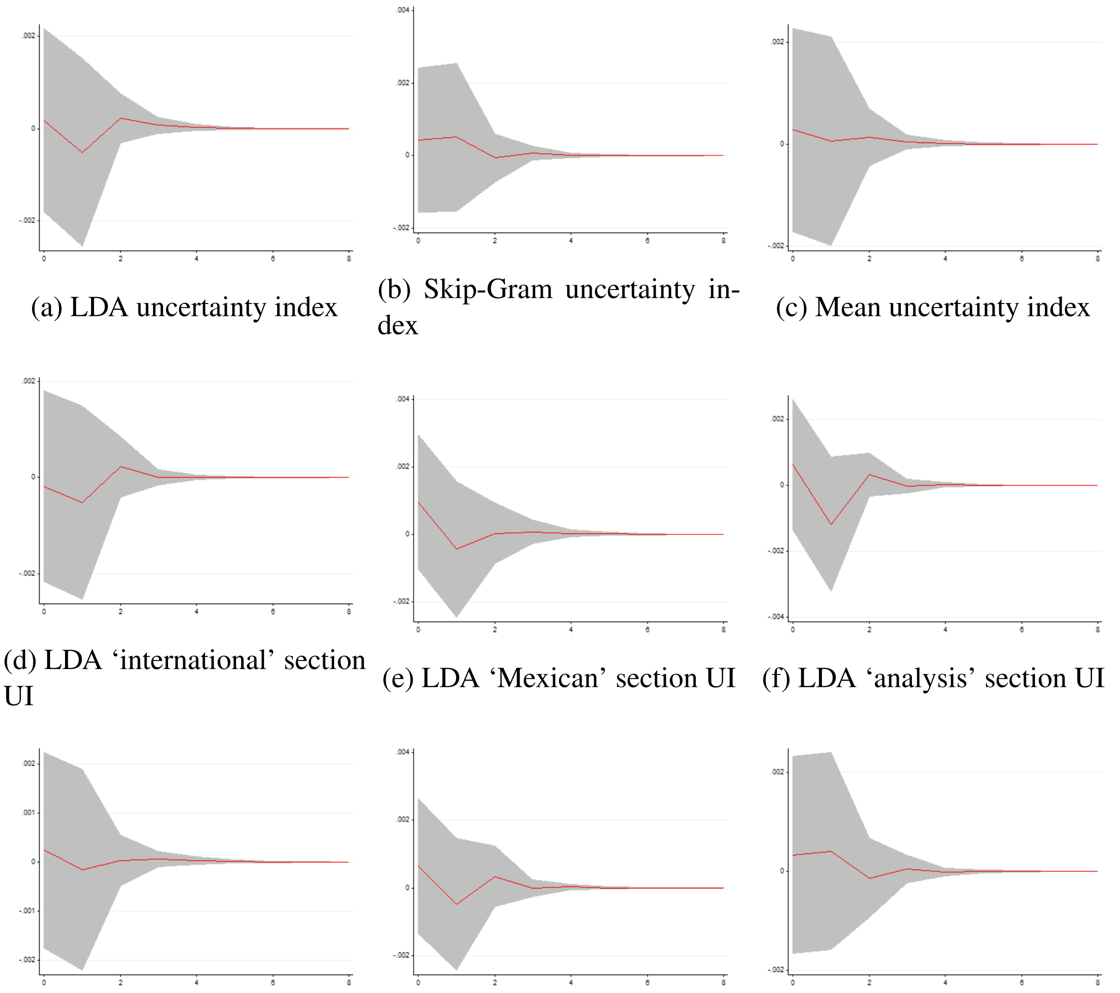
(f) LDA 'analysis' section UI

(g) Skip-Gram 'international' section UI

(h) Skip-Gram 'Mexican' section UI

(i) Skip-Gram 'analysis' section UI

Figure A.6: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices for the minutes of the Bank of Mexico for the period 2011-2018. The gray area shows the 95% confidence intervals computed using bootstrapped standard errors (200 replications). The Y -axis is the % change in the monthly interbank rate (24 hours) and the X-axis represents time in months (8 months). The LDA and Skip-Gram 'international' section UI refers to the LDA and Skip-Gram uncertainty indices for the 'description of international economic and financial situation' section. 'Mexican' and 'analysis' refer to the other two sections.



(a) LDA uncertainty index

(b) Skip-Gram uncertainty index

(c) Mean uncertainty index

(d) LDA ‘international’ section UI

(e) LDA ‘Mexican’ section UI

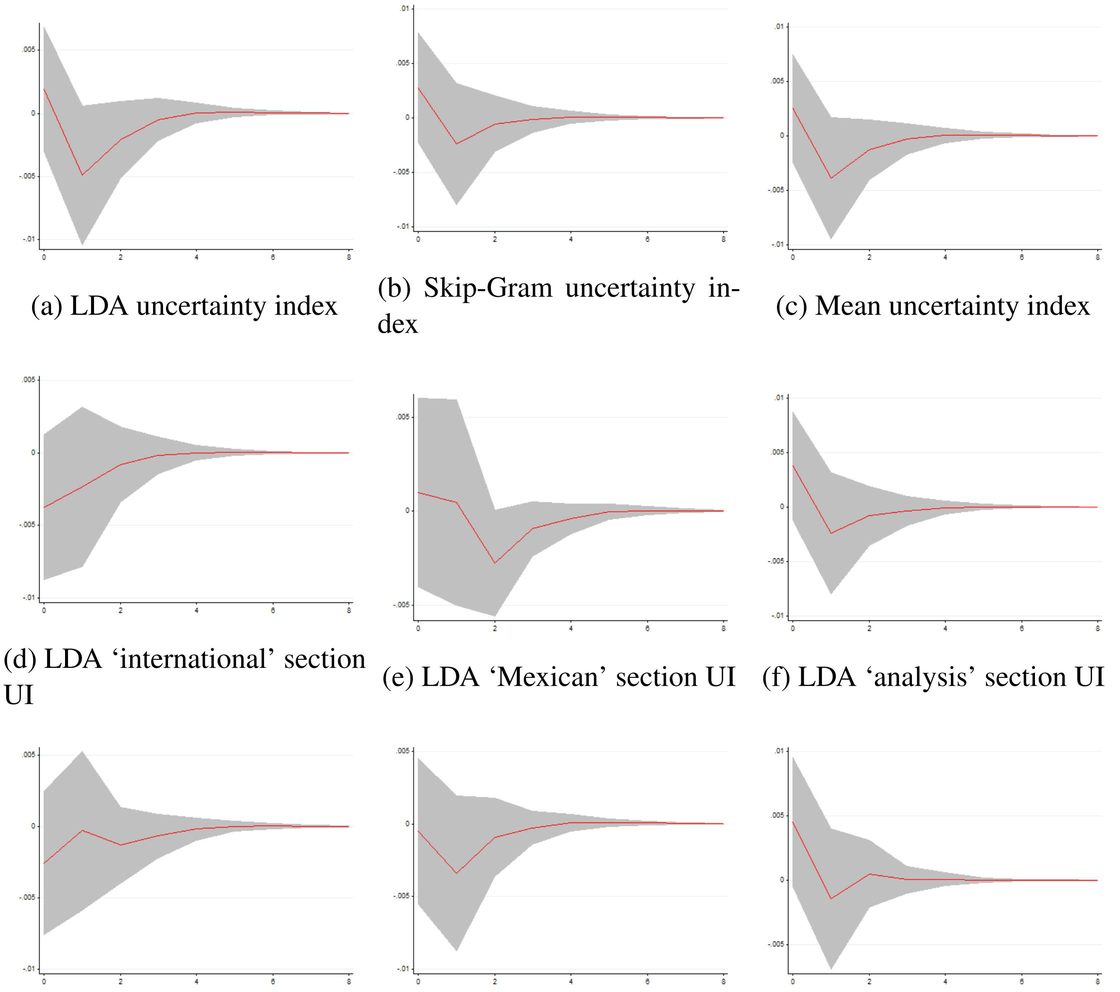
(f) LDA ‘analysis’ section UI

(g) Skip-Gram ‘international’ section UI

(h) Skip-Gram ‘Mexican’ section UI

(i) Skip-Gram ‘analysis’ section UI

Figure A.7: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices for the minutes of the Bank of Mexico for the period 2011-2018. The gray area shows the 95% confidence intervals computed using bootstrapped standard errors (200 replications). The Y -axis is the % change in M3 and the X-axis represents time in months (8 months). The LDA and Skip-Gram ‘international’ section UI refers to the LDA and Skip-Gram uncertainty indices for the ‘description of international economic and financial situation’ section. ‘Mexican’ and ‘analysis’ refer to the other two sections.



(a) LDA uncertainty index

(b) Skip-Gram uncertainty index

(c) Mean uncertainty index

(d) LDA 'international' section UI

(e) LDA 'Mexican' section UI

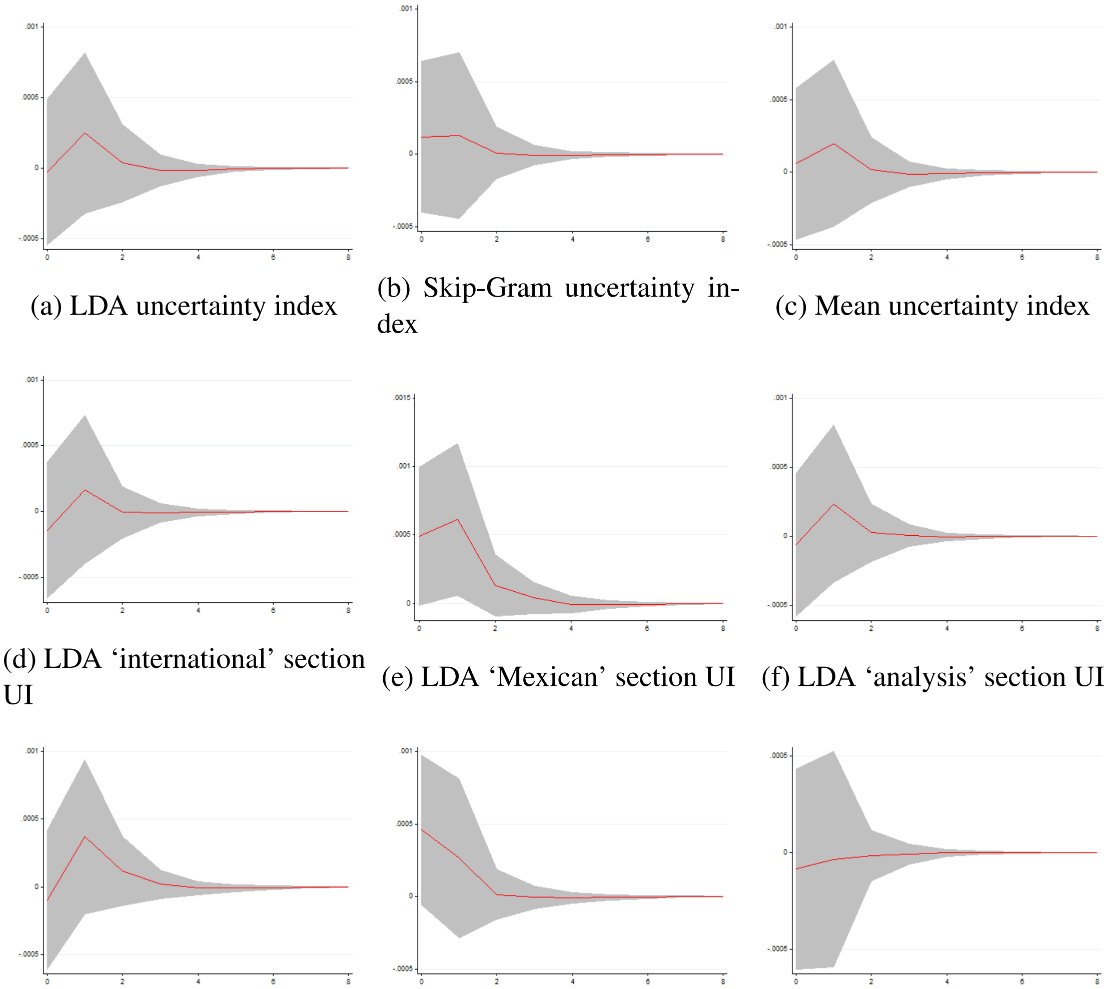
(f) LDA 'analysis' section UI

(g) Skip-Gram 'international' section UI

(h) Skip-Gram 'Mexican' section UI

(i) Skip-Gram 'analysis' section UI

Figure A.8: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices for the minutes of the Bank of Mexico for the period 2011-2018. The gray area shows the 95% confidence intervals computed using bootstrapped standard errors (200 replications). The Y -axis is the % change in the exchange rate and the X-axis represents time in months (8 months). The LDA and Skip-Gram 'international' section UI refers to the LDA and Skip-Gram uncertainty indices for the 'description of international economic and financial situation' section. 'Mexican' and 'analysis' refer to the other two sections.



(a) LDA uncertainty index

(b) Skip-Gram uncertainty index

(c) Mean uncertainty index

(d) LDA 'international' section UI

(e) LDA 'Mexican' section UI

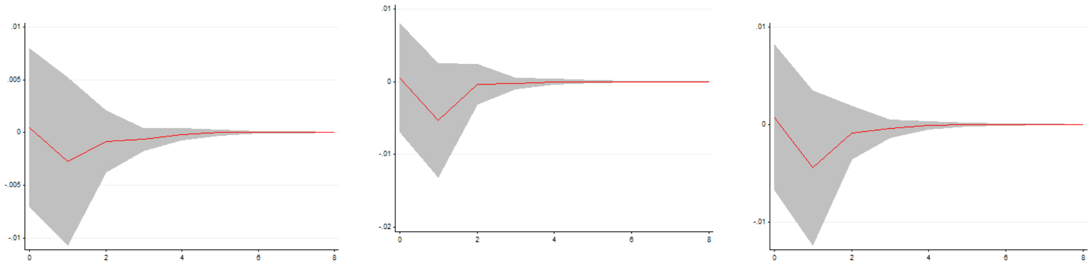
(f) LDA 'analysis' section UI

(g) Skip-Gram 'international' section UI

(h) Skip-Gram 'Mexican' section UI

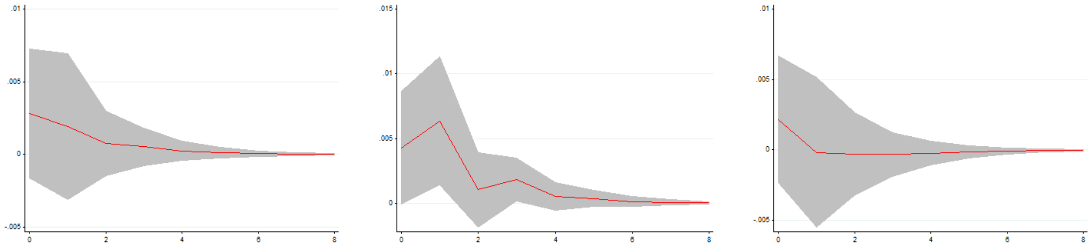
(i) Skip-Gram 'analysis' section UI

Figure A.9: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices for the minutes of the Bank of Mexico for the period 2011-2018. The gray area shows the 95% confidence intervals computed using bootstrapped standard errors (200 replications). The Y -axis is the % change in the consumer price index and the X-axis represents time in months (8 months). The LDA and Skip-Gram 'international' section UI refers to the LDA and Skip-Gram uncertainty indices for the 'description of international economic and financial situation' section. 'Mexican' and 'analysis' refer to the other two sections.



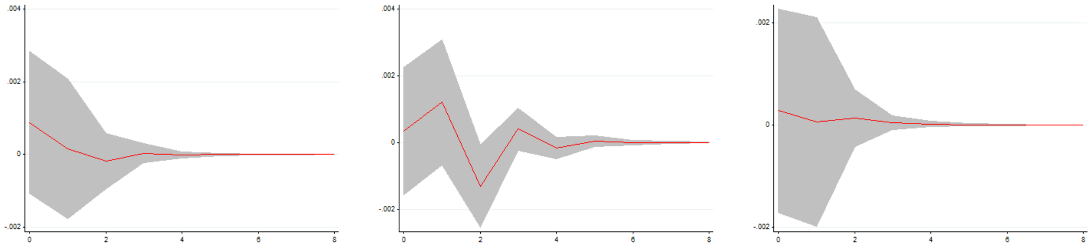
(a) LDA uncertainty index (b) Skip-Gram uncertainty index (c) Mean uncertainty index

Figure A.10: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices for the minutes of the Bank of Mexico for the period 2011-2018. The gray area shows the 95% confidence intervals computed using bootstrapped standard errors (200 replications). The Y -axis is the % change in the target interest rate and the X-axis represents time in months (8 months).



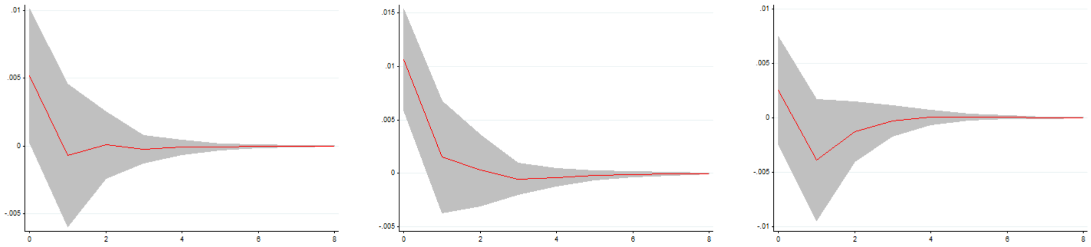
(a) Global EPU index (b) Mexico EPU index (c) Mean uncertainty index

Figure A.11: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices considered for the period 2011-2018. The gray area shows the 95% confidence intervals computed using bootstrapped standard errors (200 replications). The Y -axis is the % change in the monthly inter-bank rate (24 hours) and the X-axis represents time in months (8 months).



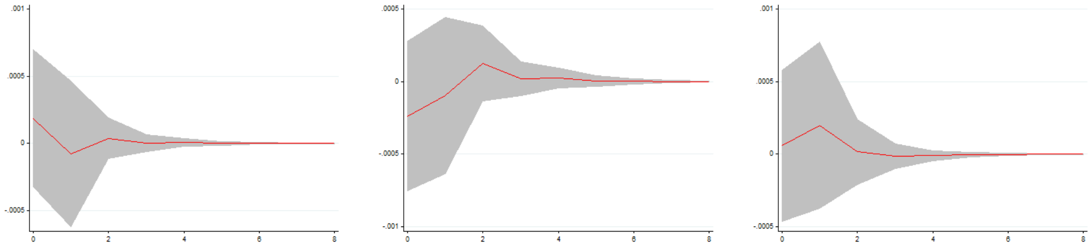
(a) Global EPU index (b) Mexico EPU index (c) Mean uncertainty index

Figure A.12: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices considered for the period 2011-2018. The gray area shows the 95% confidence intervals computed using bootstrapped standard errors (200 replications). The Y -axis is the % change in M3 and the X-axis represents time in months (8 months).



(a) Global EPU index (b) Mexico EPU index (c) Mean uncertainty index

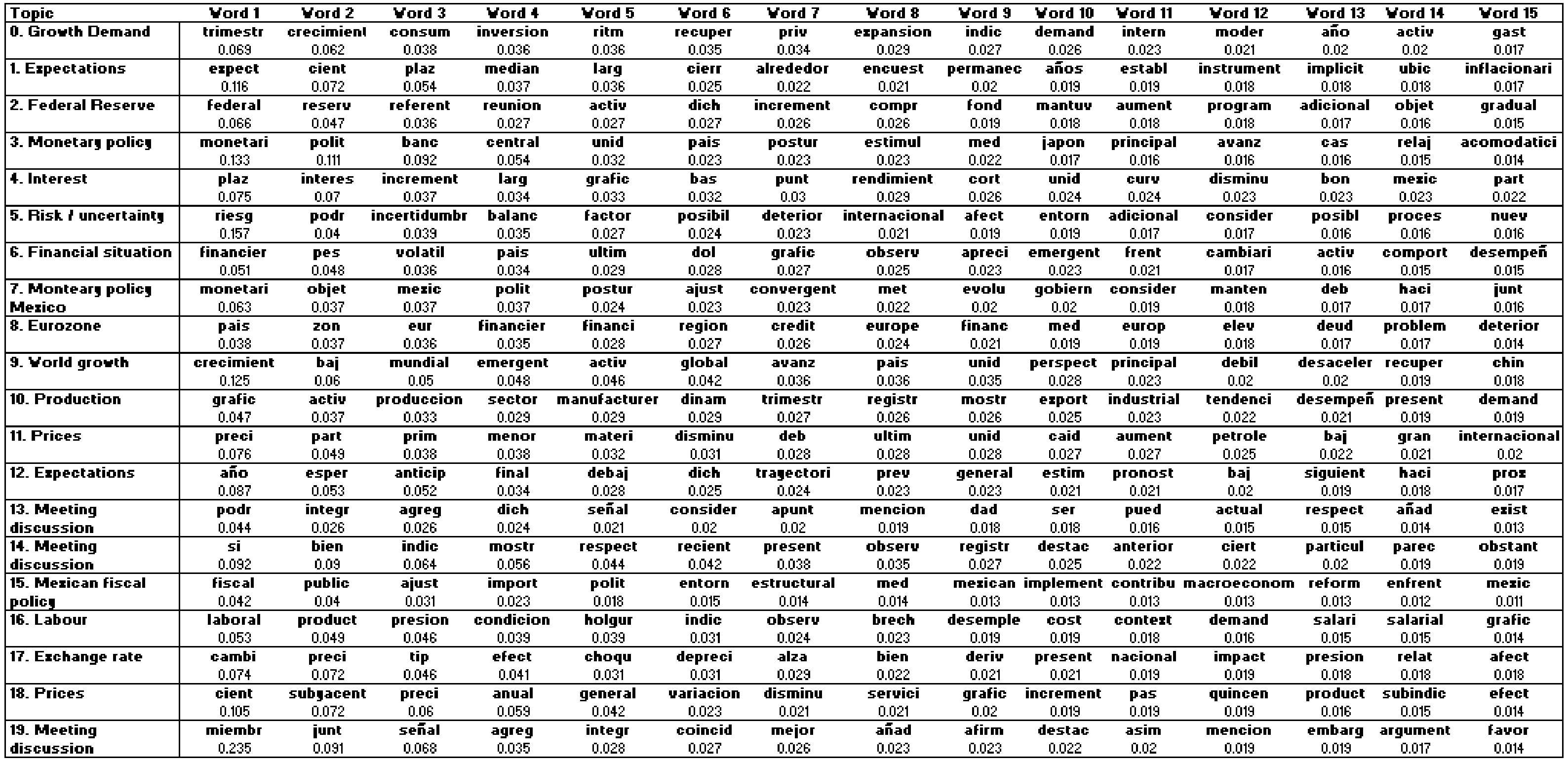
Figure A.13: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices considered for the period 2011-2018. The gray area shows the 95% confidence intervals computed using bootstrapped standard errors (200 replications). The Y -axis is the % change in the exchange rate and the X-axis represents time in months (8 months).



(a) Global EPU index (b) Mexico EPU index (c) Mean uncertainty index

Figure A.14: Impulse response functions from the Structural VAR model corresponding to one standard-deviation in each of the uncertainty indices considered for the period 2011-2018. The gray area shows the 95% confidence intervals computed using bootstrapped standard errors (200 replications). The Y -axis is the % change in the consumer price index and the X-axis represents time in months (8 months).

Table A.1: For each of the twenty topics of the LDA analysis, the table displays the first fifteen words with the highest probability. A description (tag) is proposed for each topic to increase intuition, though they do not affect at all the results of our analysis.



Topic	Vord 1	Vord 2	Vord 3	Vord 4	Vord 5	Vord 6	Vord 7	Vord 8	Vord 9	Vord 10	Vord 11	Vord 12	Vord 13	Vord 14	Vord 15
0. Growth Demand	trimestr	crecimient	consum	inversion	ritm	recuper	priv	expansion	indic	demand	intern	moder	año	activ	gast
	0.069	0.062	0.038	0.036	0.036	0.035	0.034	0.029	0.027	0.026	0.023	0.021	0.02	0.02	0.017
1. Ezpectations	ezpect	cient	plaz	median	larg	cierr	alrededor	encuest	permanec	años	establ	instrument	implicit	ubic	inflacionari
	0.116	0.072	0.054	0.037	0.036	0.025	0.022	0.021	0.02	0.019	0.019	0.018	0.018	0.018	0.017
2. Federal Reserve	federal	reserv	referent	reunion	activ	dich	increment	compr	fond	mantuv	aument	program	adicional	objet	gradual
	0.066	0.047	0.036	0.027	0.027	0.027	0.026	0.026	0.019	0.018	0.018	0.018	0.017	0.016	0.015
3. Monetary policy	monetari	polit	banc	central	unid	pais	postur	estimul	med	japon	principal	avanz	cas	relaj	acomodatici
	0.133	0.111	0.092	0.054	0.032	0.023	0.023	0.023	0.022	0.017	0.016	0.016	0.016	0.015	0.014
4. Interest	plaz	interes	increment	larg	grafic	bas	punt	rendimient	cort	unid	curv	disminu	bon	mezic	part
	0.075	0.07	0.037	0.034	0.033	0.032	0.03	0.029	0.026	0.024	0.024	0.023	0.023	0.023	0.022
5. Risk / uncertainty	riesg	podr	incertidumbr	balanc	factor	posibil	deterior	internacional	afect	entorn	adicional	consider	posibl	proces	nuev
	0.157	0.04	0.039	0.035	0.027	0.024	0.023	0.021	0.019	0.019	0.017	0.017	0.016	0.016	0.016
6. Financial situation	financier	pes	volatil	pais	ultim	dol	grafic	observ	apreci	emergent	frent	cambiari	activ	comport	desempeñ
	0.051	0.048	0.036	0.034	0.029	0.028	0.027	0.025	0.023	0.023	0.021	0.017	0.016	0.015	0.015
7. Monteary policy Mezico	monetari	objet	mezic	polit	postur	ajust	convergent	met	evolu	gobiern	consider	manten	deb	haci	junt
	0.063	0.037	0.037	0.037	0.024	0.023	0.023	0.022	0.02	0.02	0.019	0.018	0.017	0.017	0.016
8. Eurozone	pais	zon	eur	financier	financi	region	credit	europe	financ	med	europ	elev	deud	problem	deterior
	0.038	0.037	0.036	0.035	0.028	0.027	0.026	0.024	0.021	0.019	0.019	0.018	0.017	0.017	0.014
9. Vorld growth	crecimient	baj	mundial	emergent	activ	global	avanz	pais	unid	perspect	principal	debil	desaceler	recuper	chin
	0.125	0.06	0.05	0.048	0.046	0.042	0.036	0.036	0.035	0.028	0.023	0.02	0.02	0.019	0.018
10. Production	grafic	activ	produccion	sector	manufacturer	dinam	trimestr	registr	mostr	ezport	industrial	tendenci	desempeñ	present	demand
	0.047	0.037	0.033	0.029	0.029	0.029	0.027	0.026	0.026	0.025	0.023	0.022	0.021	0.019	0.019
11. Prices	preci	part	prim	menor	materi	disminu	deb	ultim	unid	caid	aument	petrole	baj	gran	internacional
	0.076	0.049	0.038	0.038	0.032	0.031	0.028	0.028	0.028	0.027	0.027	0.025	0.022	0.021	0.02
12. Ezpectations	año	esper	anticip	final	debaj	dich	trayectori	prev	general	estim	pronost	baj	siguient	haci	proz
	0.087	0.053	0.052	0.034	0.028	0.025	0.024	0.023	0.023	0.021	0.021	0.02	0.019	0.018	0.017
13. Meeting discussion	podr	integr	agreg	dich	señal	consider	apunt	mencion	dad	ser	pued	actual	respect	añad	ezist
	0.044	0.026	0.026	0.024	0.021	0.02	0.02	0.019	0.018	0.018	0.016	0.015	0.015	0.014	0.013
14. Meeting discussion	si	bien	indic	mostr	respect	recient	present	observ	registr	destac	anterior	ciert	particul	parec	obstant
	0.092	0.09	0.064	0.056	0.044	0.042	0.038	0.035	0.027	0.025	0.022	0.022	0.02	0.019	0.019
15. Mezican fiscal policy	fiscal	public	ajust	import	polit	entorn	estructural	med	mezican	implement	contribu	macroeconom	reform	enfrent	mezic
	0.042	0.04	0.031	0.023	0.018	0.015	0.014	0.014	0.013	0.013	0.013	0.013	0.013	0.012	0.011
16. Labour	laboral	product	presion	condicion	holgur	indic	observ	brech	desemple	cost	contezt	demand	salari	salarial	grafic
	0.053	0.049	0.046	0.039	0.039	0.031	0.024	0.023	0.019	0.019	0.018	0.016	0.015	0.015	0.014
17. Ezchange rate	cambi	preci	tip	efect	choqu	depreci	alza	bien	deriv	present	nacional	impact	presion	relat	afect
	0.074	0.072	0.046	0.041	0.031	0.031	0.029	0.022	0.021	0.021	0.019	0.019	0.018	0.018	0.018
18. Prices	cient	subyacent	preci	anual	general	variacion	disminu	servici	grafic	increment	pas	quincen	product	subindic	efect
	0.105	0.072	0.06	0.059	0.042	0.023	0.021	0.021	0.02	0.019	0.019	0.019	0.016	0.015	0.014
19. Meeting discussion	miembr	junt	señal	agreg	integr	coincid	mejor	añad	afirm	destac	asim	mencion	embarg	argument	favor
	0.235	0.091	0.068	0.035	0.028	0.027	0.026	0.023	0.023	0.022	0.02	0.019	0.019	0.017	0.014

Chapter 4

Supplementary Material - Monetary Policy Uncertainty in Mexico: An Unsupervised Approach

4.1 Minutes of Banxico Database

The board of governors of the Central Bank of Mexico (aka Bank of Mexico or Banxico) meets eight times a year to set the interest rate. This paper studies the Spanish version of the minutes of the board governors published in the period 2011-2018. We extract the PDF files of the minutes of the Banxico from the Central Bank of Mexico's web page.[1]

The minutes are divided in different sections and subsections. We process this division manually by assigning to each paragraph a tag identifying the corresponding section and subsection. First, the section 'description of the international economic and financial situation' presents mostly the economic and financial situation in important economies such as the United States, Europe, Japan and China. The section combines two subsections, one describing international economic activity and the other international financial activity.

The next section describes the economic, financial and inflation situation in Mexico. It is also a combination of three subsections, describing Mexican economic activity, Mexican financial activity and the situation of inflation in Mexico.

The third section illustrates the discussion of the board members concerning the economic, financial and inflation situation abroad and in Mexico. This section also includes the discussion of board members leading to the monetary policy decision.

[1] https://www.banxico.org.mx/publicaciones-y-prensa/anuncios-minutas-tasa-objetiv.html

The final section briefly explains the final decision of the board of governors. Since the minutes numbered 59 (in 2018), the minutes of the Bank of Mexico have included a new section titled ‘voting’ which publishes the vote of each member of the board. Also, since then, the minutes have included a new section titled ‘dissenting opinions’ in which board members who voted against the majority explain their reasons.

The text database of the minutes of Banxico database is included in the supplementary material with the name ‘Banxico_minutes.txt’. This database comprises four columns with diverse information of the paragraphs:

- **minutes**: this tag indicates the number of the meeting.
- **section**: this column distinguish the main sections of the minutes. The value ‘0’ corresponds to the ‘international financial and economic’ section. Value ‘1’ corresponds to the ‘Mexican financial, economic and inflation’ section. Value ‘2’ corresponds to the ‘analysis and rationale behind the voting of the governing boards’ section. Value ‘3’ corresponds to the ‘monetary policy decision’ section. Value ‘4’ corresponds to the ‘voting’ section and Value ‘5’ corresponds to the ‘dissenting opinion’ section.
- **subsection**: This column specifies the subsection. Values ‘0’, ‘1’ and ‘2’ correspond to the paragraphs of financial, economic and inflation subsections, respectively. Value ‘22’ corresponds to the ‘analysis and rationale behind the voting of the governing boards’ section and Value ‘33’ to the ‘monetary policy section’ section. Value ‘44’ corresponds to the ‘voting’ section and Value ‘55’ corresponds to the ‘dissenting opinion’ section.
- **speech**: This column contains the text database of the minutes of the Central Bank of Mexico.

We then create Figure 1 of the paper which shows the total number of words in the different sections of the minutes. To create Figure 1 of the paper, we create the database ‘mexico_minutes_database.csv’ that contains the date of the meetings and the release date of the minutes. The python code to construct Figure 1 of the paper, ‘Banxico_minutes_countwords.py’, is included in the complementary material folder and it is shown below:

```python
import pandas as pd
import matplotlib.pyplot as plt
from pylab import *
import matplotlib.patches as mpatches
from matplotlib import pyplot
import Pyro4
import seaborn as sns

#Importing Banxico minutes database as 'df' DataFrame.
df = pd.read_table("Banxico_minutes.txt", encoding="utf-8")

#Defining function to create a column in the DataFrame with
 ↪  tags for the different subsections.
def label_part (row):
    if (row['section'] == 0 and row['subsection'] == 0) :
        return 0
    elif (row['section'] == 0 and row['subsection'] == 1) :
        return 1
    elif (row['section'] == 1 and row['subsection'] == 0) :
        return 2
    elif (row['section'] == 1 and row['subsection'] == 1) :
        return 3
    elif (row['section'] == 1 and row['subsection'] == 2) :
        return 4
    elif row['section'] == 2 :
        return 5
    elif row['section'] == 3 :
        return 6
    elif row['section'] == 4 :
        return 7
    elif row['section'] == 5 :
        return 8
    else:
        return 'nan'

#Creating 'all_parts' column with tags for each subsection.
df['all_parts'] = df.apply (lambda row: label_part(row),
 ↪  axis=1)

#Creating column 'TotalWordCount' in DataFrame 'df' for
 ↪  total word count.
df = pd.concat([df, pd.DataFrame(columns =
 ↪  ['TotalWordCount'])])

#Counting total number of words in each paragraph of the
 ↪  minutes.
```

```
for i,article in enumerate(df.speech):
    if str(article) != 'nan':
        df.TotalWordCount[i] = len(article.split(' '))

#Creating a new DataFrame 'df_min' with only the columns
 ↪  'minutes','TotalWordCount' and 'all_parts'.
df_min =
 ↪  df[['minutes','TotalWordCount','all_parts']].copy()

#Saving DataFrame 'df_min' in csv.
df_min.to_csv("Mexico_CountWords_uncertainty.csv")

#Grouping the total number of words by minutes and
 ↪  subsections in a new DataFrame 'temp_total'.
temp_total = df_min.groupby(['minutes', 'all_parts'])[
 ↪  'TotalWordCount'].sum().reset_index().rename(columns =
 ↪  {'CombScore':'combsum'})

#Importing date of the minutes of the Central Bank of
 ↪  Mexico as 'date' DataFrame.
date = pd.read_csv("mexico_minutes_date.csv", sep = ';',
 ↪  encoding = "utf-8")

#Merging 'temp_total' DataFrame with 'date' DataDrame in a
 ↪  new DataFrame named 'minutes_date'.
minutes_date = pd.merge(temp_total, date,  how='left',
 ↪  left_on=['minutes'], right_on = ['minutes'])

#Changing format of the 'date' column from object to
 ↪  datetime64[ns]
minutes_date['datedecision'] =
 ↪  pd.to_datetime(minutes_date['datedecision'],
 ↪  infer_datetime_format = True, dayfirst = True)

#Setting 'date' column as index of the DataFrame
 ↪  'minutes_date'.
minutes_date = minutes_date.set_index('datedecision')

#Converting  format of 'TotalWordCount' column from object
 ↪  to int64 in order to apply resample.
```

```python
minutes_date["TotalWordCount"] = pd.to_numeric(minutes_date["TotalWordCount"])

#Checking the format of the DataFrame 'minutes_date'.
minutes_date.dtypes

#Constructing a DataFrame for the section 'international economic activity'.
all_part_0 = minutes_date[minutes_date.all_parts == 0].copy()

#Creating copy DataFrame of the section 'international economic activity'.
all_part_0['total_words_zero'] = all_part_0['TotalWordCount'].copy()

#Creating DataFrame of the section 'international economic activity' with only the total word count.
all_part_0_min = all_part_0[['total_words_zero']].copy()

#Constructing DataFrame for the section 'international financial activity'.
all_part_1 = minutes_date[minutes_date.all_parts == 1].copy()

#Creating copy DataFrame of the section 'international financial activity'.
all_part_1['total_words_one'] = all_part_1['TotalWordCount'].copy()

#Creating DataFrame of the section 'international financial activity' with only the total word count.
all_part_1_min = all_part_1[['total_words_one']].copy()

#Constructing DataFrame for the section 'Mexican economic activity'.
all_part_2 = minutes_date[minutes_date.all_parts == 2].copy()

```

```
#Creating copy DataFrame of the section 'Mexican economic
 ↪  activity'.
all_part_2['total_words_two'] =
 ↪  all_part_2['TotalWordCount'].copy()

#Creating DataFrame of the section 'Mexican economic
 ↪  activity' with only the total word count.
all_part_2_min = all_part_2[['total_words_two']].copy()

#Constructing DataFrame for the section 'Mexican financial
 ↪  activity'.
all_part_3 = minutes_date[minutes_date.all_parts ==
 ↪  3].copy()

#Creating copy DataFrame of the section 'Mexican financial
 ↪  activity'.
all_part_3['total_words_three'] =
 ↪  all_part_3['TotalWordCount'].copy()

#Creating DataFrame of the section 'Mexican financial
 ↪  activity' with only the total word count.
all_part_3_min = all_part_3[['total_words_three']].copy()

#Constructing DataFrame for the section 'Mexican
 ↪  inflation'.
all_part_4 = minutes_date[minutes_date.all_parts ==
 ↪  4].copy()

#Creating copy DataFrame of the section 'Mexican
 ↪  inflation'.
all_part_4['total_words_four'] =
 ↪  all_part_4['TotalWordCount'].copy()

#Creating DataFrame of the section 'Mexican inflation' with
 ↪  only the total word count.
all_part_4_min = all_part_4[['total_words_four']].copy()

#Constructing DataFrame for the section 'analysis and
 ↪  rationale behind the voting of the governing boards'.
```

```
all_part_5 = minutes_date[minutes_date.all_parts ==
  ↪ 5].copy()

#Creating copy DataFrame of the section 'analysis and
  ↪ rationale behind the voting of the governing boards'.
all_part_5['total_words_five'] =
  ↪ all_part_5['TotalWordCount'].copy()

#Creating DataFrame of the section 'analysis and rationale
  ↪ behind the voting of the governing boards' with only
  ↪ the total word count.
all_part_5_min = all_part_5[['total_words_five']].copy()

#Constructing DataFrame for the section 'monetary policy
  ↪ decision'.
all_part_6 = minutes_date[minutes_date.all_parts ==
  ↪ 6].copy()

#Creating copy DataFrame of the section 'monetary policy
  ↪ decision'.
all_part_6['total_words_six'] =
  ↪ all_part_6['TotalWordCount'].copy()

#Creating DataFrame of the section 'monetary policy
  ↪ decision' with only the total word count.
all_part_6_min = all_part_6[['total_words_six']].copy()

#Constructing DataFrame for the section 'voting'.
all_part_7 = minutes_date[minutes_date.all_parts ==
  ↪ 7].copy()

#Creating copy DataFrame of the section 'voting'.
all_part_7['total_words_seven'] =
  ↪ all_part_7['TotalWordCount'].copy()

#Creating DataFrame of the section 'voting' with only the
  ↪ total word count.
all_part_7_min = all_part_7[['total_words_seven']].copy()

```

```python
#Constructing DataFrame for the section 'dissenting opinions'.
all_part_8 = minutes_date[minutes_date.all_parts == 8].copy()

#Creating copy DataFrame of the section 'dissenting opinions'.
all_part_8['total_words_eight'] = all_part_8['TotalWordCount'].copy()

#Creating DataFrame of the section 'dissenting opinions' with only the total word count.
all_part_8_min = all_part_8[['total_words_eight']].copy()

##################################
# MERGING ALL SUBSECTIONS DATAFRAMES #
##################################

mix_1 =  pd.merge(all_part_0_min, all_part_1_min, left_index=True, right_index=True)
mix_2 =  pd.merge(mix_1, all_part_2_min, left_index=True, right_index=True)
mix_3 =  pd.merge(mix_2, all_part_3_min, left_index=True, right_index=True)
mix_4 =  pd.merge(mix_3, all_part_4_min, left_index=True, right_index=True)
mix_5 =  pd.merge(mix_4, all_part_5_min, left_index=True, right_index=True)
mix_6 =  pd.merge(mix_5, all_part_6_min, left_index=True, right_index=True)
mix_7 =  pd.merge(mix_6, all_part_7_min,  how='left', left_index=True, right_index=True)
mix_total_words =  pd.merge(mix_7, all_part_8_min, how='left', left_index=True, right_index=True).copy()


##################################
#COUNT TOTAL WORDS PER SUBSECTION GRAPH #
##################################

```

```python
# Use seaborn style defaults and set the default figure size.
sns.set(rc={'figure.figsize':(14, 10)})


mix_total_words['total_words_zero'].plot(color='red')
mix_total_words['total_words_one'].plot(color='yellow')
mix_total_words['total_words_two'].plot(color='green')
mix_total_words['total_words_three'].plot(color='blue')
mix_total_words['total_words_four'].plot(color='pink')
mix_total_words['total_words_five'].plot(color='orange')
mix_total_words['total_words_six'].plot(color='black')
mix_total_words['total_words_seven'].plot(color='purple')
mix_total_words['total_words_eight'].plot(color='brown')

axvline('2016-09-29', color='red', ls="dotted")
axvline('2018-05-17', color='red', ls="dotted")

plt.ylabel("Total number of words of each part of the minutes")
plt.xlabel("Minutes across time")

red_patch = mpatches.Patch(color='red', label='Description of international economic activity')
yellow_patch = mpatches.Patch(color='yellow', label='Description of international financial activity')
green_patch = mpatches.Patch(color='green', label='Description of Mexican economic activity')
blue_patch = mpatches.Patch(color='blue', label='Description of  Mexican financial activity')
pink_patch = mpatches.Patch(color='pink', label='Description of Mexican inflation')
orange_patch = mpatches.Patch(color='orange', label='Analysis and rationale behind the voting of the governing boards')
black_patch = mpatches.Patch(color='black', label='Monetary policy decission')
purple_patch = mpatches.Patch(color='purple', label='Voting')
```

```
brown_patch = mpatches.Patch(color='brown',
 ↪  label='Dissenting opinions')

plt.legend(handles=[red_patch, yellow_patch, green_patch,
 ↪  blue_patch, pink_patch, orange_patch, black_patch,
 ↪  purple_patch, brown_patch],loc='center left',
 ↪  bbox_to_anchor=(0, 0.85))
```

4.2 Latent Dirichlet Allocation

This section shows the python code to estimate Latent Dirichlet Allocation. As text data, we use the Spanish version of the minutes of the Bank of Mexico.

To apply Latent Dirichlet Allocation with Spanish language, we use the python code provided by the Professor Stephen Hansen as in the first chapter.[2] The 'cleaning' data process for LDA requires three steps to eliminate non-relevant information from the text. The first step is to remove the punctuation and stop words such as 'the', 'all', 'because', 'this', not relevant since they provide no information about the theme of the paragraph. The second step is to stem the remaining words. Stemming is a process that consists in reducing words into their word stem or base root. Finally, we rank these stems according to the term frequency-inverse document frequency (tf-idf). However, the code of Professor Stephen Hansen does not include the first two steps for texts that are in Spanish. We build a python code to delete the stop words and stem texts in Spanish language. Besides, we use the version of python 3.7 since the version of python 2.7 is not capable of reading Spanish characters such as 'ñ' or 'è'. The following python code shows the adaption to the Spanish language of the code of Stephen Hansen.

```
import topicmodels
import string
import numpy as np
import nltk; nltk.download('stopwords')
import re
import numpy as np
import pandas as pd
from pprint import pprint
import gensim
import gensim.corpora as corpora
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel
```

[2] https://github.com/sekhansen

```python
import pyLDAvis
import pyLDAvis.gensim  # don't skip this
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore",category=DeprecationWarning)
from nltk.stem import SnowballStemmer
#stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
from nltk.tokenize import sent_tokenize, word_tokenize

# Run in python console:
#import nltk; nltk.download('stopwords')

#We import the dataset of stopwords of NLTK in Spanish and we include extra stopwords.
from nltk.corpus import stopwords
stop_words = stopwords.words('spanish')
stop_words.extend(['meses','febrero','marzo','abril','junio','julio','agosto','septiembre','noviembre','diciembre','octubre','mayo','enero','un','uno','una','dos','tres','cuatro','cinco','seis','siete','ocho','nueve','diez','primer','primera','segundo','segunda','tercer','tercero','primero','tercera','cuarto','cuarta','quinto','quinta','sexto', 'sexta','septimo','septima','octavo','octava','noveno','novena','decimo', 'decima'])

#We import the dataset of the minutes of the Bank of Mexico as 'df' DataFrame.
df = pd.read_csv('Banxico_minutes.txt', sep='\t', encoding="utf-8")

#We pass the 'speech' column of the 'df' DataFrame to list format.
data = df.speech.values.tolist()

#We remove punctuation signs, numbers and non-relevant characters.
data = [re.sub('\S*@\S*\s?', '', sent) for sent in data]
data = [re.sub('\s+', ' ', sent) for sent in data]
data = [re.sub("\'", "", sent) for sent in data]
pprint(data[:1])
```

```

#Defining function to pass list of strings to list of lists.
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=False))  # deacc=True removes punctuations

#Passing 'data' list format from list of strings to list of lists.
data_words = list(sent_to_words(data ))

print(data_words[:1])

#Defining remove stop words function.
def remove_stopwords(texts):
    return [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for doc in texts]

#Defining function for stemming in Spanish language.
porter = SnowballStemmer("spanish")
def stemSentence(sentence):
    token_words=word_tokenize(sentence)
    token_words
    stem_sentence=[]
    for word in token_words:
        stem_sentence.append(porter.stem(word))
        stem_sentence.append(" ")
    return "".join(stem_sentence)


#Removing the stop words.
data_words_nostops = remove_stopwords(data_words)

#Stemming process, we change the format of the text to adapt it to the stemming function. Once the text is stemmed, we change the format again to the one accepted by the LDA functions of the code provided by Stephen Hansen.
```

```
implodeList = []

for item in data_words_nostops :
    implodeList.append(' '.join(item))

with open('data_lda_mexico_withoustop.txt', 'w',
 ↪  encoding="utf-8") as f:
    for item in implodeList:
        f.write("%s\n" % item)

file=open("data_lda_mexico_withoustop.txt",
 ↪  encoding="utf-8")
my_lines_list=file.readlines()
my_lines_list

print(my_lines_list[0])
print("Stemmed sentence")
x=stemSentence(my_lines_list[0])
print(x)

#Stemming the minutes text.
stem_file=open("mexicostem.txt",mode="w", encoding="utf-8")
for word in my_lines_list:
    stem_sentence=stemSentence(word)
    stem_file.write("%s\n" % stem_sentence)

file=open("mexicostem.txt", "r",newline = "\n",
 ↪  encoding="utf-8")
data_chile_stem=file.readlines()

#We include the stemmed and cleaned dataset in the column
 ↪  'bigrams' of the DataFrame 'data'.
data['bigrams'] = data_chile_stem

#Including the column 'bigrams' of the DataFrame 'data' in
 ↪  the code of  Prof. Hansen.
docsobj = topicmodels.RawDocs(data.bigrams, "long")
docsobj.token_clean(1)
```

```python
# we rank these stems according to the term frequency-inverse document frequency (tf-idf).
docsobj.term_rank("tokens")

#We disregard all stems that have a value of the tf-idf ranking of 2,600 or lower.
docsobj.rank_remove("tfidf", "tokens", docsobj.tfidf_ranking[2600][1])

#Plotting the tfidf ranking.
plt.plot([x[1] for x in docsobj.tfidf_ranking])

#Printing number of unique and total stems in the database.
all_stems = [s for d in docsobj.tokens for s in d]
print("number of unique stems = %d" % len(set(all_stems)))
print("number of total stems = %d" % len(all_stems))

#Latent Dirichelt Allocation application with 20 topics.
ldaobj = topicmodels.LDA.LDAGibbs(docsobj.tokens, 20)

#we run twice 20 samples from points in the chain that are thinned with a thinning interval of 50.
ldaobj.sample(500, 50, 20)
print(ldaobj.perplexity())
ldaobj.sample(500, 50, 20)
print(ldaobj.perplexity())

ldaobj.samples_keep(4)
ldaobj.topic_content(20)

dt = ldaobj.dt_avg()
tt = ldaobj.tt_avg()
ldaobj.dict_print()

data = data.drop('bigrams', 1)

#LDA output: topics per document.
for i in range(ldaobj.K):
    data['T' + str(i)] = dt[:, i]
data.to_csv("document_topic_mexico.csv", index=False)
```

```

#Querying documents by minutes. LDA output: topics per
 ↪  minutes.
data['bigrams'] = [' '.join(s) for s in docsobj.tokens]
aggspeeches = data.groupby(['minutes'])['bigrams'].\
    apply(lambda x: ' '.join(x))
aggdocs = topicmodels.RawDocs(aggspeeches)

queryobj = topicmodels.LDA.QueryGibbs(aggdocs.tokens,
 ↪  ldaobj.token_key,
                                      ldaobj.tt)
queryobj.query(10)
queryobj.perplexity()
queryobj.query(30)
queryobj.perplexity()

dt_query = queryobj.dt_avg()
aggdata = pd.DataFrame(dt_query, index=aggspeeches.index,
                       columns=['T' + str(i) for i in
                        ↪  range(queryobj.K)])
aggdata.to_csv("agg_mexico.csv")

#Querying documents by sections. LDA output: topics per
 ↪  sections.
data['bigrams'] = [' '.join(s) for s in docsobj.tokens]
aggspeeches1 =
 ↪  data.groupby(['minutes','section'])['bigrams'].\
    apply(lambda x: ' '.join(x))
aggdocs1 = topicmodels.RawDocs(aggspeeches1)

queryobj1 = topicmodels.LDA.QueryGibbs(aggdocs1.tokens,
 ↪  ldaobj.token_key,
                                      ldaobj.tt)
queryobj1.query(10)
queryobj1.perplexity()
queryobj1.query(30)
queryobj1.perplexity()

dt_query1 = queryobj1.dt_avg()
```

```
aggdata1 = pd.DataFrame(dt_query1,
 ↪  index=aggspeeches1.index,
                                columns=['T' + str(i) for i in
                                 ↪  range(queryobj.K)])
aggdata1.to_csv("agg_mexico_section.csv")
```

The results are not reproducible. However, the results tend always to be similar after several trials. The following list shows the name of the python code and the different outputs included in the supplementary material folder. An explanation of each document is given within brackets.

1. ‘Mexico_LDA.py’ (LDA python code);
2. ‘Topic description.csv’ (LDA output: words per topic);
3. ‘document_topic_mexico.csv’ (LDA output: topics per document);
4. ‘agg_mexico.csv’ (LDA output: topics per minutes);
5. ‘agg_mexico_section.csv’ (LDA output: topics per sections);
6. ‘df_ranking.csv’ (LDA output: ranking of stems by the document frequency);
7. ‘tfidf_ranking.csv’ (LDA output: ranking of stems by the tf-idf measure).

4.3 Skip-Gram and K-Means

This paper estimates the Skip-Gram model and K-Means with the Spanish version of the minutes of Banxico. This section does not show the python code to estimate the Skip-Gram model and K-Means to avoid repetition since it is almost identical to the python code of the first chapter of the thesis. Nonetheless, the python code is included in the complementary material folder with the name ‘mexico_skipgram_k145_s200_w10_big10.py’.

As for LDA, we use python 3.7 to estimate the Skip-Gram model since it recognizes characters of the Spanish language such as ‘ñ’ that are not recognized by python 2.7. To make the results reproducible in python 3.7, we set the seed such as ‘set PYTHONASHSEED=0’ in the terminal before opening python . We then open python from the terminal to estimate the Skip-Gram and K-Means.

The complementary material folder comprises the lists of words of the clusters of the words ‘incertidumbre’ (uncertainty), ‘incierto’ (uncertain), ‘inquietud’ (unrest or concern) and ‘riesgo’ (risk). Moreover, we include all the words of all above clusters into one excel document. The documents included in the supplementary material are described in the following list:

1. ‘incertidumbre_list_words_k145_s200_w10_big10.xlsx’ (List of words of the cluster of the word ‘incertidumbre’).
2. ‘incierto_list_words_k145_s200_w10_big10.xlsx’ (List of words of the cluster of the word ‘incierto’).
3. ‘inquietud_list_words_k145_s200_w10_big10.xlsx’ (List of words of the cluster of the word ‘inquietud’).
4. ‘riesgo_list_words_k145_s200_w10_big10.xlsx’ (List of words of the cluster of the word ‘riesgo’).
5. ‘mexico_list_uncertainty_words_all_clusters_k145_w10_s200.xlsx’ (Combination of the words of the clusters of the words ‘incertidumbre’, ‘incierto’, ‘inquietud’ and ‘riesgo’).

4.4 Uncertainty Indices

This section shows the python code to construct the LDA and the Skip-Gram uncertainty indices for the minutes and the sections. We then combine the LDA and the Skip-Gram uncertainty indices to build the ‘mean uncertainty index’. Moreover, we show the python code to create Figures A.1, A.2, A.3, A.4, A.5 of the paper that show the evolution of the uncertainty indices.

4.4.1 LDA uncertainty indices

We use the probability in the minutes of topic 5 related to ‘uncertainty’, as the LDA uncertainty index. We also construct different uncertainty indices for the various sections to understand the main sources of uncertainty in the minutes. This section shows the python code to construct the LDA uncertainty index for the minutes that is included in the supplementary material folder with the name ‘Banxico_lda_uncertainty_index.py’.

```python
import pandas as pd
import numpy as np

```

```
#Loading the  LDA output 'topics per minutes' as the
 ↪  DataFrame 'minutes'.
minutes = pd.read_csv("agg_mexico.csv", encoding="utf-8")

#Making copy DataFrame 'minutes' with the name
 ↪  'minutes_zero'.
minutes_zero = minutes.copy()

#Loading the database of the date of the minutes of Banxico
 ↪  as the DataFrame 'date'.
date = pd.read_csv("mexico_minutes_date.csv", sep = ';',
 ↪  encoding = "utf-8")

#Merging DataFrame 'minutes_zero' with the DataFrame 'date'
 ↪  in a new DataFrame named 'minutes_date_zero'.
minutes_date_zero = pd.merge(minutes_zero, date,
 ↪  how='left', left_on=['minutes'], right_on =
 ↪  ['minutes'])

#Changing the format of the 'datedecision' column from
 ↪  object to datetime64[ns]. In particular, we take the
 ↪  date in which the meeting took place ('datedecision')
 ↪  and not the release date of the minutes.
minutes_date_zero['datedecision'] =
 ↪  pd.to_datetime(minutes_date_zero['datedecision'],
 ↪  infer_datetime_format =True,dayfirst=True)

#Checking the format of the DataFrame 'minutes_date_zero'.
minutes_date_zero.dtypes

#Setting 'datedecision' column as index of the DataFrame
 ↪  'minutes_date_zero'.
minutes_date_zero =
 ↪  minutes_date_zero.set_index('datedecision')
minutes_date_zero.head(3)

#Creating copy DataFrame 'minutes_date_zero' with the name
 ↪  DataFrame 'month_df'.
month_df = minutes_date_zero.copy()

```

```
#Creating LDA uncertainty index as a column of the
 ↪  DataFrame 'month_df'.
month_df['unc_lda_norm'] = (100 * month_df['T5']) /
 ↪  month_df["T5"].mean()

#We create a new DataFrame 'month' that includes the months
 ↪  that do not have observations.
month = month_df.resample('MS').sum()

#We replace the values of  the 'unc_lda_norm' column  with
 ↪  zero values instead of nan.
month['unc_lda_norm'] = month['unc_lda_norm'].replace(0,
 ↪  np.nan)

#Replacing the values of the column 'unc_lda_norm_total'
 ↪  that have the value 0 with the values of the previous
 ↪  observation.
month['unc_lda_norm'] =
 ↪  month['unc_lda_norm'].fillna(method='ffill')

#Creating the column 'unc_lda_norm_total' in the DataFrame
 ↪  'month' to assign a new name to the LDA uncertainty
 ↪  index.
month['unc_lda_norm_total'] = month['unc_lda_norm'].copy()

#Creating the DataFrame 'month_min' only with the
 ↪  'unc_lda_norm_total' column.
month_min = month[['unc_lda_norm_total']].copy()

#Saving LDA uncertainty index of the 'month_min' DataFrame
 ↪  in a csv file.
month_min.to_csv("final_mexico_unc_lda_k20_2600_500_part
 ↪  _total.csv")
```

The following list comprises the python codes and the csv output files of the LDA uncertainty indices that are included in the supplementary material folder.

1. ‘final_mexico_unc_lda_k20_2600_500_part_total.csv’ (LDA uncertainty index of the minutes);

2. ‘Banxico_lda_uncertainty_index_section_0.py’ (Python code to construct the LDA

uncertainty index of the ‘international financial and economic’ section);

3. ‘final_mexico_unc_lda_k20_2600_500_part_zero.csv’ (Excel file that comprises the LDA uncertainty index of the ‘international financial and economic’ section);
4. ‘Banxico_lda_uncertainty_index_section_1.py’ (Python code to construct the LDA uncertainty index of the ‘Mexican financial, economic and inflation’ section);
5. ‘final_mexico_unc_lda_k20_2600_500_part_one.csv’ (Excel file that comprises the LDA uncertainty index of the ‘Mexican financial, economic and inflation’ section);
6. ‘Banxico_lda_uncertainty_index_section_2.py’ (Python code to construct the LDA uncertainty index of the ‘analysis and rationale behind the voting of the governing boards’ section);
7. ‘final_mexico_unc_lda_k20_2600_500_part_two.csv’ (Excel file that comprises the LDA uncertainty index of the ‘analysis and rationale behind the voting of the governing boards’ section);
8. ‘Banxico_lda_uncertainty_index_section_3.py’ (Python code to construct the LDA uncertainty index of the ‘monetary policy decision’ section);
9. ‘final_mexico_unc_lda_k20_2600_500_part_three.csv’ (Excel file that comprises the LDA uncertainty index of the ‘monetary policy decision’ section).

4.4.2 Skip-Gram uncertainty indices

This section shows the python code to count the frequency of the words of the ‘uncertainty’ dictionary in the minutes. We then construct the Skip-Gram uncertainty index for the whole minutes and for each of the four main sections. Here, we only show the python code - ‘Skip-Gram uncertainty index - whole minutes.py’ - to build the Skip-Gram uncertainty index for the minutes:

```python
import re
import numpy as np
import pandas as pd
import pickle

#Loading the database of the minutes of Banxico as the DataFrame 'df'.
df = pd.read_table("Banxico_minutes.txt", encoding="utf-8")

```

```python
#We import the 'cleaned' dataset of the minutes of the Central Bank of Mexico and we include it as column 'clean' in the 'df' DataFrame.
with open ('mexico_wor2vec_order', 'rb') as fp:
    df['clean'] = pickle.load(fp)

#We import the 'uncertainty' dictionary obtained in the Skip-Gram and K-Means model as the DataFrame 'data'.
data = pd.read_csv("mexico_list_uncertainty_words_all_clusters_k145_w10_s200.csv", sep = ",", encoding="utf-8")

#We change the format of the words of the 'uncertainty' dictionary from list of lists to list of strings the list. Then, we pass the letters to upper capital letters.
uncer_index = data['words']
implodeList =list(uncer_index)

#Passing from low to upper capital letters.
uncertainty = []
for word in implodeList:
   uncertainty.append(word.upper())
print(uncertainty)

# We create two columns in the DataFrame called 'UncerScore' and 'TotalWordCount' for the total number of uncertainty number of words and the total word count column respectively.
df = pd.concat([df, pd.DataFrame(columns = ['UncerScore']),
                  pd.DataFrame(columns = ['TotalWordCount'])])

#Counting the number of uncertainty and total number of words.
bow_uncer = []

for i,article in enumerate(df.clean):
    if str(article) != 'nan':
        m = 0
        for word in article.split(' '):
               if word.upper() in uncertainty:
                    m+= 1
                    bow_uncer.append(word)
        df.UncerScore[i]      = m
        df.TotalWordCount[i] = len(article.split(' '))

#Creating new DataFrame 'df_min' only with the columns: 'minutes', 'UncerScore' and 'TotalWordCount'.
df_min = df[['minutes','UncerScore', 'TotalWordCount']].copy()

#Grouping the minutes by the number of uncertainty words and the total number of words per meeting in a new DataFrame called 'temp_total'.
temp_total = df_min.groupby(['minutes'])['TotalWordCount','UncerScore'].sum().reset_index().rename(columns={'CombScore': 'combsum'})

#Loading the database of the date of the minutes of Banxico as the DataFrame 'date'.
date = pd.read_csv("mexico_minutes_date.csv", sep = ';', encoding = "utf-8")

#Merging the 'temp_total' DataFrame with the 'date' DataFrame in a new DataFrame named 'minutes_date'.
minutes_date = pd.merge(temp_total, date,  how='left', left_on=['minutes'], right_on = ['minutes'])

#Changing the format of the 'datedecision' column from object to datetime64[ns]. In particular, we take the date in which the meeting took place and not the release date of the minutes.
minutes_date['datedecision'] = pd.to_datetime(minutes_date['datedecision'], infer_datetime_format=True,dayfirst=True)

#Setting 'datedecision' column as index of the DataFrame 'minutes_date'.
```

```
minutes_date = minutes_date.set_index('datedecision')

#Converting format of columns "TotalWordCount" and
 ↪  "UncerScore" from object to int64 in order to apply
 ↪  resample.
minutes_date["TotalWordCount"] =
 ↪  pd.to_numeric(minutes_date["TotalWordCount"])
minutes_date["UncerScore"] =
 ↪  pd.to_numeric(minutes_date["UncerScore"])

#Checking the format of the DataFrame 'minutes_date'.
minutes_date.dtypes

#We create a new DataFrame 'month' that includes the months
 ↪  that do not have observations.
month = minutes_date.resample('MS').sum()

#We create the score variable as column 'score'.
month['score'] = month['UncerScore'] /
 ↪  month['TotalWordCount']

#Creating Skip-Gram uncertainty index as a column
 ↪  'unc_skip_norm' of the DataFrame 'month'.
month['unc_skip_norm'] = (100 * month['score']) /
 ↪  month["score"].mean()

#Replacing the values of the column 'unc_skip_norm' that
 ↪  have the value 0 with the value of the previous
 ↪  observation.
month['unc_skip_norm'] =
 ↪  month['unc_skip_norm'].fillna(method='ffill')

#Creating a new DataFrame 'month_min' only with the
 ↪  Skip-Gram uncertainty index.
month_min = month[['unc_skip_norm']].copy()

#Saving the new DataFrame 'month_min' in an excel file.
month_min.to_csv("mexico_unc_skipgram_k145_s200_w10
 ↪  _totalminutes.csv")
```

The Skip-Gram uncertainty index for the minutes is saved in the excel file 'mexico_unc_skipgram_k145_s200_w10_totalminutes.csv'. The python codes and the results of the section Skip-Gram uncertainty indexes are comprised in the supplementary material folder as follows:

1. 'Skip-Gram uncertainty - section zero.py' (Python code to construct the Skip-Gram uncertainty index for the 'international financial and economic' section);
2. 'mexico_unc_skipgram_k145_s200_w10_zero.csv' (Excel file that contains the Skip-Gram uncertainty index for the 'international financial and economic' section);
3. 'Skip-Gram uncertainty - section one.py' (Python code to construct the Skip-Gram uncertainty index for the 'Mexican financial, economic and inflation' section);
4. 'mexico_unc_skipgram_k145_s200_w10_one.csv' (Excel file that comprises the Skip-Gram uncertainty index of the 'Mexican financial, economic and inflation' section);
5. 'Skip-Gram uncertainty - section two.py' (Python code to construct the Skip-Gram uncertainty index of the 'analysis and rationale behind the voting of the governing boards' section);
6. 'mexico_unc_skipgram_k145_s200_w10_two.csv' (Excel file that comprises the Skip-Gram uncertainty index of the 'analysis and rationale behind the voting of the governing boards' section);
7. 'Skip-Gram uncertainty - section three.py' (Python code to construct the Skip-Gram uncertainty index of the 'monetary policy decision' section);
8. 'fmexico_unc_skipgram_k145_s200_w10_three.csv' (Excel file that comprises the Skip-Gram uncertainty index of the 'monetary policy decsion' section).

4.4.3 Construction of the mean uncertainty index and the graphs

This section comprises the python code to construct the mean uncertainty index and to normalize the EPU index for Brazil. The EPU index for Mexico of Baker, Bloom and Davis (2016) is extracted from their web page[3] in an excel file with the name 'Mexico_Policy_Uncertainty_Data.csv'.

To create the Figures A.1, A.2, A.3, A.4, A.5 of the paper, we merge all the uncertainty indices in one excel file and we save it with the name 'lda_skip_mean_epu_combined.csv'. The python code to create the figures and the database is attached in the supplementary material folder with the name 'Mexico graphs uncertainty index.py'.

[3] https://www.policyuncertainty.com/

4.5 Structural VAR Model

This section explains the new databases that are used in the Structural VAR estimation that are not described above. We then display the python code to merge the different databases such as the uncertainty index database and the FRED database. Finally, we show the stata code to estimate the Structural VAR model.

4.5.1 Databases

The following list comprises the variables used in the Structural VAR estimation that are not explained before.

1. **Interest rate target**. The target interest rate decided in the board of governors of the Bank of Mexico is extracted from the web page of the Banxico.[4] The target interest rate is comprised in the file 'tipo_interes.csv'.

2. **Global EPU index**. The global EPU index of Baker, Bloom and David (2016) describes the economic policy uncertainty in the world. It is extracted from their web page.[5] The csv file is included in the supplementary material folder with the name 'Global_Policy _Uncertainty_Data.csv'.

3. **Federal Reserve Bank of St. Louis or FRED database**. The following financial variables are extracted from the FRED database and included in the supplementary material folder like 'Mexico_fred.csv':

   - **Consumer price index**: Series ID: CPALCY01MXM661N; Title: consumer price index: total, all items for Mexico; Units: index 2015 = 100; Frequency = monthly; Seasonal adjustment = not seasonally adjusted; Excel tag = cpi_hundred;
   - **Exchange rate**: Series ID: EXMXUS; Title: Mexico / U.S. foreign exchange rate; Units: Mexican new pesos to one U.S. dollar; Frequency = monthly; Seasonal adjustment = not seasonally adjusted; Excel tag = exmxus;
   - **Interbank** rate for Mexico: Series ID: IRSTCI01MXM156N; Title: immediate rates: less than 24 hours: call money/interbank rate for Mexico; Units: percent; Frequency = monthly; Seasonal adjustment = not seasonally adjusted; Excel tag = int_twentyfourhours;
   - **Money supply M3**: Series ID: MABMM301MXM189S; Title: M3 for Mexico; Units: national currency; Frequency = monthly; Seasonal adjustment = seasonally adjusted; Excel tag = m_three_pesos.

[4] https://www.banxico.org.mx/

[5] https://www.policyuncertainty.com/

4.5.2 Merging databases

The different databases are merged for Structural VAR estimation with the name ‘mexico_svar.xlsx’. We then transform it to stata format with the name ‘mexico_dta.xlsx. The following lines show the python code to merge the different databases and is included in the supplementary material folder with the name ‘mexico_creating_database_svar.py’.

```python
import numpy as np
import pandas as pd
import csv

#Importing databases as DataFrames.
total = pd.read_csv("lda_skip_mean_epu_combined.csv", sep = ",", encoding="utf-8")
rate = pd.read_csv("tipo_interes.csv", sep = ";", encoding="utf-8")
epu_global = pd.read_csv("Global_Policy_Uncertainty_Data.csv", sep = ";", encoding="utf-8")
mexico_financial = pd.read_csv("Mexico_fred.csv", sep = ";", encoding="utf-8", decimal=",")

###########################################
#Format of the date of the 'total' DataFrame #
###########################################

#Setting time format.
total['datedecision'] = pd.to_datetime(total['datedecision'], infer_datetime_format=True,dayfirst=True)

#We create new columns in the DataFrame with the values of the year, the month and the day.
#However, the values of the columns 'month' and 'day' are changed the one for the other to correct the initial date.
total['year'] = total['datedecision'].dt.year
total['day'] = total['datedecision'].dt.month
total['month'] = total['datedecision'].dt.day

```

```python
#We change the 'datedecision' column with the correct values of  the columns 'day' and 'month'.
total['datedecision'] = pd.to_datetime(total[["year", "month", "day"]])

#We set the column 'datedecision' as index of the DataFrame.
total = total.set_index('datedecision')

##########################################################
#Format of the date of the 'mexico_financial' DataFrame #
##########################################################

#Setting time format.
mexico_financial['datedecision'] = pd.to_datetime(mexico_financial['datedecision'], infer_datetime_format=True,dayfirst=True)

#We create new columns in the DataFrame with the values of the year, the month and the day.
#However, the values of the columns 'month' and 'day' are changed the one for the other to correct the initial date.
mexico_financial['year'] = mexico_financial['datedecision'].dt.year
mexico_financial['day'] = mexico_financial['datedecision'].dt.day
mexico_financial['month'] = mexico_financial['datedecision'].dt.month

#We change the 'datedecision' column with the correct values of  the columns 'day' and 'month'.
mexico_financial['datedecision'] = pd.to_datetime(mexico_financial[["year", "month", "day"]])

#We set the column 'datedecision' as index of the DataFrame.
mexico_financial = mexico_financial.set_index('datedecision')
```

```python

##################################################
#Format of the date of the 'epu_global' DataFrame #
##################################################

#We create new columns in the DataFrame with the values of the year, the month and the day.
epu_global['day'] = 1
epu_global['year'] = epu_global['Year']
epu_global['month'] = epu_global['Month']

#Limiting the DataFrame 'epu_global' to our sample.
epu_global = epu_global[epu_global.year >= 2011]
epu_global = epu_global[epu_global.year <= 2018]

#We create the 'datedecision' column with the correct values of  the columns 'day', 'month'  and 'year'.
epu_global['datedecision'] = pd.to_datetime(epu_global[["year", "month", "day"]])

#We set the column 'datedecision' as index of the DataFrame.
epu_global = epu_global.set_index('datedecision')

#We normalize the Global uncertainty index for our sample in the column 'unc_epu_global_norm'.
epu_global['unc_epu_global_norm'] = (100 * epu_global['GEPU_current']) / epu_global["GEPU_current"].mean()

#Creating the DataFrame 'epu_global_min' only with the column 'unc_epu_global_norm'.
epu_global_min = epu_global[['unc_epu_global_norm']].copy()

############################################
#Format of the date of the 'rate' DataFrame #
############################################

#Setting time format.
```

```
rate['datedecision'] = pd.to_datetime(rate['fecha'],
 ↪  infer_datetime_format=True,dayfirst=True)

#We set the column 'datedecision' as index of the
 ↪  DataFrame.
rate = rate.set_index('datedecision')

#We create a new DataFrame 'rate' that includes the months
 ↪  that do not have observations.
rate = rate.resample('MS').sum()

#We replace values of  the 'unc_lda_norm' column  with zero
 ↪  values instead of nan.
rate['tipo_interes'] = rate['tipo_interes'].replace(0,
 ↪  np.nan)

#Replacing the values of the column 'unc_skip_norm_one'
 ↪  that have the value 0 with the value of the previous
 ↪  observation.
rate['tipo_interes'] =
 ↪  rate['tipo_interes'].fillna(method='ffill')
###########################
#Merging the different DataFrames in the DataFrame 'unc'.
unc1 = pd.merge(rate, total, left_index=True,
 ↪  right_index=True)
unc2 = pd.merge(epu_global, unc1, left_index=True,
 ↪  right_index=True)
unc = pd.merge(mexico_financial, unc2, left_index=True,
 ↪  right_index=True)

#Creating DataFrame 'unc_min' only with the columns of the
 ↪  DataFrame 'unc' of interest for the Structural Var.
unc_min = unc[['unc_epu_global_norm','tipo_interes',
 ↪  'cpi_hundred', 'exmxus',  'int_twentyfourhours',
 ↪  'm_three_pesos', 'unc_skip_norm_one',
 ↪  'unc_skip_norm_zero', 'unc_skip_norm_two',
 ↪  'unc_lda_norm_one', 'unc_lda_norm_zero',
 ↪  'unc_lda_norm_two', 'unc_lda_norm_total',
 ↪  'unc_epu_norm', 'unc_skip_norm', 'mean_unc']].copy()

```

```
#Saving the DataFrame 'unc_min' for Structural Var.
unc_min.to_csv('mexico_svar.csv')
unc_min.to_excel("mexico_svar.xlsx")
```

4.5.3 Structural VAR: estimation

We investigate how uncertainty in the minutes of the meetings of the Bank of Mexico board of governors affects the key financial variables for monetary policy such as the interbank rate. For this purpose, we compute a Structural VAR model with stata. The stata code to estimate SVAR is saved in the supplementary material folder as 'SVAR_mexico.do'. The following stata code corresponds to the construction of the impulse response functions of a rise in one standard shock in the mean uncertainty index.

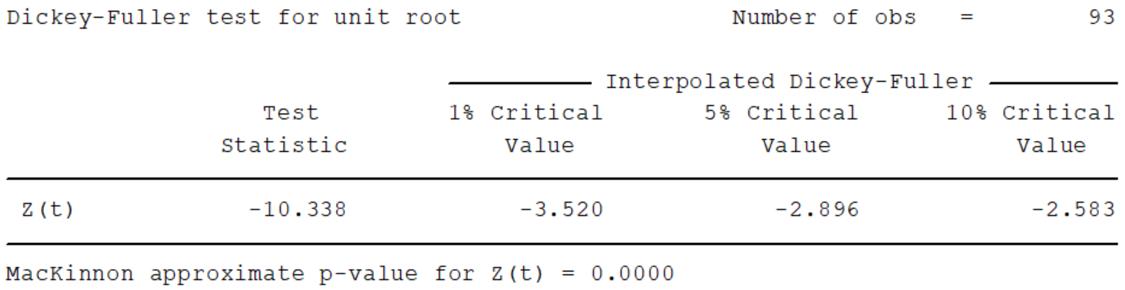
```
*Setting date index from January 2011.
gen date = m(2011m1) + _n - 1
format %tm date
tsset date

*Descriptive statistics between January 2011 and December
 ↪  2018.
summarize tipo_interes cpi_hundred exmxus
 ↪  int_twentyfourhours m_three_pesos unc_skip_norm_two
 ↪  unc_skip_norm_one unc_skip_norm_zero unc_lda_norm_one
 ↪  unc_lda_norm_zero unc_lda_norm_two unc_lda_norm_total
 ↪  unc_epu_norm unc_skip_norm mean_unc if date>=tm(2011m1)

*Creating log variables.
gen ln_mean_unc = log(mean_unc)
gen ln_tipo_interes = log(tipo_interes)
gen ln_m_three_pesos = log(m_three_pesos)
gen ln_exmxus = log(exmxus)
gen ln_cpi_hundred = log(cpi_hundred)
gen ln_int_twentyfourhours = log(int_twentyfourhours)

gen ln_unc_skip_norm_one = log(unc_skip_norm_one)
gen ln_unc_skip_norm_zero = log(unc_skip_norm_zero)
gen ln_unc_skip_norm_two = log(unc_skip_norm_two)

gen ln_unc_lda_norm_one = log(unc_lda_norm_one)
gen ln_unc_lda_norm_zero = log(unc_lda_norm_zero)
gen ln_unc_lda_norm_two = log(unc_lda_norm_two)

gen ln_unc_lda_norm_total = log(unc_lda_norm_total)
gen ln_unc_epu_norm = log(unc_epu_norm)
gen ln_unc_skip_norm = log(unc_skip_norm)
gen ln_unc_epu_global_norm = log(unc_epu_global_norm)

*Creating log difference variables.
gen dln_mean_unc   = ln_mean_unc  - L.ln_mean_unc
gen dln_rate  = ln_tipo_interes  - L.ln_tipo_interes
gen dln_m_three_pesos = ln_m_three_pesos -
 ↪  L.ln_m_three_pesos
gen dln_exmxus = ln_exmxus - L.ln_exmxus
gen dln_cpi_hundred = ln_cpi_hundred - L.ln_cpi_hundred
gen dln_int_twentyfourhours = ln_int_twentyfourhours -
 ↪  L.ln_int_twentyfourhours

gen dln_unc_skip_norm_one = ln_unc_skip_norm_one-
 ↪  L.ln_unc_skip_norm_one
gen dln_unc_skip_norm_zero = ln_unc_skip_norm_zero -
 ↪  L.ln_unc_skip_norm_zero
gen dln_unc_skip_norm_two = ln_unc_skip_norm_two -
 ↪  L.ln_unc_skip_norm_two

gen dln_unc_lda_norm_one = ln_unc_lda_norm_one-
 ↪  L.ln_unc_lda_norm_one
gen dln_unc_lda_norm_zero = ln_unc_lda_norm_zero -
 ↪  L.ln_unc_lda_norm_zero
gen dln_unc_lda_norm_two = ln_unc_lda_norm_two -
 ↪  L.ln_unc_lda_norm_two

gen dln_unc_lda_norm_total = ln_unc_lda_norm_total-
 ↪  L.ln_unc_lda_norm_total
gen dln_unc_epu_norm = ln_unc_epu_norm - L.ln_unc_epu_norm
gen dln_unc_skip_norm = ln_unc_skip_norm -
 ↪  L.ln_unc_skip_norm
gen dln_unc_epu_us_norm = ln_unc_epu_us_norm -
 ↪  L.ln_unc_epu_us_norm
gen dln_unc_epu_global_norm = ln_unc_epu_global_norm -
 ↪  L.ln_unc_epu_global_norm
```

```
*We drop observations before February 2011
drop if date <= tm(2011m2)

*We drop observations after December 2018
drop if date > tm(2018m12)

*We check if our variables pass the Dickey Fuller.
dfuller dln_mean_unc
dfuller dln_rate
dfuller dln_m_three_pesos
dfuller dln_exmxus
dfuller dln_cpi_hundred
dfuller dln_int_twentyfourhours

dfuller dln_unc_skip_norm_one
dfuller dln_unc_skip_norm_zero
dfuller dln_unc_skip_norm_two

dfuller dln_unc_lda_norm_one
dfuller dln_unc_lda_norm_zero
dfuller dln_unc_lda_norm_two

dfuller dln_unc_lda_norm_total
dfuller dln_unc_epu_norm
dfuller dln_unc_skip_norm

*Then, we define the Cholesky restrictions.
matrix A = (1,0,0,0,0\.,1,0,0,0\.,.,1,0,0\.,.,.,1,0\.,.,.,.,1)
matrix B = (.,0,0,0,0\0,.,0,0,0\0,0,.,0,0\0,0,0,.,0\0,0,0,0,.)


****************************************
*Estimation of SVAR with mean uncertainty index from February 2011 until December 2018 *
****************************************

```

```
*The varsoc test reports the final prediction error (FPE), Akaike's information criterion (AIC), Schwarz's Bayesian information criterion (SBIC), and the Hannan and Quinn information criterion (HQIC) lagorder selection statistics.
varsoc dln_mean_unc dln_int_twentyfourhours dln_m_three_pesos dln_exmxus dln_cpi_hundred  if date>=tm(2011m2), lutstats

*Estimation of the SVAR model for the mean uncertainty index from February 2011 until December 2018.
svar dln_mean_unc dln_int_twentyfourhours dln_m_three_pesos dln_exmxus dln_cpi_hundred  if date>=tm(2011m2), dfk aeq(A) beq(B) lags(1)
matrix Aest = e(A)
matrix Best = e(B)
matrix chol_est = inv(Aest)*Best
matrix list chol_est
matrix sig_var = e(Sigma)
matrix chol_var = cholesky(sig_var)
matrix list chol_var

*varnorm reports the Jarque-Bera statistic.
varnorm

*varlmar reports the Lagranger-Multiplier test for residual autocorrelation after SVAR.
varlmar, mlag(5)

*varstable indicates the eigenvalue stability conditions.
varstable

*Impulse response functions from the Structural VAR model corresponding to one standard-deviation in the mean uncertainty index in interbank interest rate from February 2011 until December 2018.
irf create order1, step(8) set(myirf1)
irf graph oirf, impulse(dln_mean_unc)
    ↪  response(dln_int_twentyfourhours) subtitle("")
    ↪  plot1opts(lcolor(red))  byopts(legend(off))
    ↪  byopts(graphregion(color(white)))
    ↪  byopts(bgcolor(white))    byopts(note("")) xtitle("")

*Impulse response functions from the Structural VAR model
    ↪  corresponding to one standard-deviation in the mean
    ↪  uncertainty index in money supply from February 2011
    ↪  until December 2018.
irf create order1, step(8) set(myirf2)
irf graph oirf, impulse(dln_mean_unc)
    ↪  response(dln_m_three_pesos) subtitle("")
    ↪  plot1opts(lcolor(red))  byopts(legend(off))
    ↪  byopts(graphregion(color(white)))
    ↪  byopts(bgcolor(white))    byopts(note("")) xtitle("")

*Impulse response functions from the Structural VAR model
    ↪  corresponding to one standard-deviation in the mean
    ↪  uncertainty index in exchange rate from February 2011
    ↪  until December 2018.
irf create order1, step(8) set(myirf3)
irf graph oirf, impulse(dln_mean_unc) response(dln_exmxus)
    ↪  subtitle("") plot1opts(lcolor(red))
    ↪  byopts(legend(off)) byopts(graphregion(color(white)))
    ↪  byopts(bgcolor(white))    byopts(note("")) xtitle("")

*Impulse response functions from the Structural VAR model
    ↪  corresponding to one standard-deviation in the mean
    ↪  uncertainty index in consumer price index from February
    ↪  2011 until December 2018.
irf create order1, step(8) set(myirf4)
irf graph oirf, impulse(dln_mean_unc)
    ↪  response(dln_cpi_hundred) subtitle("")
    ↪  plot1opts(lcolor(red))  byopts(legend(off))
    ↪  byopts(graphregion(color(white)))
    ↪  byopts(bgcolor(white))    byopts(note("")) xtitle("")
```

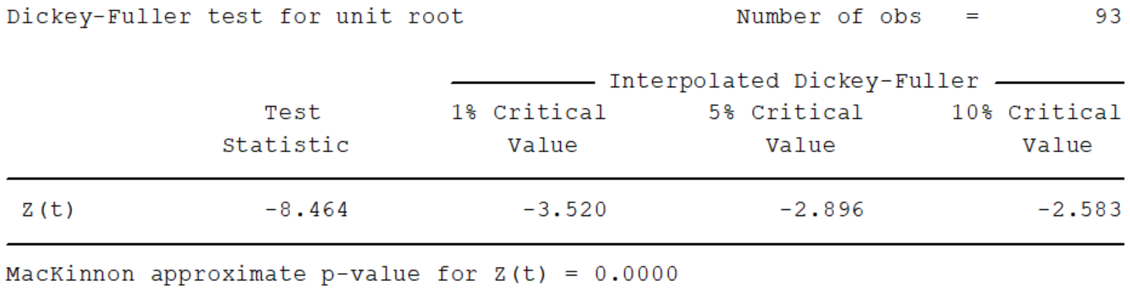
4.5.4 Structural VAR: measures of goodness of fit

We show the results of the measures of goodness of fit of the Structural VAR estimations that are not included in the paper. All the log variables are differentiated to overcome the problem of non-stationary since the augmented Dickey-Fuller test of the variables in levels indicates that they are I(1). Figures 1, 2, 3, 4 and 5 show the results of the Dickey Fuller test that check if the log difference variables are I(1). Our results show that all the log difference variables are stationary or I(1).



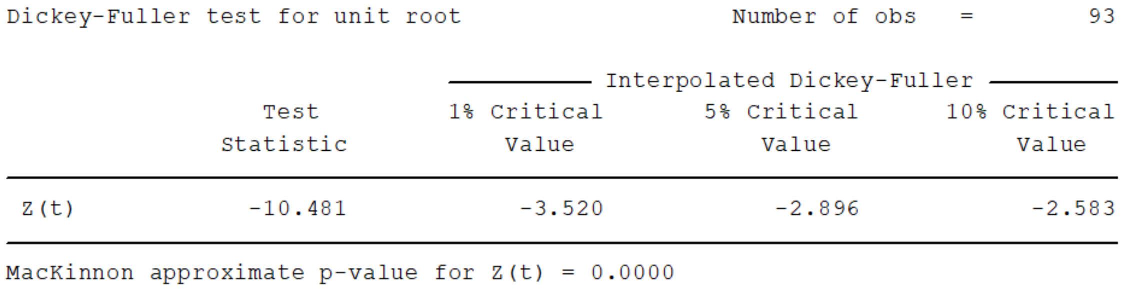
```
Dickey-Fuller test for unit root                   Number of obs   =        93

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)            -10.338            -3.520            -2.896            -2.583
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0000
```

Figure 1: Dickey-Fuller test for unit root for the log difference of the mean uncertainty index.



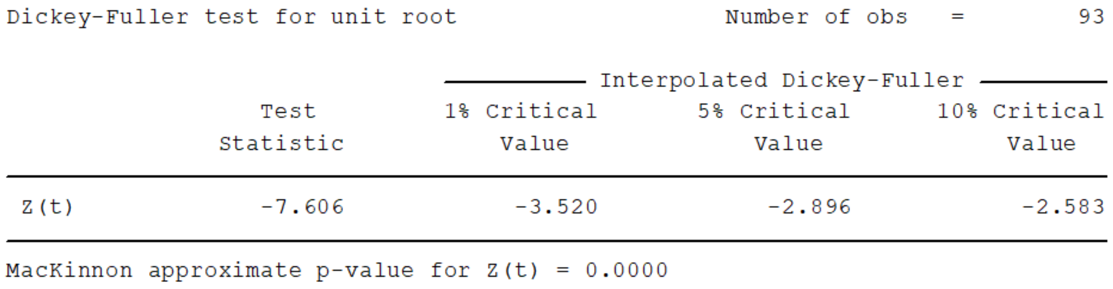
```
Dickey-Fuller test for unit root                   Number of obs   =        93

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)             -8.464            -3.520            -2.896            -2.583
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0000
```

Figure 2: Dickey-Fuller test for unit root to the log difference of the 24 hours inter-bank interest rate.



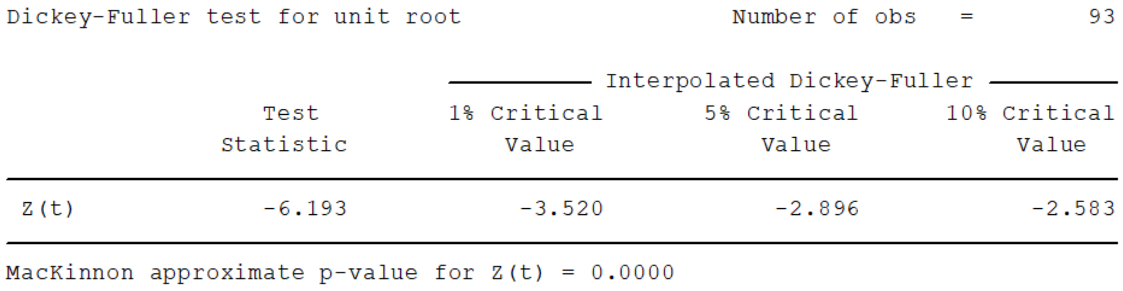
```
Dickey-Fuller test for unit root                   Number of obs   =        93

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)            -10.481            -3.520            -2.896            -2.583
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0000
```

Figure 3: Dickey-Fuller test for unit root to the log difference of the money supply.



```
Dickey-Fuller test for unit root                   Number of obs   =        93

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)             -7.606            -3.520            -2.896            -2.583
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0000
```

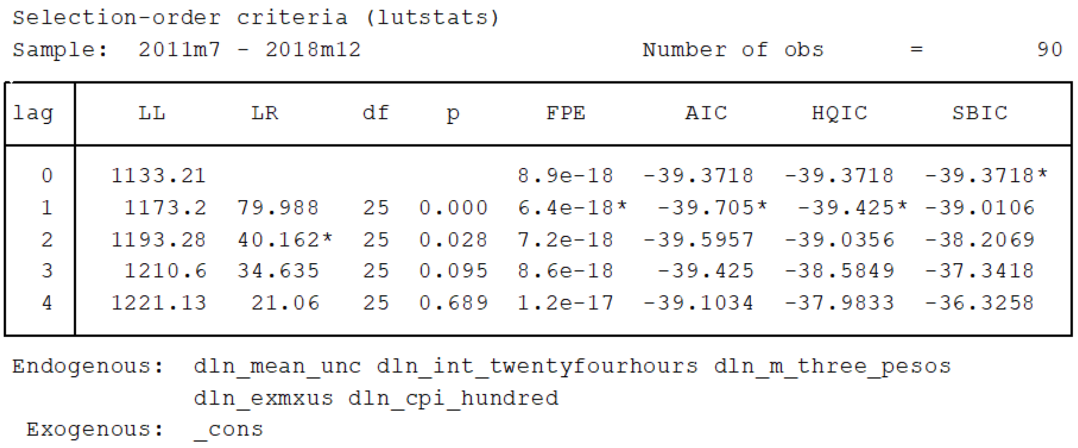
Figure 4: Dickey-Fuller test for unit root to the log difference of the exchange rate.



```
Dickey-Fuller test for unit root                   Number of obs   =        93

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)             -6.193            -3.520            -2.896            -2.583
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0000
```

Figure 5: Dickey-Fuller test for unit root to the log difference of the consumer price index.

Figure 6 shows the results of the varsoc test that reports the final prediction error (FPE), Akaike's information criterion (AIC), Schwarz's Bayesian information criterion (SBIC), and the Hannan and Quinn information criterion (HQIC) lag order selection statistics. The optimal number of lags is one according to the AIC, SBIC, HQIC and FPE.

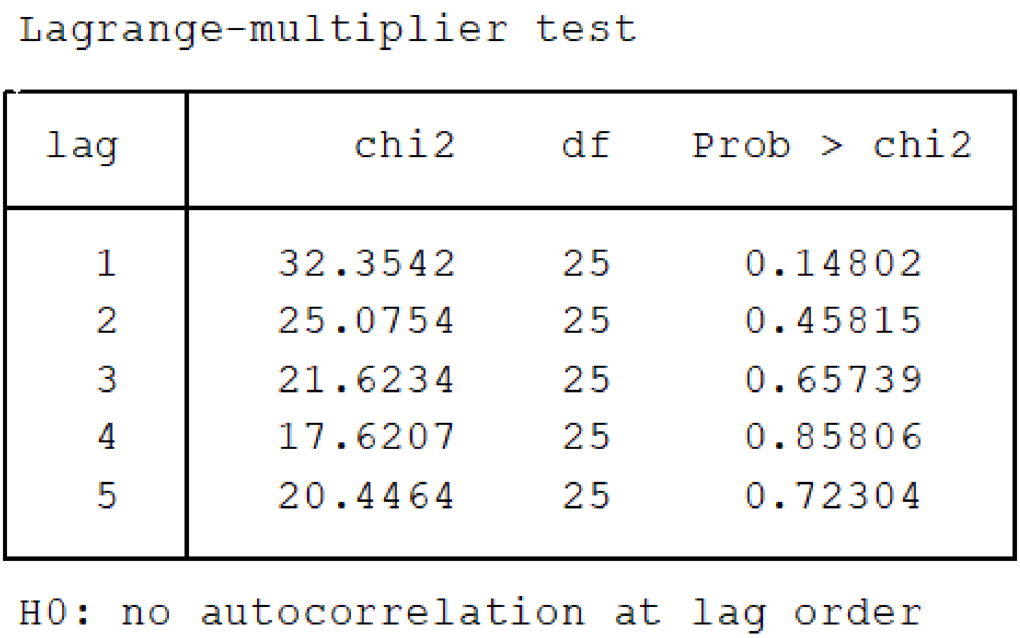


```
   Selection-order criteria (lutstats)
   Sample:  2011m7 - 2018m12                        Number of obs      =       90
  +---------------------------------------------------------------------------+
  |lag |    LL      LR      df    p      FPE       AIC      HQIC      SBIC    |
  |----+----------------------------------------------------------------------|
  |  0 |  1133.21                      8.9e-18  -39.3718  -39.3718  -39.3718* |
  |  1 |   1173.2  79.988   25  0.000  6.4e-18*  -39.705*  -39.425* -39.0106  |
  |  2 |  1193.28  40.162*  25  0.028  7.2e-18  -39.5957  -39.0356  -38.2069  |
  |  3 |   1210.6  34.635   25  0.095  8.6e-18   -39.425  -38.5849  -37.3418  |
  |  4 |  1221.13   21.06   25  0.689  1.2e-17  -39.1034  -37.9833  -36.3258  |
  +---------------------------------------------------------------------------+
   Endogenous:  dln_mean_unc dln_int_twentyfourhours dln_m_three_pesos
                dln_exmxus dln_cpi_hundred
    Exogenous:  _cons
```

Figure 6: Final prediction error (FPE), Akaike's information criterion (AIC), Schwarz's Bayesian information criterion (SBIC), and the Hannan and Quinn information criterion (HQIC) lagorder selection statistics.

Figures 7 and 8 show the outputs of the tests of the Structural VAR estimations of one standard-deviation in the mean uncertainty index. Figure 7 shows the output of the

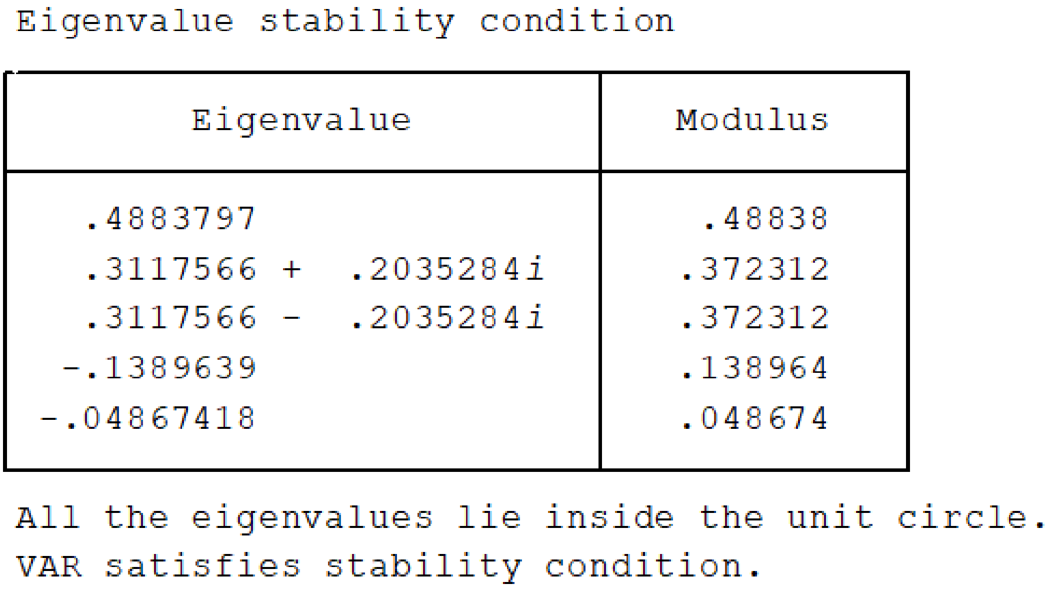
Lagrange multiplier test. Our results do not reject the null hypothesis which means there is no autocorrelation in the residuals for all of the five lags tested.



```
Lagrange-multiplier test

  lag        chi2     df    Prob > chi2

   1       32.3542    25      0.14802
   2       25.0754    25      0.45815
   3       21.6234    25      0.65739
   4       17.6207    25      0.85806
   5       20.4464    25      0.72304

H0: no autocorrelation at lag order
```

Figure 7: Lagrange multipier test.

Figure 8 shows that our SVAR model complies with the stability condition since all roots of the characteristic polynomial are outside of the unit circle.



```
Eigenvalue stability condition

          Eigenvalue            Modulus

   .4883797                      .48838
   .3117566 +  .2035284i        .372312
   .3117566 -  .2035284i        .372312
  -.1389639                     .138964
 -.04867418                     .048674

All the eigenvalues lie inside the unit circle.
VAR satisfies stability condition.
```

Figure 8: Eigen value stability condition.

Chapter 5

Collapsing Financial Markets: Unsupervised Modelling of the Coronavirus and Trade War News

5.1 Introduction

During 2019, US financial markets rose steadily despite the growing concern about a possible trade war between the US and China, and a non-deal Brexit. At the beginning of 2020, in particular on 19 February 2020, the S&P 500 index reached an historic peak. Then, the spread of COVID-19 in European countries and in Asia led to a memorable collapse of the financial markets, followed by a quick recovery due to the interventions of the Fed and of the US government's fiscal packages. In this paper, we investigate the relation between newspapers articles and financial indices, from the beginning of 2019 until mid 2020, using unsupervised machine learning techniques for text mining.

In the economic literature, text mining techniques are becoming increasingly popular to investigate the effect of the news on the real economy and on the markets. For example, Kalamara et al. (2020) make extensive use of text mining techniques for extracting information from three leading UK newspapers, to forecast macroeconomic variables with machine learning methods. Hansen and McMahon (2016) use unsupervised machine learning methods, in particular Latent Dirichlet Allocation (LDA), for constructing text measures of the information released by the Federal Open Market Committee (FOMC), to investigate the impact of FOMC communications on the markets and on some economic variables. Similarly, Hansen, McMahon, and Prat (2018) use LDA and dictionary methods to study the effect of transparency on the decisions of the FOMC.

Machine learning techniques are also used to build measures of uncertainty based on various text sources. For instance, Ardizzi et al. (2019) construct Economic Policy

Uncertainty (EPU) indices for Italy from newspaper and twitter data to study debit card expenditure. In particular, Soto (2021) uses unsupervised machine learning techniques to construct uncertainty measures from the text information released by commercial banks in their quarterly conference calls. He uses the Skip-gram model for Word Embedding and K-Means to find the word vectors nearest to the vector representations of the words 'uncertainty' and 'uncertain' and thereby constructs a list of uncertainty words, whose frequency in the documents is used to build an uncertainty index. Then, with the help of LDA, he constructs topic-specific uncertainty indices. On the other hand, an example of derivation of uncertainty measures from newspapers articles is given by Azqueta-Gavaldon et al. (2020). These authors use Word Embedding (with the Skip-gram model) and LDA to construct national uncertainty indices from Italian, Spanish, German and French newspapers. Then, they use a Structural VAR model to investigate the impact of the national uncertainty indices on some macroeconomic variables such as investment in machinery and equipment.

Other authors also investigate the use of sentiment indices based on various text sources concerning news on the financial markets. Just to mention, Zhu et al. (2019) utilize a monthly sentiment index named the Equity Market Volatility (EMV) and the daily VIX index to predict the evolution of US financial markets. In particular, they use a GARCH-MIDAS model to incorporate variables with different frequencies (daily and monthly) and conclude that the EMV index is more helpful than the VIX index in predicting volatility.

As far as the COVID-19 pandemic is concerned, Baker et al. (2020) construct three measures to capture different sources of uncertainty: stock market volatility, EPU and unsureness in business expectations. On the other hand, Haroon and Rizvi (2020) investigate how sentiment has driven financial markets during the first months of the coronavirus pandemic. These authors use an EGARCH model to study the effect of sentiment and panic in investors (using the Ravenpack Panic Index and the Global Sentiment Index) on the volatility of a wide range of financial indices relative to the world and US markets and to 23 sectors of the Dow Jones. In similar fashion, Albulescu (2020) investigates the effect on the VIX index of the US EPU index, the number of COVID-19 cases and the COVID-19 death rates. They find that the Chinese and world COVID-19 death rates are positively associated with the VIX index and that the US EPU index is positively associated with the volatility in the financial markets. Moreover, to deepen the analysis, a few authors also proceeded to create their own sentiment indices. Among these, Mamaysky (2020) builds several topic-specific sentiment indices solely for coronavirus news. In particular, he selects news mentioning the words 'coronavirus' and 'COVID-19' from the beginning of 2019 to the end of April 2020, and then applies LDA to classify coronavirus news under nine headings. Thus, constructing a daily positive-negative sentiment index with the Loughran-McDonald dictionary (Loughran and McDonald, 2011), he creates topic-specific sentiment indices and finds that they are correlated with the evolution of the stock markets.

In this paper, we create text measures to quantify the content and sentiment of US news, related in particular to the COVID-19 pandemic, using unsupervised machine learning algorithms such as LDA, Word Embedding (with the Skip-gram model) and K-Means. In particular, we construct text measures from the headlines and snippets of articles in the English version of the New York Times from 2 January 2019 to 1 May 2020. To infer the content or theme of the news in the documents, that is, in the newspaper articles, we run LDA with sixty topics. Then, we determine the daily probability distribution of each topic and use it as a daily measure of attention to each topic in the daily news. To create sentiment measures, we resort to Word Embedding (using the Skip-gram model) and K-Means. With these, we come out with a list of words having a meaning similar to the word 'uncertainty'. Actually, we consider in this list all the words that are in the same clusters of the words 'uncertain', 'uncertainty', 'fears', 'fears' and 'worries', since they share a similar semantic meaning. This list is then used as an uncertainty dictionary to construct a daily uncertainty index by counting the frequency of its words present in all the articles of a given day. To create topic-specific uncertainty indices, we then combine the daily LDA probabilities of each topic with the uncertainty index obtained with Word Embedding and K-Means. In this way, we come out with uncertainty indices for specific topics such as, in particular, 'coronavirus', 'trade war', 'climate change', 'economic-Fed' and 'Brexit'.

To complete the analysis, we investigate, using an EGARCH model, the relationship between these topic-specific uncertainty indices and the returns of several US financial indices such as the S&P 500, the Nasdaq and the Dow Jones, as well as the 10 year US treasury bond yields. We find that in the period under scrutiny, the 'trade war' and 'coronavirus' uncertainty indices have a significant negative effect on the mean returns of the S&P 500. The 'trade war' uncertainty index explains most of the behavior of the S&P 500 during 2019, whereas the 'coronavirus' uncertainty index explains most of the behavior of the S&P 500 in the first months of 2020. Moreover, an increase in the 'trade war' and 'coronavirus' uncertainty indices significantly increases the volatility of the S&P 500 returns and the mean returns of the VIX index. We also find that a rise in the 'economic-Fed' uncertainty index significantly increases the mean returns of the S&P 500 index. This would mean that news about interventions of the Fed or the US government have a positive effect on the S&P 500 in days of uncertainty.

The paper is organized as follows. In Section 2 we introduce our text data and explain the construction of the topic-specific uncertainty indices with the help of LDA, Word

Embedding and K-Means. In Section 3 we illustrate the EGARCH analysis and comment on the results. Finally, in Section 4 we give some conclusions.

5.2 Topic and Sentiment Analysis

5.2.1 Text data

Our raw data are the headlines and the snippets of the English version of the articles of the New York Times from 2 January 2019 to 1 May 2020. We downloaded the headlines and the snippets of the articles using the New York Times API and then eliminated several sections that were not pertinent for the analysis, that is, not containing relevant information that might affect the financial markets (see Table 1). Articles published after 4:00 pm, when the stock exchanges are close, were assigned to the next day. Also, articles published over the weekend or on days in which the New York Stock Exchange was closed were assigned to the next working day (usually the next Monday).

Table 1: List of sections of the New York Times not considered in the analysis.

arts and leisure, at home, book, briefing, corrections, crosswords and games, culture, dining, express, fashion, fashion and style, food, games, gender, graphics, health, insider, learning, letters, live, magazine, metropolitan, movies, multimedia / photos, New York, none, obit, obituaries, parenting, photo, reader center, smarter living, real state, society, special section, sports, style, styles, Sunday review, t magazine, t magazine / art, t magazine / fashion and beauty, tstyle, the learning network, the weekly, theater, times insider, travel, weekend and well.

5.2.2 Topic analysis: Latent Dirichlet Allocation

To extract the topics (the subjects, the themes) of the articles, we use Latent Dirichlet Allocation (LDA), an unsupervised machine learning technique introduced by Blei, Ng and Jordan (2003) for text mining. The power of LDA resides in its ability to automatically identify the topics in the articles without the need of human intervention, that is, without the need to read them by an experienced reader. LDA assumes that each document, which is a newspaper article in our case (or, more precisely, the headline and the snippet of the article), is made up of various words, and that the set of all documents form what we call the corpus. In this setting, topics are latent (non observable) probability distributions over words, and words with the highest weights are normally used to assign meaningful names to the topics. Of course, this somehow subjective labelling of the topics does not affect in any way the analysis and is used to help in the interpretation of the results. LDA supplies the most probable topics related to each article.

Before applying LDA, our raw text data needs to be 'cleaned'. First of all, the preprocessing involves converting all words in the corpus in lowercase and removing any punctuation mark. Next, it requires the removal of all 'stop' words such as 'a', 'you', 'themselves', etc., which are repeated in the documents without providing relevant information on the topics. The remaining words are then stemmed to their base root. For instance, the words 'inflationary', 'inflation', 'consolidate' and 'consolidating' are converted into their stems, which are 'inflat' and 'consolid', respectively. Thus, the stems are ordered according to the *term frequency-inverse document frequency* (tf-idf) index. This index grows with the number of times a stem appears in a document, and decreases as the number of documents containing that stem increases. It serves to eliminate common and unusual words. All stems with a value of 12,000 or lower have been disregarded. Overall, we came out with a corpus containing a total number of 502,173 stems and 10,314 unique stems.

After preprocessing the data, we carried out the LDA analysis (Hansen, McMahon, and Prat, 2018) on the 'cleaned' corpus, fixing at 60 the total number of topics, and setting the hyperparameters of the Dirichlet priors following the suggestions of Griffiths and Steyvers (2004). To obtain a sample from the posterior distribution, we then considered two runs of the Markov chain Monte Carlo Gibbs sampler, each one providing 1,000 draws, using a burn-in period of 1,000 iterations and a thinning interval of 50.

Tables 2 and 3 show for each of the 60 topics the first six words with the highest (posterior) probability. That is, for each topic, word 1 is the word (stem) with the highest probability in that topic, word 2 is the word (stem) with the second highest probability in that topic, and so on. On the basis of the probability distribution of words in a topic, we are able to somehow interpret it and then to assign it a tag. For instance, we assigned the tag 'coronavirus' to topic 29 since, for this topic, the words (stems) with the highest probability are 'coronaviru', which has a probability of 0.217, 'test', which has a probability of 0.067, 'pandem', which has a probability of 0.063, and 'viru', which has a probability of 0.051. In this way, we see that topics related to the economy and to the financial markets are those numbered 3, 10, 36, 44 and 51. Topics related to politics are those numbered 12, 13, 15, 24, 28, 30 and 35. Whereas topics related to the international economy and the political conditions include those numbered 8, 14, 23, 33, 44, 48 and 53. We should remark that we carried out the LDA analysis fixing at 60 the number of topics since with this number we were able to clearly distinguishes between the 'coronavirus' and 'trade war' topics. A larger number of topics supplies several topics related with the coronavirus pandemic (and not just one), whereas a lower number of topics, such as 40, for instance, does not clearly distinguish the ‘trade war’ topic from the others.

In addition to the above probability distributions of words characterizing each topic, the LDA analysis also provides the topic distribution for each document in the corpus, that is, it supplies the most probable topics associated with each article of the New York Times. These distributions will be used to obtain the daily distributions of topics over the period under scrutiny. In particular, we will consider the daily probability of each topic, $P_{i,t}$, where subscript i refers to the topic and subscript t to the day. This text measure will be used in Section 2.4 to construct our topic-specific uncertainty indices.

5.2.3 Sentiment analysis: Word Embedding and K-Means

In our situation, an article may convey a *certain* or *uncertain* feeling about a topic. This feeling, or sentiment, or tone, of an article will be deduced by using Word Embedding (with the Skip-gram model) and K-Means. These algorithms will provide a list of words, having a meaning similar to that of the word ‘uncertainty’, which will operate as an *uncertainty dictionary*. This, in turn, will be employed to measure the uncertainty present in each article and so to build a daily uncertainty index.

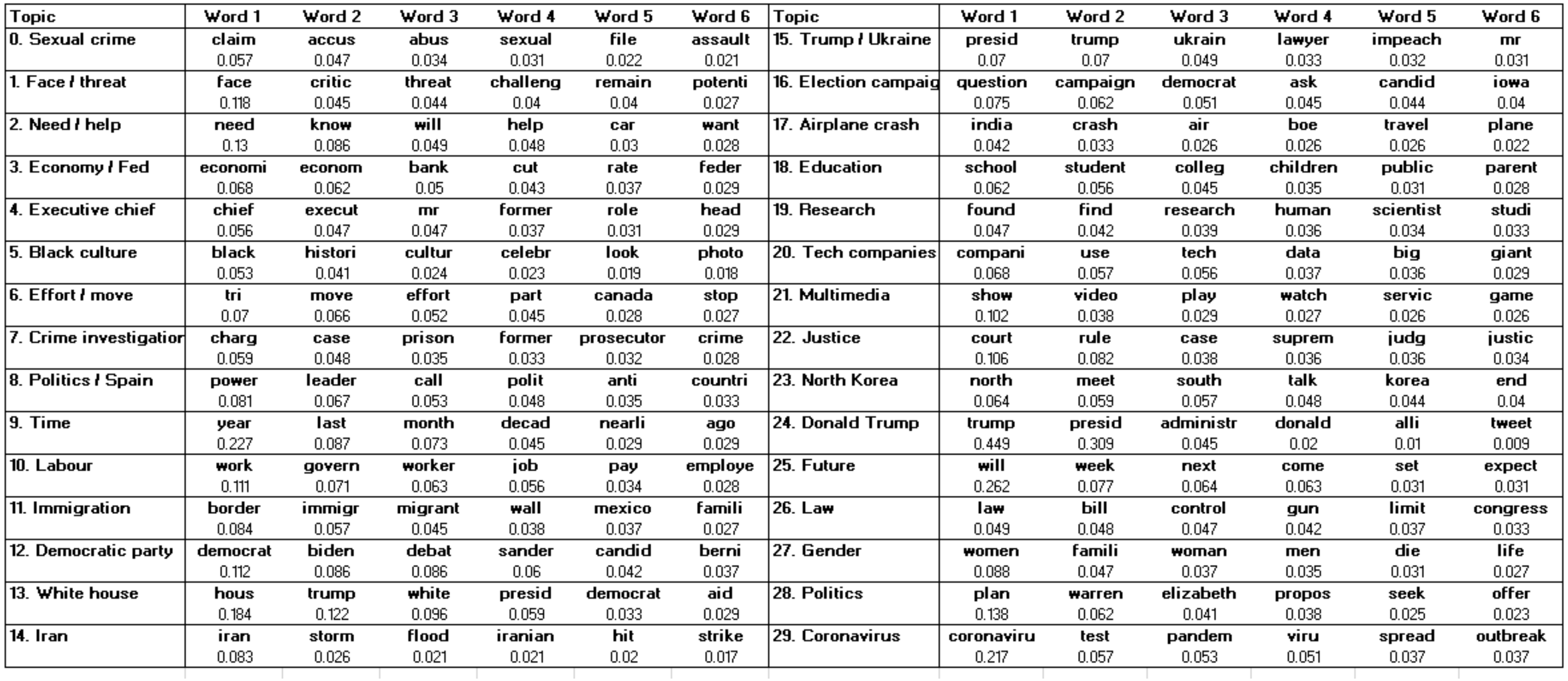
Word Embedding, introduced by Mikolov et al. (2013), is a continuous vector representation of words in a suitable low-dimensional Euclidean space, which aims to capture syntactic and semantic similarities between words, associating words with a similar meaning with vectors that are closer to each other, that is, that are in the same region of the space. Usually, this can be implemented adopting either the Common Bag Of Words (CBOW) model or the Skip-gram model. The main idea of these models is the possibility to extract a considerable amount of the meaning of a word from its *context* words, that is, from the words surrounding it. For instance, consider the following two sentences:

the economy experienced a period of increasing *uncertainty* about the growth capacity;

the economy experienced a period of increasing *fears* about the growth capacity.

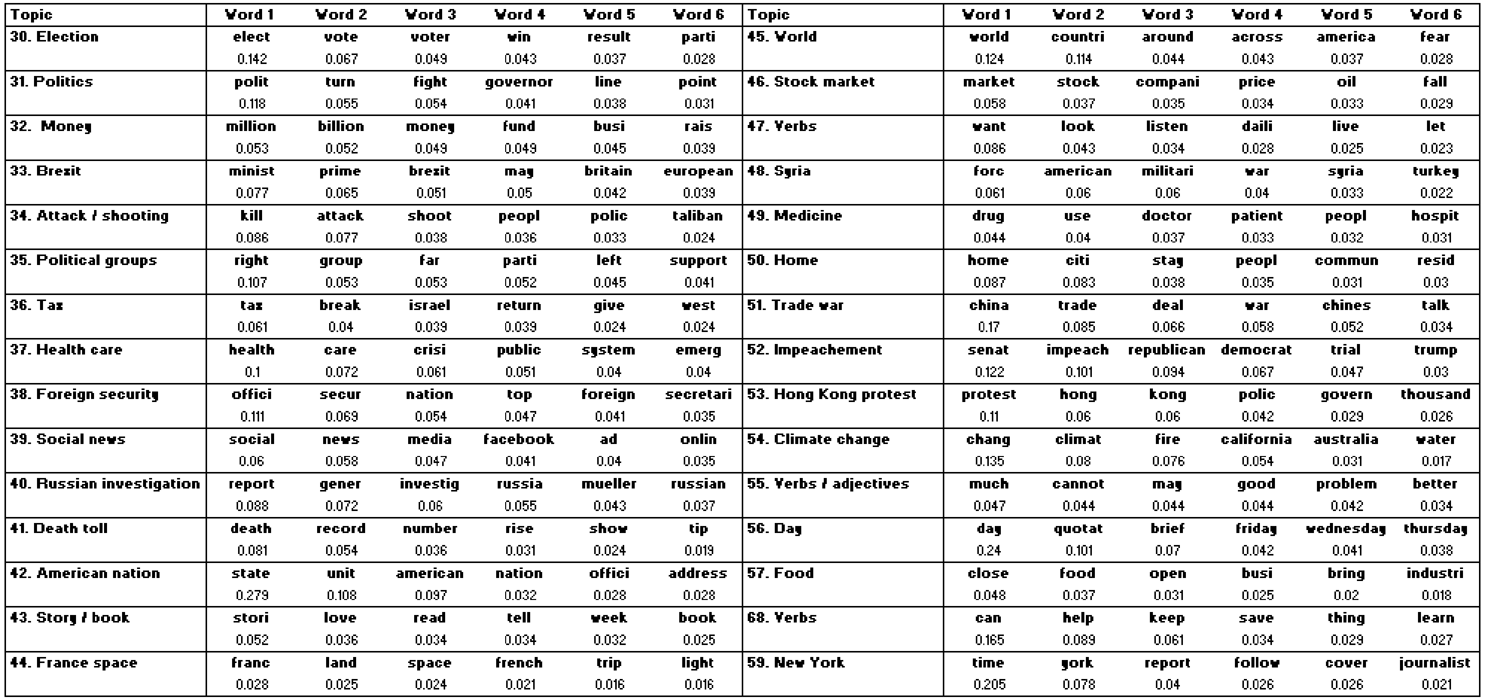
Here, the words ‘uncertainty’ and ‘fears’ have a similar meaning, which is related to doubt and worry. Both words are preceded by ‘the economy experienced a period of increasing’ and are followed by ‘about the growth capacity’. For our purposes, to carry out the Word Embedding we adopt the Skip-gram model as introduced by Mikolov et al. (2013). The basic idea of this model is that to create a dense vector representation of each word that is good at predicting the words that appear in its context. This involves the use of a neural network designed to predict context words on the basis of a given *center* word.

Table 2: Topic descriptions for the LDA analysis. The table shows the first six words with the highest (posterior) probability for each of the first thirty topics.



Topic	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Topic	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
0. Sexual crime	claim 0.057	accus 0.047	abus 0.034	sexual 0.031	file 0.022	assault 0.021	15. Trump / Ukraine	presid 0.07	trump 0.07	ukrain 0.049	lawyer 0.033	impeach 0.032	mr 0.031
1. Face / threat	face 0.118	critic 0.045	threat 0.044	challeng 0.04	remain 0.04	potenti 0.027	16. Election campaig	question 0.075	campaign 0.062	democrat 0.051	ask 0.045	candid 0.044	iowa 0.04
2. Need / help	need 0.13	know 0.086	will 0.049	help 0.048	car 0.03	want 0.028	17. Airplane crash	india 0.042	crash 0.033	air 0.026	boe 0.026	travel 0.026	plane 0.022
3. Economy / Fed	economi 0.068	econom 0.062	bank 0.05	cut 0.043	rate 0.037	feder 0.029	18. Education	school 0.062	student 0.056	colleg 0.045	children 0.035	public 0.031	parent 0.028
4. Executive chief	chief 0.056	execut 0.047	mr 0.047	former 0.037	role 0.031	head 0.029	19. Research	found 0.047	find 0.042	research 0.039	human 0.036	scientist 0.034	studi 0.033
5. Black culture	black 0.053	histori 0.041	cultur 0.024	celebr 0.023	look 0.019	photo 0.018	20. Tech companies	compani 0.068	use 0.057	tech 0.056	data 0.037	big 0.036	giant 0.029
6. Effort / move	tri 0.07	move 0.066	effort 0.052	part 0.045	canada 0.028	stop 0.027	21. Multimedia	show 0.102	video 0.038	play 0.029	watch 0.027	servic 0.026	game 0.026
7. Crime investigatior	charg 0.059	case 0.048	prison 0.035	former 0.033	prosecutor 0.032	crime 0.028	22. Justice	court 0.106	rule 0.082	case 0.038	suprem 0.036	judg 0.036	justic 0.034
8. Politics / Spain	power 0.081	leader 0.067	call 0.053	polit 0.048	anti 0.035	countri 0.033	23. North Korea	north 0.064	meet 0.059	south 0.057	talk 0.048	korea 0.044	end 0.04
9. Time	year 0.227	last 0.087	month 0.073	decad 0.045	nearli 0.029	ago 0.029	24. Donald Trump	trump 0.449	presid 0.309	administr 0.045	donald 0.02	alli 0.01	tweet 0.009
10. Labour	work 0.111	govern 0.071	worker 0.063	job 0.056	pay 0.034	employe 0.028	25. Future	will 0.262	week 0.077	next 0.064	come 0.063	set 0.031	expect 0.031
11. Immigration	border 0.084	immigr 0.057	migrant 0.045	wall 0.038	mexico 0.037	famili 0.027	26. Law	law 0.049	bill 0.048	control 0.047	gun 0.042	limit 0.037	congress 0.033
12. Democratic party	democrat 0.112	biden 0.086	debat 0.086	sander 0.06	candid 0.042	berni 0.037	27. Gender	women 0.088	famili 0.047	woman 0.037	men 0.035	die 0.031	life 0.027
13. White house	hous 0.184	trump 0.122	white 0.096	presid 0.059	democrat 0.033	aid 0.029	28. Politics	plan 0.138	warren 0.062	elizabeth 0.041	propos 0.038	seek 0.025	offer 0.023
14. Iran	iran 0.083	storm 0.026	flood 0.021	iranian 0.021	hit 0.02	strike 0.017	29. Coronavirus	coronaviru 0.217	test 0.057	pandem 0.053	viru 0.051	spread 0.037	outbreak 0.037

Table 3: Topic descriptions for the LDA analysis. The table shows the first six words with the highest (posterior) probability for each of the last thirty topics.



Topic	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Topic	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
30. Election	elect	vote	voter	win	result	parti	45. World	world	countri	around	across	america	fear
	0.142	0.067	0.049	0.043	0.037	0.028		0.124	0.114	0.044	0.043	0.037	0.028
31. Politics	polit	turn	fight	governor	line	point	46. Stock market	market	stock	compani	price	oil	fall
	0.118	0.055	0.054	0.041	0.038	0.031		0.058	0.037	0.035	0.034	0.033	0.029
32. Money	million	billion	money	fund	busi	rais	47. Verbs	want	look	listen	daili	live	let
	0.053	0.052	0.049	0.049	0.045	0.039		0.086	0.043	0.034	0.028	0.025	0.023
33. Brexit	minist	prime	brexit	may	britain	european	48. Syria	forc	american	militari	war	syria	turkey
	0.077	0.065	0.051	0.05	0.042	0.039		0.061	0.06	0.06	0.04	0.033	0.022
34. Attack / shooting	kill	attack	shoot	peopl	polic	taliban	49. Medicine	drug	use	doctor	patient	peopl	hospit
	0.086	0.077	0.038	0.036	0.033	0.024		0.044	0.04	0.037	0.033	0.032	0.031
35. Political groups	right	group	far	parti	left	support	50. Home	home	citi	stay	peopl	commun	resid
	0.107	0.053	0.053	0.052	0.045	0.041		0.087	0.083	0.038	0.035	0.031	0.03
36. Tax	tax	break	israel	return	give	west	51. Trade war	china	trade	deal	war	chines	talk
	0.061	0.04	0.039	0.039	0.024	0.024		0.17	0.085	0.066	0.058	0.052	0.034
37. Health care	health	care	crisi	public	system	emerg	52. Impeachement	senat	impeach	republican	democrat	trial	trump
	0.1	0.072	0.061	0.051	0.04	0.04		0.122	0.101	0.094	0.067	0.047	0.03
38. Foreign security	offici	secur	nation	top	foreign	secretari	53. Hong Kong protest	protest	hong	kong	polic	govern	thousand
	0.111	0.069	0.054	0.047	0.041	0.035		0.11	0.06	0.06	0.042	0.029	0.026
39. Social news	social	news	media	facebook	ad	onlin	54. Climate change	chang	climat	fire	california	australia	water
	0.06	0.058	0.047	0.041	0.04	0.035		0.135	0.08	0.076	0.054	0.031	0.017
40. Russian investigation	report	gener	investig	russia	mueller	russian	55. Verbs / adjectives	much	cannot	may	good	problem	better
	0.088	0.072	0.06	0.055	0.043	0.037		0.047	0.044	0.044	0.044	0.042	0.034
41. Death toll	death	record	number	rise	show	tip	56. Day	day	quotat	brief	friday	wednesday	thursday
	0.081	0.054	0.036	0.031	0.024	0.019		0.24	0.101	0.07	0.042	0.041	0.038
42. American nation	state	unit	american	nation	offici	address	57. Food	close	food	open	busi	bring	industri
	0.279	0.108	0.097	0.032	0.028	0.028		0.048	0.037	0.031	0.025	0.02	0.018
43. Story / book	stori	love	read	tell	week	book	68. Verbs	can	help	keep	save	thing	learn
	0.052	0.036	0.034	0.034	0.032	0.025		0.165	0.089	0.061	0.034	0.029	0.027
44. France space	franc	land	space	french	trip	light	59. New York	time	york	report	follow	cover	journalist
	0.028	0.025	0.024	0.021	0.016	0.016		0.205	0.078	0.04	0.026	0.026	0.021

Before proceeding with the Word Embedding, using the Skip-gram model, for the words in the articles of the relevant sections of the New York Times, we first need to preprocess the raw text data, though in a different manner than we did for LDA. Now, words are not stemmed since we could lose semantic differences between some of them. Instead, we now single out bigrams, that is, pairs of consecutive words such as, for instance, 'south_korean' or 'defense_secretary', that jointly bear a particular meaning or idea. Bigrams, that is, the two words forming it, are considered as a single token, that is, as if they were a single word. In the analysis, we considered all bigrams appearing with a frequency higher than 50. We fixed this threshold since it allows to capture many relevant bigrams, although excluding those with relatively low frequency. Moreover, we discarded from the analysis all articles that do not normally have an effect on financial markets, such as, for instance, articles on local crime or on New York local news, which might bias the results. Specifically, we eliminated all the articles whose main topic, that is, whose highest LDA topic probability is relative to one of the following topics: 0, 5, 6, 7, 8, 9, 11, 18, 21, 22, 27, 28, 34, 35, 37, 43, 44, 48, 57 and 59. After this cleaning, we remained with a corpus of 342,038 tokens (which are either bigrams or single words). On the cleaned set of articles, we considered Word Embedding, using the Skip-gram model, with a hidden layer of $H = 200$ elements and a context window of size 10 on each side of the center word (we also tried a hidden layer of 100 and 150 elements, and a context window of size 5 and 8). We implemented it using Word2Vec of the Gensim Python library. This embedding has been carried out for all unique terms (words) and all identified bigrams in the selected set of articles, to obtain, for each token (word or bigram), a dense vector of dimension H.

Then, to identify tokens with a similar meaning, we performed a K-Means clustering on the dense vectors thus obtained. K-Means is an unsupervised machine learning technique that clusters similar objects, which are in some sense close to each other, in a set of disjoint clusters (Chakraborty and Joseph, 2017). After some investigations in which we tried different combinations of the number of elements of the hidden layer, the context window size and the number of clusters, we fixed the number of clusters at 120. The chosen combination and in particular the chosen number of clusters is the one that provides, with respect to the purposes of our investigation, the most meaningful results in terms of semantic similarities.

Having obtained clusters of vectors related to tokens (words or bigrams) with similar meaning, we went on (as in Soto, 2021) to identify those clusters containing words related to *uncertainty*. Precisely, we considered the clusters containing the words 'fear', 'fears', 'worries', 'uncertain' and 'uncertainty'. Tables 4, 5, 6, 7 and 8 show the words that appear in these clusters. We can note that the cluster containing the word 'uncertainty' mainly includes words related to the trade war between China and the US, whereas the cluster containing the word 'worries' mainly includes words related to stock markets. It should also be noted that, a number of clusters smaller than 120 leads to clusters containing more than one of these five uncertainty words, but also containing many words that are not of interest.

All the words in these five clusters where merged together to build a list of words to be used as a dictionary of words related to the sentiment of uncertainty. For our purposes, this uncertainty dictionary seems to be better than other pre-established uncertainty dictionaries, such as that of Loughran and McDonald (2011), since it is tailored to our particular text data.

A better uncertainty dictionary could reasonably be obtained by considering a larger set of articles, maybe considering more than one newspaper.

Table 4: List of words in the cluster containing the word ‘fear’.

anxious, anywhere, battling, belt, born, brutal, civilians, communist, contagion, crisis, deep, fake_news, fear, feels, fighting, fingers, girl, greatest, indians, isis, isolation, italy, landslide, latin_america, lockdown, locked, looks_like, memories, neighbors, nightmare, outrage, poland, react, relative, revolution, shame, siege, solidarity, suffers, test, thailand, tour, tradition, trauma, turns, upheaval, war_ii, west, widening.

Table 5: List of words in the cluster containing the word ‘fears’.

analysts, bond_yields, central_banks, climb, damage, drop, exports, factories, fears, fell, financial_markets, fueled, gas, grew, growing, higher, highest, increase, increasing, oil, oil_prices, plunge, policymakers, prices, producers, rate, rattled, rise, rising, slide, slowdown, slowing, slows, slump, spike, supply, tourism, tumbled, worsening.

Table 6: List of words in the cluster containing the word ‘worries’.

central_bank, cut_interest, cut_rates, economic, economy, fed, federal_reserve, global, growth, interest_rates, investors, markets, rates, recession, stocks, worries.

With our uncertainty dictionary, we are now in a position to set up a daily uncertainty index for the US economy, which can be used to investigate the effect of uncertainty about the US economy on the financial markets. To construct this index, we first count the number of words of the uncertainty dictionary that are present in each article. The

Table 7: List of words in the cluster containing the word ‘uncertain’.

accord, agreed, alternative, approaching, backs, backstop, bloc, blow, boris, brinkmanship, brussels, closer, collision_course, complicate, compromise, corbyn, customs, deadline, deepening, europeans, extending, failed, failure, fate, forge, gives, grant, guarantee, heads, jan, john_bercow, last_ditch, likely, limbo, looming, macedonia, maneuver, mideast, nears, negotiating, obstacles, oct, paris_climate, persuade, pound, promises, prospect, quick, rather, rebels, remain, reverse, shinzo_abe, stalemate, stamp, step, suspend_parliament, suspension, throws, tries, two_sides, uncertain, unpredictable, vacuum, vowed, wall, yearlong.

Table 8: List of words in the cluster containing the word ‘uncertainty’.

chinese_goods, goods, mexico, negotiations, negotiators, progress, tariff, tariffs, trade, trade_deal, trade_talks, trade_war, uncertainty.

daily sum of uncertainty words, over all articles of a particular day t, is indicated by U_t. A daily uncertainty score S_t can then be obtained by dividing U_t by the total number N_t of words present in the articles that day:

$$S_t = U_t/N_t. \tag{1}$$

Our daily *US uncertainty index* is then given by

$$D_t = 100 \cdot \frac{S_t}{\frac{1}{M}\sum_{m=1}^{M} S_m}, \tag{2}$$

where M is the number of days of the period under study. Figure 1 shows the evolution of our US uncertainty index compared with the S&P 500 closing price index. The three peaks over a value of 125 of the moving average (with a 9-day rolling window) of the US uncertainty index correspond to important drops in the S&P 500 index.

5.2.4 Topic-specific uncertainty measures

Following Mamaysky (2020), we build topic-specific sentiment measures by multiplying the daily topic probabilities by the daily sentiment index. In our case, the sentiment index is given by the daily US uncertainty index obtained through Word Embedding and K-Means clustering. Thus, to measure the sentiment, or better the uncertainty, related to specific topics, we consider the following *topic-specific uncertainty indices*,

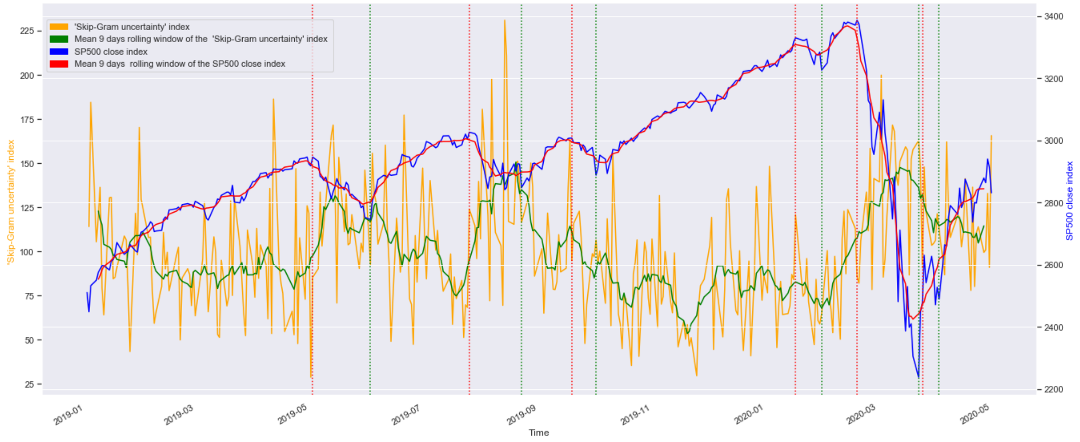


Figure 1: The yellow line shows the US uncertainty index obtained with the Skip-gram model. The green line represents the moving average of this index using a 9-day rolling window. The blue line shows the S&P 500 closing price index; the red line is the moving average with a 9-day rolling window. The vertical dotted red lines indicate some of the local maxima of the S&P 500 closing price index, whereas the vertical dotted green lines indicates some of the local minimums of the S&P 500 closing price index.

$$T_{i,t} = P_{i,t} \cdot D_t, \tag{3}$$

where subscript i indicates a specific topic and subscript t refers to a specific day.

Figure 2 shows the evolution of two topic-specific uncertainty indices, specifically of the ‘coronavirus’ and ‘trade war’ uncertainty indices. Similarly, Figures 3, 4 and 5 show the evolution of the ‘Brexit’, ‘economic-Fed’ and ‘climate change’ uncertainty index, respectively. From these behaviours it is immediate to notice that the peaks of the ‘trade war’ uncertainty index during 2019 correspond to drops in the S&P 500 closing price index, whereas the huge increase of the ‘coronavirus’ uncertainty index in the first months of 2020 corresponds to an historic drop in the S&P 500 index.

5.3 Uncertainty in news and financial markets volatility

To quantify how much of the behaviour of some US financial indices such as the S&P 500 index, the Dow Jones index, the Nasdaq Composite index, the VIX index and the US 10-year Treasury bond yields, can be explained by our topic-specific uncertainty indices, we estimated various Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) models (Nelson, 1991). As before, we considered the interval from 2 January 2019 to 1 May 2020, which is characterized by a period of extremely high volatility that goes from February 2020 to the end of our sample. The choice of a model of the ARCH family is suggested by the desire to explain phases of high and low volatility in the interval under study. An advantage of the EGARCH model over the more standard

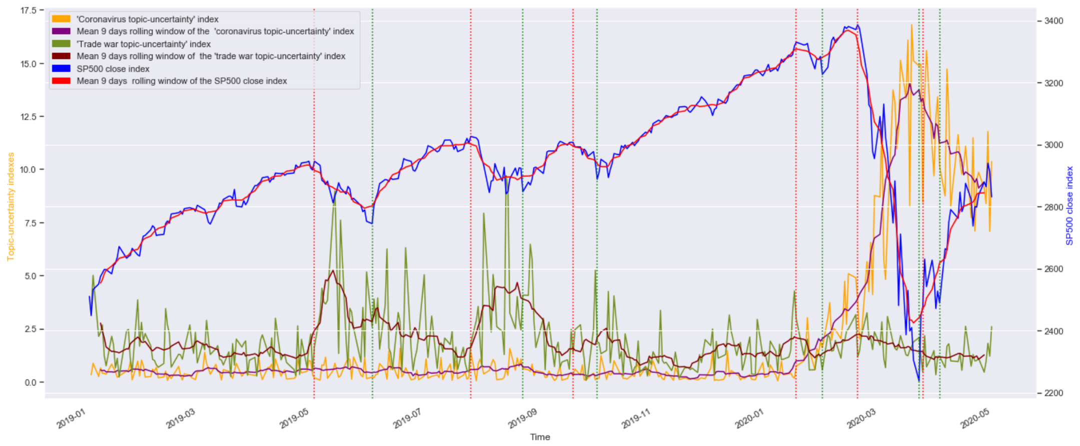


Figure 2: The yellow line represents the 'coronavirus' uncertainty index; the purple line is the moving average with a 9-day rolling window. The green line represents the 'trade war' uncertainty index; the brown line is the moving average with a 9-day rolling window. The blue line represents the S&P 500 closing price index; the red line is the moving average with a 9-day rolling window. The vertical dotted red lines indicate some of the local maxima of the S&P 500 closing price index, whereas the vertical dotted green lines represent some of the local minima of the S&P 500 closing price index.

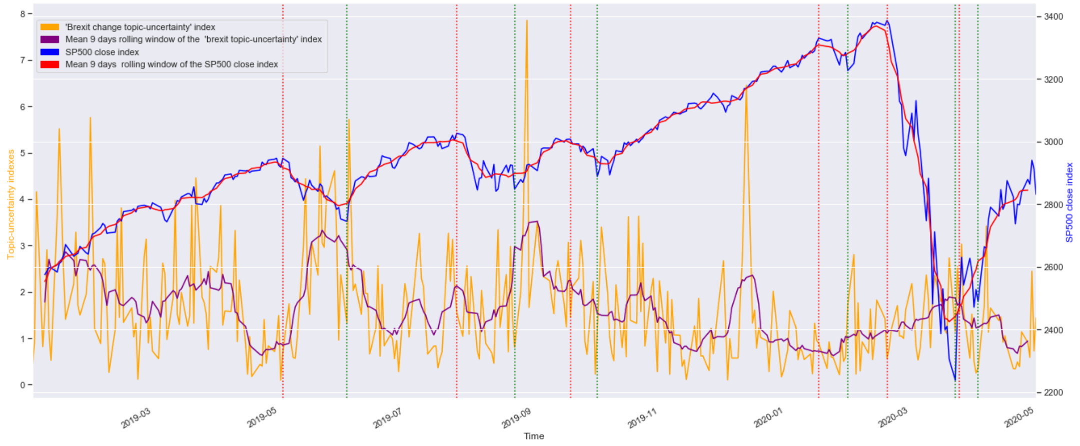


Figure 3: The yellow line represents the 'Brexit' uncertainty index; the purple line is the moving average with a 9-day rolling window. The blue line represents the S&P 500 closing price index; the red line is the moving average with a 9-day rolling window. The vertical dotted red lines indicate some of the local maxima of the S&P 500 closing price index, whereas the vertical dotted green lines represent some of the local minima of the S&P 500 closing price index.

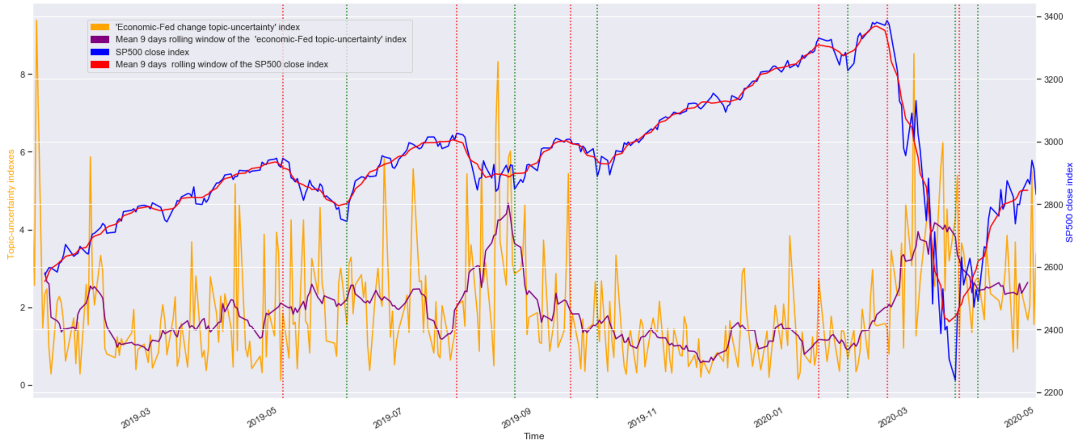


Figure 4: The yellow line represents the 'economic-Fed' uncertainty index; the purple line is the moving average with a 9-day rolling window. The blue line represents the S&P 500 closing price index; the red line is the moving average with a 9-day rolling window. The vertical dotted red lines indicate some of the local maxima of the S&P 500 closing price index, whereas the vertical dotted green lines represent some of the local minima of the S&P 500 closing price index.

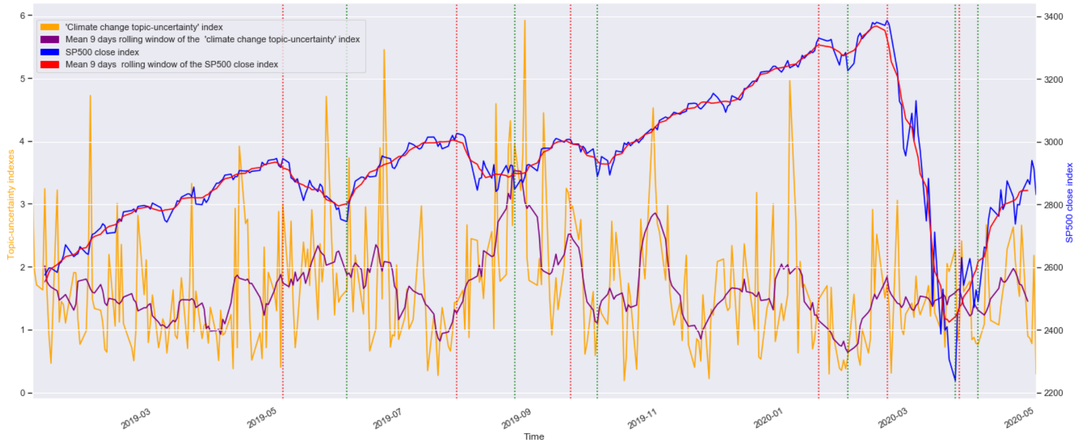


Figure 5: The yellow line represents the 'climate change' uncertainty index; the purple line is the moving average with a 9-day rolling window. The blue line represents the S&P 500 closing price index; the red line is the moving average with a 9-day rolling window. The vertical dotted red lines indicate some of the local maxima of the S&P 500 closing index, whereas the vertical dotted green lines represent some of the local minima of the S&P 500 closing price index.

GARCH model is its ability to capture asymmetric behaviours, also known as leverage effects, that is, to model the asymmetric effect on volatility of good and bad news. Specifically, a positive leverage means that high positive returns are followed by larger increases in volatility than in the case of negative returns of the same size, whereas a negative leverage means that high negative returns are followed by larger increases in volatility than in the case of positive returns.

In particular, for a given financial index f, let us consider the returns

$$\Delta C_{f,t} = \frac{C_{f,t} - C_{f,t-1}}{C_{f,t-1}} \cdot 100, \tag{4}$$

where $C_{f,t}$ is the daily closing price of the financial index f at time t. We first investigate how much of the mean and volatility of the S&P 500 returns can be explained by each of our topic-specific uncertainty indices: 'trade war', 'coronavirus', 'Brexit', 'climate change' and 'economic-Fed'. To do this, we estimated a separate EGARCH model for each of these topic-specific uncertainty index, considering the same combination of explanatory variables used by Mamaysky (2020) in his contemporaneous regressions. Precisely, we estimated the following EGARCH(1,1) model for the S&P 500 returns $\Delta C_{S,t}$ and for each of our topic-specific uncertainty indices:

$$\begin{aligned}\Delta C_{S,t} = {} & b_0 + b_1 \Delta C_{S,t-1} + b_2 T_{i,t} + b_3 T_{i,t}(\text{VIX}_{t-1} - \overline{\text{VIX}}) \\ & + b_4 \text{VIX}_{t-1} + \theta \epsilon_{t-1} + \epsilon_t,\end{aligned} \tag{5}$$

$$\begin{aligned}\ln \sigma_t^2 = {} & \omega + b_5 T_{i,t} + b_6 T_{i,t}(\text{VIX}_{t-1} - \overline{\text{VIX}}) + b_7 \text{VIX}_{t-1} \\ & + \beta \ln \sigma_{t-1}^2 + \alpha \left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| + \gamma \frac{\epsilon_{t-1}}{\sigma_{t-1}}.\end{aligned} \tag{6}$$

The mean equation in (5), measuring the influence of the explanatory variables on the mean returns of the S&P 500, includes as explanatory variables: the ith topic-specific uncertainty index $T_{i,t}$, the product of this index and the difference between the lag value VIX_{t-1} and the mean value $\overline{\text{VIX}}$ of the VIX index, and the lag value of the VIX index. Similarly for the conditional variance equation with asymmetric effects, given in (6), which measures the effect of the explanatory variables on the volatility in the returns of the S&P 500. In the equations, ϵ_t refers to the zero mean and unit variance independent and identically distributed error term (ARCH error), whereas σ_t indicates the conditional variance (GARCH term). Moreover, the coefficient ω is a constant, β is the GARCH coefficient (persistence term), α is the coefficient of the ARCH term, and γ indicates the asymmetric or leverage effect.

Table 9 shows the estimates and standard errors of the parameters of the EGARCH(1,1) model in Equations (5) and (6), for each of the five topic-specific uncertainty indices used as an explanatory variable in the models. The figures show the effect of a unit increase in a given topic-specific uncertainty index on the mean and volatility of the returns of the S&P 500. As expected, we see that the 'trade war' and 'coronavirus' uncertainty indices have a negative effect on the mean, and a positive effect on the volatility, of the returns of the S&P 500, though the volatility coefficient of the 'trade war' uncertainty index is not significant. Our findings about the 'trade war' uncertainty index are similar to those of Burggraf et al. (2020), which suggest that tweets from the US President Donald Trump's Twitter account related to the trade war between US and China had a positive effect on the VIX index and a negative effect on the S&P 500 returns. Moreover, our findings about the 'coronavirus' uncertainty index are in agreement, among others, with those of Baker et al. (2020) and Haroon and Rizvi (2020), which find that the panic during the coronavirus crisis at the beginning of 2020 is associated with an increase in volatility.

Table 9 also shows that a rise in the 'Brexit' uncertainty index implies an increase in the mean of S&P 500 returns; in other words, uncertain news about Brexit did not cause negative effects on these returns. On the other hand, the 'climate change' uncertainty index seems to have a small negative effect on the mean returns of the S&P 500. Furthermore, the 'economic-Fed' uncertainty index, which accounts for news on the actions of the Fed and of the US government, seems to be positively associated with both the mean and the volatility of the S&P 500 returns. Indeed, this uncertainty index seems to incorporate news about possible future actions of the Fed and the US government in addressing economic turmoils during periods of great uncertainty. A greater value of this index might be due to the negative economic scenarios associated with the actions of the Fed and US government, which are, these latter, immediately absorbed by the markets with changes of companies' stock value.

As we can see from the results reported at the bottom of Table 9, the models related to the 'coronavirus', 'trade war' and 'economic-Fed' uncertainty indices passed numerous tests, including the weighted Ljung-Box test, which means that the standardized residuals are not autocorrelated, and the weighted ARCH LM test, which says that the EGARCH(1,1) models are correctly fitted. The two EGARCH(1,1) models with the best fit are those for the 'coronavirus' and 'trade war' uncertainty indices. In comparison with the other three models, these two uncertainty indices obtain the highest log-likelihood and the smallest values for the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). These findings seem in agreement with the graphs in Figure 2, which suggest a negative correlation between the 'trade war' and 'coronavirus' uncertainty indices and the mean returns of the S&P 500. In particular, the 'trade war' uncertainty index

Table 9: Estimates and standard errors (in parentheses) of the parameters of the EGARCH(1,1) model in Equations (5) and (6), for each of the five topic-specific uncertainty indices. Each column header indicates the topic-specific uncertainty index used as an explanatory variable in the model. The dependent variable in all five models are the returns of the S&P 500.

	Trade War	Coronavirus	Brexit	Climate	Economic-Fed
b_0	-0.10^{**}	-0.16^{***}	0.61^{***}	0.01^{***}	-0.04^{***}
	(0.03)	(0.00)	(0.00)	(0.00)	(0.00)
b_1	0.83^{***}	0.17^{***}	-0.53^{***}	-0.53^{***}	-0.53^{***}
	(0.05)	(0.00)	(0.00)	(0.00)	(0.00)
b_2	-0.10^{***}	-0.01^{***}	0.12^{***}	-0.02^{***}	0.28^{***}
	(0.02)	(0.00)	(0.00)	(0.00)	(0.00)
b_3	-0.01	-0.00^{***}	0.02^{***}	-0.00^{***}	-0.00^{***}
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
b_4	0.02^{***}	0.01^{***}	-0.04^{***}	-0.01^{***}	-0.01^{***}
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
θ	-0.87^{***}	-0.20^{***}	0.18^{***}	0.18^{***}	0.17^{***}
	(0.04)	(0.00)	(0.00)	(0.00)	(0.00)
ω	-0.56	-0.45^{***}	3.29^{***}	3.28^{***}	3.28^{***}
	(0.34)	(0.00)	(0.00)	(0.00)	(0.00)
b_5	0.09	0.06^{***}	0.08^{***}	-0.05^{***}	0.16^{***}
	(0.05)	(0.00)	(0.00)	(0.00)	(0.00)
b_6	0.00	-0.00^{***}	-0.00^{***}	0.04^{***}	-0.05^{***}
	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)
b_7	0.02	0.02^{***}	-0.21^{***}	-0.14^{***}	-0.21^{***}
	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)
β	0.76^{***}	0.83^{***}	0.90^{***}	0.90^{***}	0.90^{***}
	(0.10)	(0.00)	(0.00)	(0.00)	(0.00)
α	-0.34^{***}	-0.48^{***}	0.07^{***}	0.04^{***}	0.04^{***}
	(0.06)	(0.00)	(0.00)	(0.00)	(0.00)
γ	0.15	-0.44^{***}	0.13^{***}	0.11^{***}	0.11^{***}
	(0.11)	(0.00)	(0.00)	(0.00)	(0.00)
Log likelihood	-425.55	-416.87	-1766.02	-2698.76	-2646.47
AIC	2.61	2.56	10.59	16.14	15.83
BIC	2.76	2.71	10.74	16.29	15.98
Ljung-Box Test (p-value in parentheses)					
Lag[1]	0.01027	0.3799	$1.872e-04$	0.05225	0.001937
	(0.9193)	(0.5377)	$(9.891e-01)$	$(8.192e-01)$	(0.9649)
Lag[2*(p+q)+(p+q)-1][5]	0.57671	1.0545	$1.877e+00$	19.62855	1.655162
	(1.0000)	(1.0000)	$(9.768e-01)$	$(0.000e+00)$	(0.9937)
Lag[4*(p+q)+(p+q)-1][9]	3.20533	4.5854	$2.417e+01$	30.59862	2.760406
	(0.8571)	(0.5507)	$(2.907e-10)$	$(3.897e-14)$	(0.9236)
ARCH LM Test (p-value in parentheses)					
ARCH Lag[3]	0.4612	0.1345	0.01335	15.11	0.003194
	(0.4971)	(0.71379)	(0.908002)	$(1.017e-04)$	(0.9549)
ARCH Lag[5]	0.5281	1.3910	0.57237	15.44	0.018647
	(0.8753)	(0.62143)	(0.862059)	$(2.970e-04)$	(0.9999)
ARCH Lag[7]	0.7583	8.0744	15.10207	19.69	0.031256
	(0.9494)	(0.05051)	(0.001089)	$(7.647e-05)$	(1.0000)

p-value: *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$

seems to explain much of the behavior of the S&P 500 during 2019, whereas the 'coronavirus' uncertainty index seems to best explain the beginning of 2020. Overall, these two indices seem to do better than the other three uncertainty indices in explaining the returns of the S&P 500 from the beginning of 2019 to the and of April 2020.

To deepen the investigation on the relationship between uncertainty in the news and behaviour of the financial markets, we estimated some other EGARCH models to study the joint effect of the 'coronavirus' and 'trade war' uncertainty indices on the returns of some US financial indices, in particular the S&P 500 index, the Dow Jones index, the Nasdaq Composite index, the VIX index as well as the US 10-year Treasury bonds yields. Precisely, for each of these five financial indices we considered the following EGARCH(1,1) model:

$$\begin{aligned}\Delta C_{f,t} &= b_0 + b_1 \Delta C_{f,t-1} + b_2 T_{\mathrm{C},t} + b_3 T_{\mathrm{W},t} + b_4 T_{C,t}(\mathrm{VIX}_{t-1} - \overline{\mathrm{VIX}}) \\ &\quad + b_5 T_{W,t}(\mathrm{VIX}_{t-1} - \overline{\mathrm{VIX}}) + b_6 \mathrm{VIX}_{t-1} + \theta \epsilon_{t-1} + \epsilon_t,\end{aligned} \tag{7}$$

$$\begin{aligned}\ln \sigma_t^2 &= \omega + b_7 T_{\mathrm{C},t} + b_8 T_{\mathrm{W},t} + b_9 T_{C,t}(\mathrm{VIX}_{t-1} - \overline{\mathrm{VIX}}) + b_{10} T_{W,t}(\mathrm{VIX}_{t-1} - \overline{\mathrm{VIX}}) \\ &\quad + b_{11} \mathrm{VIX}_{t-1} + \beta \ln \sigma_{t-1}^2 + \alpha \left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| + \gamma \frac{\epsilon_{t-1}}{\sigma_{t-1}},\end{aligned} \tag{8}$$

where $T_{\mathrm{C},t}$ and $T_{\mathrm{W},t}$ refer to the 'coronavirus' and 'trade war' uncertainty indices, respectively, and $\Delta C_{f,t}$ indicates the returns of the financial index f at time t.

Table 10 shows the estimates and standard errors of the parameters of the EGARCH(1,1) model in Equations (7) and (8), for each of the five financial indices used for the dependent variable in the mean equation. As expected, we see that both the 'coronavirus' and

'trade war' uncertainty indices have a negative effect on the mean, and a positive effect on the volatility, of the returns of the S&P 500. In particular, we notice that an increase in the 'trade war' uncertainty index has a greater negative effect on the mean returns of the S&P 500 than an increase in the 'coronavirus' uncertainty index. Let us also observe that the 'coronavirus' uncertainty index has a negative effect on the mean returns of the Nasdaq, but not on that of the Dow Jones, and vice-versa for the 'trade war' uncertainty index. Moreover, we see that the mean returns of the VIX is positively affected by the 'coronavirus' and 'trade war' uncertainty indices. Lastly, as far as the 10-year US Treasury bond yields are concerned, the results show that an increase in the 'coronavirus' and 'trade war' uncertainty indices leads to a decrease in their mean returns. In line with common opinion, we can reasonably argue that investors may see US bonds as a safe refuge during periods of high uncertainty.

Table 10: Estimates and standard errors (in parenthesis) of the parameters of the EGARCH(1,1) model in Equations (7) and (8), for each of the five financial indices. Each column header indicates the financial index used for the dependent variable in the mean equation; the dependent variable is the returns of the index. In all five models, the explanatory variables are the 'coronavirus' and 'trade war' topic-specific uncertainty indices.

	S&P 500	Nasdaq	Dow Jones	VIX	Treasury yields 10 years
b_0	-0.87^{***}	0.86^{***}	-1.16^{***}	1.45^{***}	-2.19^{***}
	(0.00)	(0.02)	(0.00)	(0.33)	(0.22)
b_1	-0.76^{***}	0.06^{***}	-0.60^{***}	-0.48^{***}	-0.87^{***}
	(0.00)	(0.01)	(0.00)	(0.12)	(0.05)
b_2	-0.03^{***}	-0.58^{***}	0.08^{***}	0.63^{***}	-0.41^{**}
	(0.00)	(0.00)	(0.00)	(0.08)	(0.15)
b_3	-0.17^{***}	0.03^{***}	-0.22^{***}	0.45^{***}	-0.38^{***}
	(0.00)	(0.00)	(0.00)	(0.11)	(0.10)
b_4	-0.00^{***}	-0.03^{***}	-0.02^{***}	0.02^{***}	-0.00
	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)
b_5	-0.02^{***}	0.01^{***}	-0.05^{***}	-0.15^{***}	-0.06^{**}
	(0.00)	(0.00)	(0.00)	(0.03)	(0.02)
b_6	0.07^{***}	-0.03^{***}	0.07^{***}	-0.22^{***}	0.16^{***}
	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)
θ	0.76^{***}	-0.38^{***}	0.29^{***}	0.34^{**}	0.83^{***}
	(0.00)	(0.00)	(0.00)	(0.12)	(0.07)
ω	0.22^{***}	-1.88^{***}	-1.86^{***}	0.48^{***}	0.21
	(0.00)	(0.01)	(0.00)	(0.04)	(0.40)
b_7	0.07^{***}	-0.78^{***}	-0.88^{***}	0.09^{***}	0.14^{**}
	(0.00)	(0.00)	(0.00)	(0.00)	(0.05)
b_8	0.08^{***}	0.08^{***}	0.15^{***}	0.05^{***}	0.09
	(0.00)	(0.00)	(0.00)	(0.01)	(0.06)
b_9	-0.00^{***}	-0.34^{***}	-0.35^{***}	-0.00	-0.00
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
b_{10}	0.01^{***}	0.01^{***}	0.03^{***}	-0.01^{***}	0.00
	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)
b_{11}	-0.03^{***}	0.10^{***}	0.10^{***}	-0.01^{***}	0.03
	(0.00)	(0.00)	(0.00)	(0.00)	(0.03)
β	0.89^{***}	1.00^{***}	0.93^{***}	0.87^{***}	0.41^{*}
	(0.00)	(0.00)	(0.00)	(0.00)	(0.19)
α	-0.35^{***}	-0.23^{***}	0.22^{***}	0.38^{***}	-0.14
	(0.00)	(0.00)	(0.00)	(0.04)	(0.09)
γ	-0.31^{***}	0.58^{***}	0.39^{***}	-0.13^{**}	0.70^{***}
	(0.00)	(0.00)	(0.00)	(0.05)	(0.16)
Log likelihood	-409.42	-1838.45	-1774.76	-1133.32	-811.42
AIC	2.54	11.04	10.67	6.85	4.93
BIC	2.73	11.24	10.86	7.04	5.12
Ljung-Box Test (p-value in parentheses)					
Lag[1]	1.058	0.4831	0.1755	0.6897	0.3377
	(0.3037)	(0.487)	(0.6752)	(0.4063)	(0.5611)
Lag[2*(p+q)+(p+q)-1][5]	1.640	273.7142	160.8419	1.2335	0.7844
	(0.9943)	(0.000)	(0.0000)	(0.9998)	(1.0000)
Lag[4*(p+q)+(p+q)-1][9]	5.917	447.9748	222.9947	3.3749	3.7797
	(0.2703)	(0.000)	(0.0000)	(0.8260)	(0.7415)
ARCH LM Test (p-value in parentheses)					
ARCH Lag[3]	0.218	0.2553	0.07989	0.1928	0.8753
	(0.6406)	(0.6134)	$(7.774e-01)$	(0.6606)	(0.3495)
ARCH Lag[5]	1.181	141.7525	36.51865	2.3831	3.3546
	(0.6802)	(0.0000)	$(1.365e-09)$	(0.3926)	(0.2423)
ARCH Lag[7]	4.256	216.2402	44.58176	3.6654	6.6470
	(0.3111)	(0.0000)	$(2.090e-11)$	(0.3974)	(0.1033)

p-value: *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$

The bottom of Table 10 shows that the models for the S&P 500, the VIX and the 10-year US Treasury bond yields passed both the weighted Ljung-Box test, which indicates that the standardized residuals are not autocorrelated, and the weighted ARCH LM test, which means that the EGARCH process is correctly fitted. By far, the EGARCH(1,1) model with the best fit is that for the S&P 500. Comparing it with the other four models, this model has the highest log-likelihood and the smallest values for the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC).

5.4 Conclusions

In this paper we use unsupervised machine learning techniques to construct text measures able to explain recent past movements in US financial markets. Our raw text data are the headlines and snippets of the articles of the New York Times from 2 January 2019 to 1 May 2020. We first use LDA to infer the content (topics) of the articles and thus to obtain daily indices on the presence of these topics in the New York Times. Then we use Word Embedding (implemented with the Skip-gram model) and K-Means to construct a daily uncertainty measure. Thus, we combine all these measures to obtain daily topic-specific uncertainty indices. In particular, we obtain five uncertainty indices related to news about 'coronavirus', 'trade war', 'Brexit', 'economic-Fed' and 'climate change', capturing the daily degree of uncertainty in these topics.

To quantify how much of the behaviour of the S&P 500 index can be explained by uncertainty in the news, we estimated an EGARCH(1,1) model for each of our five topic-specific uncertainty indices. We verify that the 'coronavirus' and 'trade war' uncertainty indices are negatively associated with the mean, and positively associated with the volatility, of the returns of the S&P 500. Also, we find that the 'climate change' and 'economic-Fed' uncertainty indices are negatively and positively, respectively, associated with the mean of the S&P 500 returns. This suggests that news about economic measures of the Fed and the US government has a positive effect on the S&P 500 in days of uncertainty. Overall, we can argue that the 'trade war' uncertainty index explains much of the behavior of the S&P 500 returns during 2019, whereas the 'coronavirus' uncertainty index explains most of the movements of the S&P 500 index during the first four months of 2020.

To further investigate how much these two uncertainty indices explain the behaviour of the US financial markets, we estimated, using these two indices as explanatory variables, some other EGARCH(1,1) models, one for each of the following financial indices (as dependent variable): the S&P 500, the Nasdaq, the Dow Jones, the VIX and the US 10-year Treasury bond yields. We find that the 'coronavirus' and 'trade war' uncertainty indices have a negative effect on the mean, and a positive effect on the volatility, of the returns of the S&P 500. We also find that these two uncertainty indices have a positive effect both on the mean and the volatility of the returns of the VIX index.

Future research might address some issues raised by the use of the headlines and the snippets instead of the (lacking) full text of the articles in the New York Times. It would also be interesting to study the robustness of our analysis on a longer period of time. From a methodological point of view, it should also be explored the use of other machine learning methods for the construction of text measures such as Dynamic Topic Models (Blei and Lafferty, 2006) and Support Vector Machines. Similarly, more sophisticated GARCH-MIDAS models could be used to incorporate, as explanatory variables, macroeconomic and other variables sampled at different frequency.

Bibliography

[1] Albulescu, C. (2020). Coronavirus and financial volatility: 40 days of fasting and fear. *ArXiv Preprint*, 2003.04005.

[2] Ardizzi, G., Emiliozzi, S., Marcucci, J., and Monteforte, L. (2019). News and consumer card payments. *Banca d'Italia Working Paper*, 1233.

[3] Azqueta-Gavaldon, A., Hirschbühl, D., Onorante, L., and Saiz, L. (2020). Economic policy uncertainty in the Euro area: an unsupervised machine learning approach. *European Central Bank Working Paper Series*, 2359.

[4] Baker, S. R., Bloom, N., Davis, S. J., and Terry, S. J. (2020). Covid-induced economic uncertainty. *National Bureau of Economic Research*, No. w26983.

[5] Blei, D. M., and Lafferty, J. D. (2006). Dynamic topic models. *Proceedings of the 23rd International Conference on Machine Learning*, 113–120.

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

[7] Burggraf, T., Fendel, R., and Huynh, T. L. D. (2020). Political news and stock prices: evidence from Trump's trade war. *Applied Economics Letters*, 27, 1485–1488.

[8] Chakraborty, C., and Joseph, A. (2017). Machine learning at central banks. *Bank of England Staff Working Paper*, 865.

[9] Griffiths, T. L., and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101, 5228–5235.

[10] Hansen, S., and McMahon, M. (2016). Shocking language: understanding the macroeconomic effects of central bank communication. *Journal of International Economics*, 99, S114–S133.

[11] Hansen, S., McMahon, M.,and Prat, A. (2018). Transparency and deliberation within the FOMC: a computational linguistics approach. *The Quarterly Journal of Economics*, 133, 801–870.

[12] Haroon, O., and Rizvi, S. A. R. (2020). COVID-19: Media coverage and financial markets behavior: a sectoral inquiry. *Journal of Behavioral and Experimental Finance*, 100343.

[13] Kalamara, E., Turrell, A., Redl, C., Kapetanios, G., and Kapadia, S. (2020). Making text count: economic forecasting using newspaper text. *Bank of England Staff Working Paper*, 674.

[14] Loughran, T. and McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *Journal of Finance*, 66, 35–65.

[15] Mamaysky, H. (2020). Financial markets and news about the coronavirus. *SSRN Working Paper*, 3565597.

[16] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *Arxiv Preprint*, 1301.3781.

[17] Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: a new approach. *Econometrica*, 59, 347–370.

[18] Soto, P. E. (2021). Breaking the word bank: measurement and effects of bank level uncertainty. *Journal of Financial Services Research*, 1–2/2021.

[19] Zhu, S., Liu, Q., Wang, Y., Wei, Y., and Wei, G. (2019). Which fear index matters for predicting US stock market volatilities: text-counts or option based measurement? *Physica A: Statistical Mechanics and its Applications*, 536, 122567.

Chapter 6

Supplementary Material - Collapsing Financial Markets: Unsupervised Modelling of the Coronavirus and Trade War News

6.1 New York Times Database

This section explains how we downloaded the database of articles of the New York Times. We then explain the construction and the 'cleaning' process of the articles of the New York Times.

6.1.1 New York Times database: download

This paper uses jointly headlines and snippets from the English articles in the New York Times from 1 January, 2019 to 1 May, 2020. We download the headlines and the snippets of the articles using the New York Times API with the python code of the web page medium.[1] To download the headlines and snippets of the articles, we create an account at the New York Times web page to obtain an API key.'developer.nytimes.com' We then add the API key to the python code. The following lines show the python code - 'new_york_times_API_dowload.py' - to download the articles of May 2020.

```
import requests
import pandas as pd
import pyjq
```

[1] https://medium.com/@danalindquist/using-new-york-times-api-and-jq-to-collect-news-data-a5f386c7237b

```python

#We should obtain an API key of the New York Times in (developer.nytimes.com) by creating your own account.
your_key = '...'

#We specify the month of the year that we want to dowload. In this case, we download the data of May 2020 (2020/5).
url = 'https://api.nytimes.com/svc/archive/v1/2020/5
.json?api-key='+your_key

#We download the specified url in json file.
r = requests.get(url)
json_data = r.json()

#We extract data from json file.
copyright = pyjq.all('.copyright', json_data)
num_docs = pyjq.all('.response .docs | length',json_data)[0]

#We are interested in the snippet, headline, publication date and news desk for the documents.
jq_query = f'.response .docs [] | {{the_snippet: .snippet, the_headline: .headline .main, the_date: .pub_date, the_news_desk: .news_desk}}'
output = pyjq.all(jq_query, json_data)

#We include the data in a DataFrame 'df'.
df = pd.DataFrame(output)

#We eliminate duplicates of the articles.
df2 = df.drop_duplicates(subset='the_snippet')

#Joining the headline and the snippet in the same column.
df2['speech'] = df2['the_headline'].str.cat(df2['the_snippet'],sep=" ")

#Saving the output in a csv file.
df2.to_csv('new_york_times_2020_may_up.csv')
```

6.1.2 New York Times database: construction database

To construct the database of the New York Times, we merge all the databases of all the months. We assign the articles published after 4:00 pm, when the stock exchange closes, to the following observation. We also assign the articles appearing over the weekend to the following observation, usually next Monday. We then download the S&P 500 index in Yahoo finance and merge it with the New York Times database. Keeping only the days that the stock exchange was opened. We assign the articles that occur in a day that the New York Stock Exchange was closed to the following observation. Moreover, the python code is comprised in the supplementary material like ‘coronavirus_newspapers_database.py’ and the pre-process database with the name ‘new_york_times_merged_may2020.csv’.

6.1.3 New York Times database: eliminating non-relevant sections

We eliminate several sections that do not provide relevant information for the stock exchange such as ‘Travel’, ‘Style’ or ‘Sports’ after personally checking them. The final output is saved as ‘coronavirus_nytimes _withoutsections_mayupdated.csv’. We comprise the python code in the supplementary material folder as ‘coronavirus_restricting _database.py’, and we show it in the following lines.

```python
import pandas as pd

#Importing the New York Times database as the DataFrame 'data'.
data = pd.read_csv("new_york_times_merged_may2020.csv", sep = ",", encoding="utf-8")

#We eliminate the non-relevant sections.
data = data[data.the_news_desk != 'Well']
data = data[data.the_news_desk != 'Weekend']
data = data[data.the_news_desk != 'Travel']
data = data[data.the_news_desk != 'Times Insider']
data = data[data.the_news_desk != 'Theater']
data = data[data.the_news_desk != 'The Weekly']
data = data[data.the_news_desk != 'The Learning Network']
data = data[data.the_news_desk != 'TStyle']
data = data[data.the_news_desk != 'T Magazine / Fashion & Beauty']
data = data[data.the_news_desk != 'T Magazine / Art']
data = data[data.the_news_desk != 'T Magazine']
data = data[data.the_news_desk != 'SundayReview']
data = data[data.the_news_desk != 'Styles']
data = data[data.the_news_desk != 'Style']
data = data[data.the_news_desk != 'Sports']
data = data[data.the_news_desk != 'SpecialSections']
data = data[data.the_news_desk != 'Society']
data = data[data.the_news_desk != 'Real Estate']
data = data[data.the_news_desk != 'RealEstate']
data = data[data.the_news_desk != 'Reader Center']
data = data[data.the_news_desk != 'Photo']
data = data[data.the_news_desk != 'Parenting']
data = data[data.the_news_desk != 'Obituaries']
data = data[data.the_news_desk != 'Obits']
data = data[data.the_news_desk != 'None']
data = data[data.the_news_desk != 'New York']
data = data[data.the_news_desk != 'Multimedia/Photos']
data = data[data.the_news_desk != 'Movies']
data = data[data.the_news_desk != 'Metropolitan']
data = data[data.the_news_desk != 'Metro']
data = data[data.the_news_desk != 'Magazine']
data = data[data.the_news_desk != 'Live']
data = data[data.the_news_desk != 'Letters']
data = data[data.the_news_desk != 'Learning']
data = data[data.the_news_desk != 'Insider']
data = data[data.the_news_desk != 'Health']
data = data[data.the_news_desk != 'Guides']
data = data[data.the_news_desk != 'Graphics']
data = data[data.the_news_desk != 'Gender']
data = data[data.the_news_desk != 'Games']
data = data[data.the_news_desk != 'Food']
data = data[data.the_news_desk != 'Fashion & Style']
data = data[data.the_news_desk != 'Fashion']
data = data[data.the_news_desk != 'Express']
data = data[data.the_news_desk != 'Dining']
data = data[data.the_news_desk != 'Culture']
data = data[data.the_news_desk != 'Crosswords & Games']
data = data[data.the_news_desk != 'Corrections']
data = data[data.the_news_desk != 'Briefing']
data = data[data.the_news_desk != 'Books']
data = data[data.the_news_desk != 'BookReview']
data = data[data.the_news_desk != 'AtHome']
data = data[data.the_news_desk != 'Arts&Leisure']

#We save the DataFrame 'data' in a csv file.
data.to_csv("coronavirus_nytimes_withoutsections
  ↪  _mayupdated.csv")
```

6.2 Latent Dirichlet Allocation

The file ‘coronavirus_LDA_60t_12000_1000_withoutsections.py’ comprises the python code to apply LDA to the articles of the New York Times from 1 January, 2019 to 1 May, 2020. To apply Latent Dirichlet allocation, we use most of the python code provided by the Professor Stephen Hansen of the Imperial College Business School.[2] The results are not reproducible. However, the results tend always to be similar after several trials. The following list shows the name of the different outputs included in the supplementary material folder. An explanation of each document is given within brackets.

1. ‘topic_description_nyt_60t_reduced.csv’ (LDA output: words per topic);
2. ‘final_output_coronavirus_t60.csv’ (LDA output: topics per document);
3. ‘final_output_agg_coronavirus_t60.csv’ (LDA output: topics per day);
4. ‘df_ranking.csv’ (LDA output: each stem of this file is ranked following document frequency);
5. ‘tfidf_ranking.csv’ (LDA output: each stem of this file is ranked following the tf-idf measure).

The python code to estimate LDA with the corpus of articles of the New York Times is the following:

```
import pandas as pd
import topicmodels
import matplotlib.pyplot as plt


#Importing the dataset of the New York Times as DataFrame
  ↪  'data'.
```

[2] https://github.com/sekhansen

```
data = pd.read_csv("coronavirus_nytimes_withoutsections_mayupdated.csv",sep = ",", encoding="utf-8")

#Changing 'date' column format to date.
data['the_date'] = pd.to_datetime(data['the_date'],infer_datetime_format=True,dayfirst=True)

#Creating columns for 'year', 'month' and 'day' with the 'date' column.
data['year'] = data['the_date'].dt.year
data['day'] = data['the_date'].dt.day
data['month'] = data['the_date'].dt.month

#Using the long list of English stopwords.
docsobj = topicmodels.RawDocs(data.speech, "long")
docsobj.token_clean(1)

#We remove the stopwords.
docsobj.stopword_remove("tokens")

#We stem the corpus.
docsobj.stem()
docsobj.stopword_remove("stems")

#We rank these stems  according to the term frequency-inverse document frequency (tf-idf).
docsobj.term_rank("stems")

#We disregard all stems that have a value of the tf-idf ranking of 12,000 or lower.
docsobj.rank_remove("tfidf", "stems", docsobj.tfidf_ranking[12000][1])

#Plotting the tf-idf ranking.
plt.plot([x[1] for x in docsobj.tfidf_ranking])

#Printing number of unique and total stems in the database.
all_stems = [s for d in docsobj.stems for s in d]
print("number of unique stems = %d" % len(set(all_stems)))
print("number of total stems = %d" % len(all_stems))

#Latent Dirichelt Allocation estimation with 60 topics.
ldaobj = topicmodels.LDA.LDAGibbs(docsobj.stems, 60)

#we run twice 20 samples from points in the chain that are
 ↪   thinned with a thinning interval of 50.
ldaobj.sample(1000, 50, 20)
print(ldaobj.perplexity())
ldaobj.sample(1000, 50, 20)
print(ldaobj.perplexity())


ldaobj.samples_keep(4)
ldaobj.topic_content(20)

dt = ldaobj.dt_avg()
tt = ldaobj.tt_avg()
ldaobj.dict_print()

data = data.drop('speech', 1)

#LDA output: topics per document.
for i in range(ldaobj.K):
    data['T' + str(i)] = dt[:, i]
data.to_csv("final_output_coronavirus_t60.csv",
 ↪   index=False)

#Querying documents by minutes. LDA output: topics per day.
data['speech'] = [' '.join(s) for s in docsobj.stems]
aggspeeches = data.groupby(['year',
 ↪   'month','day'])['speech'].\
    apply(lambda x: ' '.join(x))
aggdocs = topicmodels.RawDocs(aggspeeches)

queryobj = topicmodels.LDA.QueryGibbs(aggdocs.tokens,
 ↪   ldaobj.token_key,
                                      ldaobj.tt)
queryobj.query(10)
queryobj.perplexity()
queryobj.query(30)
queryobj.perplexity()

dt_query = queryobj.dt_avg()
aggdata = pd.DataFrame(dt_query, index=aggspeeches.index,
                       columns=['T' + str(i) for i in
                        ↪ range(queryobj.K)])
aggdata.to_csv("final_output_agg_coronavirus_t60.csv")

  \vspace{\baselineskip}
```

6.3 Skip-Gram and K-Means

To create sentiment measures, we apply the Skip-Gram model and K-Means to build a list of words with similar meaning to the word ‘uncertainty’. This list can be seen as an uncertainty dictionary, which is used to construct a daily uncertainty index by counting the frequency of its words in all the articles of each day. The python code to estimate the Skip-Gram model and the K-Means is included in the supplementary material folder with the name ‘coronavirus skipgram k-means.py’. Some articles for example on local crime or New York local news discuss topics we are not interested in. These articles could bias our results since they do not normally have an effect on financial markets. Thus, we eliminate all the articles that have the highest LDA topic probability for one of these topics.[3]

Most of the code to obtain the Word Embeddings with Skip-Gram is part of the code provided in the github webpage of Florian Leitner.[4] We use Word2Vec of the package gensim to apply the Skip-Gram model.

K-Means is implemented with the code provided by the webpage https://ai.intelligentonlinetools.com/. The article is titled ‘K Means Clustering Example with Word2Vec in Data Mining or Machine Learning’.

To make the Skip-Gram results reproducible in python 3, the seed is set as ‘set PYTHONASHSEED=0’ in the terminal before opening python. We then open python from the terminal. The following lines show the python code to estimate the Skip-Gram model and K-Means:

[3] Topics 0, 5, 6, 7, 8, 9, 11, 18, 21, 22, 27, 28, 34, 35, 37, 43, 44, 48, 57 and 59.

[4] https://github.com/fnl/asdm-tm-class, Florian Leitner teaches the ‘text mining’ course of the Madrid UPM Machine Learning and Advanced Statistics Summer School

```python
import pandas as pd
import string
import numpy as np
import re
from pprint import pprint
import gensim
import gensim.corpora as corpora
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel
from gensim.models import Word2Vec
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s :
 ↪  %(message)s', level=logging.ERROR)
import warnings
warnings.filterwarnings("ignore",category=DeprecationWarning)

# Plotting tools.
import pyLDAvis
import matplotlib.pyplot as plt

import nltk
from nltk.cluster import KMeansClusterer
from nltk.stem import SnowballStemmer
import nltk; nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words = stopwords.words('english')

from IPython.display import HTML
from sklearn import cluster
from sklearn import metrics
import pickle
import random


#Loading LDA output 'topics per document' as 'df1'
 ↪  DataFrame.
df1 = pd.read_csv('final_output_coronavirus_t60.csv', sep =
 ↪  ",", encoding="utf-8")

#Loading New York Times database as 'df' DataFrame.
```

```
df = pd.read_csv('coronavirus_nytimes_withoutsections_mayupdated.csv', sep = ",", encoding="utf-8")

#Creating a new variable to know the number of each column.
col_mapping = [f"{c[0]}:{c[1]}" for c in enumerate(df1.columns)]

#Creating DataFrame 'df2' with all the columns from the column number 6.
df2 = df1.iloc[:, 6:106]

#Create 'max' column that indicates the topic with the higher probability for each document.
df2['max'] = df2.idxmax(axis=1)

#Creating copy the 'max' column in 'df' DataFrame.
df['max'] = df2['max'].copy()

#Eliminating documents that have the highest probability of topics non-relevant to our analysis.
df = df[df['max'] != 'T0']
df = df[df['max'] != 'T5']
df = df[df['max'] != 'T6']
df = df[df['max'] != 'T7']
df = df[df['max'] != 'T8']
df = df[df['max'] != 'T9']
df = df[df['max'] != 'T11']
df = df[df['max'] != 'T18']
df = df[df['max'] != 'T21']
df = df[df['max'] != 'T22']
df = df[df['max'] != 'T27']
df = df[df['max'] != 'T28']
df = df[df['max'] != 'T34']
df = df[df['max'] != 'T35']
df = df[df['max'] != 'T37']
df = df[df['max'] != 'T43']
df = df[df['max'] != 'T44']
df = df[df['max'] != 'T48']
df = df[df['max'] != 'T57']
```

```python
df = df[df['max'] != 'T59']


#We reset the index of the DataFrame 'df'. We include the date as a 'column' instead of index.
df = df.reset_index()

#Converting the 'speech' column of the 'df' DataFrame to list.
data = df.speech.values.tolist()

#Eliminating non-relevant characters.
data = [re.sub('\S*@\S*\s?', '', sent) for sent in data]
data = [re.sub('\s+', ' ', sent) for sent in data]
data = [re.sub("\'", "", sent) for sent in data]

pprint(data[:1])

#Defining function to pass format from list of stings to list of lists.
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=False))  # deacc=True removes punctuations

#Passing format of 'data' from list of strings to list of lists.
data_words = list(sent_to_words(data ))
print(data_words[:1])

#Constructing the bigram model.
bigram = gensim.models.Phrases(data_words, min_count=5, threshold=50) # higher threshold fewer phrases.
bigram_mod = gensim.models.phrases.Phraser(bigram)

#Definition of the functions for removing stopwords and constructing bigrams.
def remove_stopwords(texts):
    return [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for doc in texts]

#Defining bigram function.
def make_bigrams(texts):
    return [bigram_mod[doc] for doc in texts]

#We remove the stopwords.
data_words_nostops = remove_stopwords(data_words)

#We constuct the bigrams.
data_words_bigrams = make_bigrams(data_words_nostops)

#Passing format of 'data_words_bigrams' from a list of lists to a list of strings.
implodeList = []
for item in data_words_bigrams :
    implodeList.append(' '.join(item))

#Adding as a column the pre-processed minutes in the 'df' dataframe as 'data_words_bigrams'.
df['data_words_bigrams'] = implodeList

#Saving the pre-processed data in txt file.
with open('coronavirus_word2vec_disorder.txt', 'w', encoding = 'utf-8') as f:
    for item in df.data_words_bigrams:
        f.write("%s " % item)

#Saving the preprocessed data without format.
with open('coronavirus_word2vec_order', 'wb') as fp:
    pickle.dump(df.data_words_bigrams, fp, protocol =2)

#Opening the preprocessed data without format.
with open ('coronavirus_word2vec_order', 'rb') as fp:
    df['database'] = pickle.load(fp)

#We save the reduced the pre-processed with bigrams DataFrame 'df' as csv file.
```

```
df.to_csv('nyt_coronavirus_reducedtopicdf.csv', encoding =
 ↪  'utf-8')
#Opening the preprocessed data from txt file.
with open('coronavirus_word2vec_disorder.txt',  encoding =
 ↪  'utf-8') as f:
    tokens_bigrams = f.read().split()

print("raw n. tokens =", len(tokens_bigrams))

#We prepare the dataset for Word2Vec.
#Setting text database in right format.
with open('coronavirus_text_collocations', 'wt') as f:
    f.write(" ".join(tokens_bigrams ))

with open('coronavirus_text_collocations') as f:
    phrases = f.read().split()

HTML(" ".join(tokens_bigrams [:100]))

def text8_to_sentences(tokens):
    """The models insist on sentences; Let's build some."""
    index = 0
    inc = 200

    while index + inc < len(tokens):
        yield tokens[index:index+inc]
        index += inc

    yield tokens[index:]

sentences = list(text8_to_sentences(tokens_bigrams))

#Constuction of Word Embeddings with Word2Vec.
#In Python 3, to make the results reproducible we should
 ↪  set the seed as 'set PYTHONASHSEED=0' in the terminal
 ↪  before opening Python. Then, we should open Python from
 ↪  the terminal
PYTHONHASHSEED=0

```

```python
#Size indicates the window size of the Skip-Gram model, and window is the size of the context words. Set sg = 1 and workers = 1 to be able to reproduce the results.
model = gensim.models.Word2Vec(list(text8_to_sentences(phrases)), sg=1, size=200, window=10, seed=0, workers=1)

print(model==0)
print (list(model.wv.vocab))
print (len(list(model.wv.vocab)))

print(model)

X = model[model.wv.vocab]

#Estimation of clusters of the Word Embeddings with K-Means Clustering.
#Number of clusters.
NUM_CLUSTERS=120

#Setting seed for reproducibility.
rng = random.Random()
rng.seed(0)

#Estimation of K-Means.
kclusterer = KMeansClusterer(NUM_CLUSTERS, distance=nltk.cluster.util.cosine_distance, repeats=25, rng= rng)

assigned_clusters = kclusterer.cluster(X, assign_clusters=True)

words = list(model.wv.vocab)

kmeans = cluster.KMeans(n_clusters=NUM_CLUSTERS)

kmeans.fit(X)

labels = kmeans.labels_
centroids = kmeans.cluster_centers_
```

```

print ("Cluster id labels for inputted data")
#print (labels)
print ("Centroids data")
#print (centroids)
print ("Score (Opposite of the value of X on the K-means
 ↪  objective which is Sum of distances of samples to their
 ↪  closest cluster center):")
#print (kmeans.score(X))


silhouette_score = metrics.silhouette_score(X, labels,
 ↪  metric='euclidean')

print ("Silhouette_score: ")
print (silhouette_score)

cluster_list = pd.DataFrame(
    {'assigned_clusters': assigned_clusters,
     'words': words
    })

ffff

#Printing the number  assigned to the cluster of each word.
print(cluster_list.loc[cluster_list['words'] ==
 ↪  'uncertainty'])
print(cluster_list.loc[cluster_list['words'] ==
 ↪  'uncertain'])
print(cluster_list.loc[cluster_list['words'] == 'fears'])
print(cluster_list.loc[cluster_list['words'] == 'fear'])
print(cluster_list.loc[cluster_list['words'] == 'worries'])

#Saving in DataFrames the words of each cluster.
uncertainty =
 ↪  cluster_list.loc[cluster_list['assigned_clusters'] ==
 ↪  7]
uncertain =
 ↪  cluster_list.loc[cluster_list['assigned_clusters'] ==
 ↪  33]
```

```
fears = cluster_list.loc[cluster_list['assigned_clusters']
  ↪  == 65]
fear = cluster_list.loc[cluster_list['assigned_clusters']
  ↪  == 59]
worries =
  ↪  cluster_list.loc[cluster_list['assigned_clusters'] ==
  ↪  73]

#Saving in an excel file the words of each cluster.
uncertainty.to_excel(
  ↪  'uncertainty_coronavirus_list_words_k120.xlsx')
uncertain.to_excel(
  ↪  'uncertain_coronavirus_list_words_k120.xlsx')
fears.to_excel('fears_coronavirus_list_words_k120.xlsx')
fear.to_excel('fear_coronavirus_list_words_k120.xlsx')
worries.to_excel('worries_coronavirus_list_words_k120.xlsx')
```

The complementary material folder comprises the lists of words of the clusters of 'uncertainty', 'uncertain', 'fear', 'fears' and 'worries'. We also attach the reduced database with bigrams. The following list comprises the documents included in the supplementary material folder.

1. 'uncertain_coronavirus_list_words_k120.xlsx' (List of words of the cluster of the word 'uncertain');

2. 'uncertainty_coronavirus_list_words_k120.xlsx' (List of words of the cluster of the word 'uncertainty');

3. 'fear_coronavirus_list_words_k120.xlsx' (List of words of the cluster of the word 'fear');

4. 'fears_coronavirus_list_words_k120.xlsx' (List of words of the cluster of the word 'fears');

5. 'worries_coronavirus_list_words_k120.xlsx' (List of words of the cluster of the word 'worries');

6. 'nyt_coronavirus_reducedtopicdf.csv' (New York Times reduced database with bigrams).

6.4 Merging Databases and Topic-Uncertainty Indices Graphs

This section shows the python code (‘new york times - uncertainty index.py’) to construct the topic-uncertainty indices. Moreover, we download the financial variables with python from Yahoo Finance and merge them in the same database of the topic-uncertainty indices. We then save this database as ‘coronavirus_garch.xls’. This database is used in the Exponential GARCH computations. Moreover, we create graphs to compare the evolution of the Standard and Poor’s 500 and the topic-uncertainty indices such as Figures 2, 3, 4, and 5 of the paper. The python code is the following:

```
import pandas as pd
import pickle
from pandas_datareader import data

#Packages for times series plot.
import matplotlib.pyplot as plt
from matplotlib import pyplot
import matplotlib.patches as mpatches
from pylab import *
import Pyro4
import seaborn as sns
import dateutil.parser

#Importing list of words databases as DataFrames.
fear =
 ↪  pd.read_excel("fear_coronavirus_list_words_k120.xlsx",
 ↪  sep = ",", encoding="utf-8")
fears =
 ↪  pd.read_excel("fears_coronavirus_list_words_k120.xlsx",
 ↪  sep = ",", encoding="utf-8")
uncertaintyy = pd.read_excel(
 ↪  "uncertainty_coronavirus_list_words_k120.xlsx", sep =
 ↪  ",", encoding="utf-8")
uncertain = pd.read_excel(
 ↪  "uncertain_coronavirus_list_words_k120.xlsx", sep =
 ↪  ",", encoding="utf-8")
worries =
 ↪  pd.read_excel("worries_coronavirus_list_words_k120.xlsx",
 ↪  sep = ",", encoding="utf-8")

```

```
#Merging the DataFrames of the list of words in the DataFrame 'data'.
dictionary1 = pd.concat([fear, fears], axis=0)
dictionary2  = pd.concat([dictionary1, uncertaintyy], axis=0)
dictionary3  = pd.concat([dictionary2, uncertain], axis=0)
daata = pd.concat([dictionary3, worries], axis=0)
daata = daata.reset_index()

#We import the pre-processed database of the New York Times as DataFrame 'df'.
df = pd.read_csv("nyt_coronavirus_reducedtopicdf.csv", sep = ",", encoding="utf-8")

#Importing bigram  reduced database of the New york Times as a column of the 'df' DataFrame.
with open ('coronavirus_word2vec_order', 'rb') as fp:
    df['database_b'] = pickle.load(fp)

########################################
#Counting frequency of words of the lists of uncertain, uncertainty, fear, fears and worries #
########################################

#Passsing to list the column 'words' of the DataFrame 'daata'.
uncer_index = daata['words']
implodeList =list(uncer_index)

#Passing to upper case the 'uncertainty' dictionary.
uncertainty = []
for word in implodeList:
   uncertainty.append(word.upper())
print(uncertainty)

#Incorporate news columns in the 'df' DataFrame to include the count of uncertain and total number of words.
df = pd.concat([df, pd.DataFrame(columns = ['UncerScore']),
                  pd.DataFrame(columns = ['TotalWordCount'])])
```

```python

#Counting the total number of words by article and the number of 'uncertain' words per article.
bow_uncer = []
for i,article in enumerate(df.database_b):
    if str(article) != 'nan':
        m = 0
        for word in article.split(' '):
                if word.upper() in uncertainty:
                    m+= 1
                    bow_uncer.append(word)
        df.UncerScore[i]      = m
        df.TotalWordCount[i] = len(article.split(' '))

###################################
#Creating daily uncertainty index #
###################################

#Creating new DataFrame with the columns 'TotalWordCount', 'UncerScore' and 'the_date'.
df_min = df[['TotalWordCount', 'UncerScore', 'the_date']]

#Creating 'new_date' column with time format.
df_min['new_date'] = pd.to_datetime(df_min['the_date']).copy()

#Grouping the number of the uncertainty words by the column 'new_date'.
df_sum = df_min.groupby(df_min['new_date'])['UncerScore'].agg(['sum']).copy()

#Grouping the total number of words by the column 'new_date'.
df_sum['sum_total'] = df_min.groupby(df_min['new_date'])['TotalWordCount'].agg(['sum']).copy()

#Creating uncertainty score.
df_sum['unc'] = (df_sum['sum'] / df_sum['sum_total'] )

```

```
#Creating normalized uncertainty index.
df_sum['unc_score'] = ( df_sum['unc'] /
 ↪  df_sum['unc'].mean()) *100

###########################################
#Creating daily topic-uncertainty indexes #
###########################################

#Importing LDA output 'topics per day' as DataFrame 'lda'.
lda = pd.read_csv("final_output_agg_coronavirus_t60.csv",
 ↪  sep = ",", encoding="utf-8")

#Creating 'new_date' column with columns 'year', 'month'
 ↪  and 'day'.
lda['new_date'] =
 ↪  pd.to_datetime(lda[['year','month','day']]).copy()

#Setting the 'new_date' column as index of the 'lda'
 ↪  DataFrame.
lda = lda.set_index('new_date')

#Merging DataFrames 'lda' and 'df_sum' in the new DataFrame
 ↪  'mix'.
mix = pd.merge(lda, df_sum, how='left', left_index=True,
 ↪  right_index=True)

#We construct the topic-uncertainty indexes.
mix['brexit'] = mix['T33'] * mix['unc_score']
mix['coronavirus'] = mix['T29'] * mix['unc_score']
mix['economic'] = mix['T3'] * mix['unc_score']
mix['trade_war'] = mix['T51'] * mix['unc_score']
mix['climate_change'] = mix['T54'] * mix['unc_score']

#Constructing mean rolling window for the column
 ↪  'unc_score' and the  'topic-uncertainty' indexes.
mix['rolling_unc_score'] = mix['unc_score'].rolling(9,
 ↪  center = True).mean()
mix['rolling_brexit'] = mix['brexit'].rolling(9,  center =
 ↪  True).mean()
```

```
mix['rolling_coronavirus'] = mix['coronavirus'].rolling(9,
 ↪  center = True).mean()
mix['rolling_economic'] = mix['economic'].rolling(9,
 ↪  center = True).mean()
mix['rolling_trade_war'] = mix['trade_war'].rolling(9,
 ↪  center = True).mean()
mix['rolling_climate_change'] =
 ↪  mix['climate_change'].rolling(9,  center = True).mean()

#####################
#Financial database #
#####################

#We select all available data from 01/01/2019 until
 ↪  01/05/2020.
start_date = '2019-01-01'
end_date = '2020-05-01'

########
#SP500 #

#Downloading from Yahoo finance the variables for the
 ↪  Standard and Poor's 500 index.
sp500 = data.DataReader('^GSPC', 'yahoo', start_date,
 ↪  end_date)

#Creating a new column with the lag value of Standard and
 ↪  Poor's 500 index.
sp500['Lag_Close'] = sp500['Close'].shift(periods=1)

#Creating returns of Standard and Poor's 500.
sp500['close_score'] = ((sp500['Close'] -
 ↪  sp500['Lag_Close']) / sp500['Lag_Close'] ) *100

#Creating rolling window of Standard and Poor's 500.
sp500['rolling_w_close'] = sp500['Close'].rolling(9,
 ↪  center = True).mean()

#########
#Nasdaq #
```

```

#Downloading from Yahoo finance the variables for the
 ↪  Nasdaq index.
nasdaq = data.DataReader('^IXIC', 'yahoo', start_date,
 ↪  end_date)

#Creating new column  with the Nasdaq closing value.
nasdaq['nasdaq_close'] = nasdaq['Close']

#Creating lag value of Nasdaq index.
nasdaq['Lag_nasdaq_close'] =
 ↪  nasdaq['Close'].shift(periods=1)

#Creating returns of Nasdaq index.
nasdaq['nasdaq_close_score'] = ((nasdaq['nasdaq_close'] -
 ↪  nasdaq['Lag_nasdaq_close'] ) /
 ↪  nasdaq['Lag_nasdaq_close'] ) * 100

#Creating rolling window of Nasdaq index.
nasdaq['rolling_nasdaq_close'] =
 ↪  nasdaq['nasdaq_close'].rolling(9,  center =
 ↪  True).mean()

############
#Dow Jones #

#Downloading from Yahoo finance the variables for the Dow
 ↪  Jones index.
dow_jones = data.DataReader('^DJI', 'yahoo', start_date,
 ↪  end_date)

#Creating new column  with the Dow Jones closing value.
dow_jones['dow_close'] = dow_jones['Close']

#Creating lag value of Dow Jones index.
dow_jones['Lag_dow_close'] =
 ↪  dow_jones['Close'].shift(periods=1)

#Creating returns of Dow Jones index.
```

```python
dow_jones['dow_close_score'] = ((dow_jones['Close'] - dow_jones['Lag_dow_close']) / dow_jones['Lag_dow_close']  ) * 100

#Creating rolling window of Dow Jones index.
dow_jones['rolling_dow_close'] = dow_jones['dow_close'].rolling(9,  center = True).mean()

######
#VIX #

#Downloading from Yahoo finance the variables for the VIX index.
vix = data.DataReader('^VIX', 'yahoo', start_date, end_date)

#Creating new column  with the VIX closing value.
vix['vix_close'] = vix['Close']

#Creating lag value of VIX index.
vix['Lag_vix_close'] = vix['Close'].shift(periods=1)

#Creating lag minus mean of the VIX index for GARCH regression.
vix['vix_mean'] =  vix['Lag_vix_close'] - (vix['Close'].mean())

#Creating returns of VIX index.
vix['vix_close_score'] =  (( vix['vix_close']  - vix['Lag_vix_close'] ) / vix['Lag_vix_close'] ) * 100

#Creating rolling window of VIX index.
vix['rolling_vix_close'] = vix['vix_close'].rolling(9, center = True).mean()

##############################
#US 10 years treasury yields #

```

```
#Downloading from Yahoo finance the variables for the US 10
 ↪  years treasury yields.
t10 = data.DataReader('^TNX', 'yahoo', start_date,
 ↪  end_date)

#Creating new column  with the US 10 years treasury yields
 ↪  closing value.
t10['t10_close'] = t10['Close']

#Creating lag value of US 10 years treasury yields.
t10['Lag_t10_close'] = t10['Close'].shift(periods=1)

#Creating returns of US 10 years treasury yields.
t10['t10_close_score'] = ((t10['Close'] -
 ↪  t10['Lag_t10_close']) / t10['Lag_t10_close']  ) * 100

#Creating rolling window of US 10 years treasury yields.
t10['rolling_t10_close'] = t10['t10_close'].rolling(9,
 ↪  center = True).mean()

###############################
#Meging financial DataFrames #
comb1 = pd.merge(dow_jones, nasdaq, how='left',
 ↪  left_index=True, right_index=True)
comb2 = pd.merge(comb1, vix, how='left', left_index=True,
 ↪  right_index=True)
comb3 = pd.merge(comb2, t10, how='left', left_index=True,
 ↪  right_index=True)
finance = pd.merge(comb3, sp500, how='left',
 ↪  left_index=True, right_index=True)
finance['t10_close_score'] =
 ↪  finance['t10_close_score'].fillna(method='ffill')

#Merging 'mix' DataFrame with 'finance' DataFrame.
mixyx = pd.merge(mix, finance, how='left', left_index=True,
 ↪  right_index=True)

#We multiply the topic uncertainty indexes by the
 ↪  difference of the lag and the mean of the VIX index.
mixyx['brexit_vix'] =  mixyx['brexit'] * mixyx['vix_mean']
```

```
mixyx['coronavirus_vix'] =  mixyx['coronavirus'] *
 ↪  mixyx['vix_mean']
mixyx['economic_vix'] = mixyx['economic'] *
 ↪  mixyx['vix_mean']
mixyx['trade_war_vix'] = mixyx['trade_war'] *
 ↪  mixyx['vix_mean']
mixyx['climate_change_vix'] = mixyx['climate_change'] *
 ↪  mixyx['vix_mean']

#We eliminate the observation of second of January.
mixx = mixyx[mixyx.index >=
 ↪  dateutil.parser.parse("2019-01-03")]

#We create DataFrame 'garch' only with the variables for
 ↪  GARCH model.
garch = mixx[['coronavirus','trade_war','climate_change',
 ↪  'brexit','economic',
 ↪  'coronavirus_vix','trade_war_vix','climate_change_vix',
 ↪  'brexit_vix','economic_vix',
 ↪  'vix_close_score','Lag_vix_close', 'close_score',
 ↪  'nasdaq_close_score', 'dow_close_score',
 ↪  't10_close_score']].copy()

#Saving DataFrame 'garch' in csv and excel file for the
 ↪  GARCH estimation.
garch.to_csv('coronavirus_garch.csv')
garch.to_excel('coronavirus_garch.xls')

ffff


########
#Graphs #
########

#####################################################
#Graph coronavirus and trade war topic-uncertainty indexes
 ↪  #
######################################################
sns.set(rc={'figure.figsize':(30, 10)})
```

```

fig, ax = plt.subplots()
fig.subplots_adjust(right=0.7)

mix['coronavirus'].plot(ax=ax, color='orange')
mix['rolling_coronavirus'].plot(ax=ax, color='purple')

mix['trade_war'].plot(ax=ax, color= '#739122')
mix['rolling_trade_war'].plot(ax=ax, color='maroon')


sp500['Close'].plot(ax=ax, color='blue', secondary_y=True)
sp500['rolling_w_close'].plot(ax=ax, color='red',
 ↪  secondary_y=True)

ax.set_ylabel('Topic-uncertainty indexes  ', color=
 ↪  'Orange')
plt.ylabel( "SP500 close index ", color='blue')

ax.set_xlabel('Time')

axvline('2019-05-03', color='red', ls="dotted")
axvline('2019-06-03', color='green', ls="dotted")
axvline('2019-07-26', color='red', ls="dotted")
axvline('2019-08-23', color='green', ls="dotted")
axvline('2019-09-19', color='red', ls="dotted")
axvline('2019-10-02', color='green', ls="dotted")
axvline('2020-01-17', color='red', ls="dotted")
axvline('2020-01-31', color='green', ls="dotted")
axvline('2020-02-19', color='red', ls="dotted")
axvline('2020-03-23', color='green', ls="dotted")
axvline('2020-03-25', color='red', ls="dotted")
axvline('2020-04-3', color='green', ls="dotted")


orange_patch = mpatches.Patch(color='orange',
 ↪  label='\'Coronavirus topic-uncertainty\' index')
green_patch = mpatches.Patch(color='purple', label='Mean 9
 ↪  days rolling window of the  \'coronavirus
 ↪  topic-uncertainty\' index ')
```

```

lime_patch = mpatches.Patch(color='#739122', label='\'Trade war topic-uncertainty\' index')
purple_patch = mpatches.Patch(color='maroon', label='Mean 9 days rolling window of  the \'trade war topic-uncertainty\' index ')

blue_patch = mpatches.Patch(color='blue', label='SP500 close index ')
red_patch = mpatches.Patch(color='red', label='Mean 9 days rolling window of the SP500 close index ')

plt.legend(handles=[orange_patch, green_patch, lime_patch, purple_patch, blue_patch, red_patch],loc='center left', bbox_to_anchor=(0, 0.89))

plt.savefig('Graph2_LDA_NYTimes_coronavirus_uncertainty_tradewar.png', bbox_inches='tight')


###############################
#Graph Skip-Gram uncertainty index #
###############################
sns.set(rc={'figure.figsize':(30, 10)})

fig, ax = plt.subplots()
fig.subplots_adjust(right=0.7)

mix['unc_score'].plot(ax=ax, color='orange')
mix['rolling_unc_score'].plot(ax=ax, color='green')

sp500['Close'].plot(ax=ax, color='blue', secondary_y=True)
sp500['rolling_w_close'].plot(ax=ax, color='red', secondary_y=True)

ax.set_ylabel('\'Skip-Gram uncertainty\' index  ', color='Orange')
plt.ylabel( "SP500 close index ", color='blue')
ax.set_xlabel('Time')

```

```python
axvline('2019-05-03', color='red', ls="dotted")
axvline('2019-06-03', color='green', ls="dotted")
axvline('2019-07-26', color='red', ls="dotted")
axvline('2019-08-23', color='green', ls="dotted")
axvline('2019-09-19', color='red', ls="dotted")
axvline('2019-10-02', color='green', ls="dotted")
axvline('2020-01-17', color='red', ls="dotted")
axvline('2020-01-31', color='green', ls="dotted")
axvline('2020-02-19', color='red', ls="dotted")
axvline('2020-03-23', color='green', ls="dotted")
axvline('2020-03-25', color='red', ls="dotted")
axvline('2020-04-3', color='green', ls="dotted")

orange_patch = mpatches.Patch(color='orange', label='\'Skip-Gram uncertainty\' index')
green_patch = mpatches.Patch(color='green', label='Mean 9 days rolling window of the  \'Skip-Gram uncertainty\' index ')

blue_patch = mpatches.Patch(color='blue', label='SP500 close index ')
red_patch = mpatches.Patch(color='red', label='Mean 9 days rolling window of the SP500 close index ')

plt.legend(handles=[orange_patch, green_patch, blue_patch, red_patch],loc='center left', bbox_to_anchor=(0, 0.89))

plt.savefig('Graph3_skipgram_NYTimes_uncertaintyindex.png', bbox_inches='tight')

##################################
#Graph brexit topic-uncertainty index #
#################################
sns.set(rc={'figure.figsize':(30, 10)})

fig, ax = plt.subplots()
fig.subplots_adjust(right=0.7)

mix['brexit'].plot(ax=ax, color='orange')
mix['rolling_brexit'].plot(ax=ax, color='purple')
```

```

sp500['Close'].plot(ax=ax, color='blue', secondary_y=True)
sp500['rolling_w_close'].plot(ax=ax, color='red',
 ↪  secondary_y=True)

ax.set_ylabel('Topic-uncertainty indexes  ', color=
 ↪  'Orange')
plt.ylabel( "SP500 close index ", color='blue')
ax.set_xlabel('Time')


axvline('2019-05-03', color='red', ls="dotted")
axvline('2019-06-03', color='green', ls="dotted")
axvline('2019-07-26', color='red', ls="dotted")
axvline('2019-08-23', color='green', ls="dotted")
axvline('2019-09-19', color='red', ls="dotted")
axvline('2019-10-02', color='green', ls="dotted")
axvline('2020-01-17', color='red', ls="dotted")
axvline('2020-01-31', color='green', ls="dotted")
axvline('2020-02-19', color='red', ls="dotted")
axvline('2020-03-23', color='green', ls="dotted")
axvline('2020-03-25', color='red', ls="dotted")
axvline('2020-04-3', color='green', ls="dotted")


orange_patch = mpatches.Patch(color='orange',
 ↪  label='\'Brexit change topic-uncertainty\' index')
green_patch = mpatches.Patch(color='purple', label='Mean 9
 ↪  days rolling window of the  \'brexit
 ↪  topic-uncertainty\' index ')

blue_patch = mpatches.Patch(color='blue', label='SP500
 ↪  close index ')
red_patch = mpatches.Patch(color='red', label='Mean 9 days
 ↪  rolling window of the SP500 close index ')

plt.legend(handles=[orange_patch, green_patch, blue_patch,
 ↪  red_patch],loc='center left', bbox_to_anchor=(0.0,
 ↪  0.89))

```

```
plt.savefig('Graph4_LDA_NYTimes_brexit.png', bbox_inches='tight')


##############################################
#Graph economic-Fed topic-uncertainty index #
#############################################
sns.set(rc={'figure.figsize':(30, 10)})

fig, ax = plt.subplots()
fig.subplots_adjust(right=0.7)

mix['economic'].plot(ax=ax, color='orange')
mix['rolling_economic'].plot(ax=ax, color='purple')

sp500['Close'].plot(ax=ax, color='blue', secondary_y=True)
sp500['rolling_w_close'].plot(ax=ax, color='red', secondary_y=True)

ax.set_ylabel('Topic-uncertainty indexes  ', color='Orange')
plt.ylabel( "SP500 close index ", color='blue')
ax.set_xlabel('Time')


axvline('2019-05-03', color='red', ls="dotted")
axvline('2019-06-03', color='green', ls="dotted")
axvline('2019-07-26', color='red', ls="dotted")
axvline('2019-08-23', color='green', ls="dotted")
axvline('2019-09-19', color='red', ls="dotted")
axvline('2019-10-02', color='green', ls="dotted")
axvline('2020-01-17', color='red', ls="dotted")
axvline('2020-01-31', color='green', ls="dotted")
axvline('2020-02-19', color='red', ls="dotted")
axvline('2020-03-23', color='green', ls="dotted")
axvline('2020-03-25', color='red', ls="dotted")
axvline('2020-04-3', color='green', ls="dotted")

```

```
orange_patch = mpatches.Patch(color='orange', label='\'Economic-Fed change topic-uncertainty\' index')
green_patch = mpatches.Patch(color='purple', label='Mean 9 days rolling window of the  \'economic-Fed topic-uncertainty\' index ')

blue_patch = mpatches.Patch(color='blue', label='SP500 close index ')
red_patch = mpatches.Patch(color='red', label='Mean 9 days rolling window of the SP500 close index ')

plt.legend(handles=[orange_patch, green_patch, blue_patch, red_patch],loc='center left', bbox_to_anchor=(0.05, 0.89))

plt.savefig('Graph4_LDA_NYTimes_economic.png', bbox_inches='tight')


#############################################
#Graph climate topic-uncertainty index #
############################################
sns.set(rc={'figure.figsize':(30, 10)})

fig, ax = plt.subplots()
fig.subplots_adjust(right=0.7)

mix['climate_change'].plot(ax=ax, color='orange')
mix['rolling_climate_change'].plot(ax=ax, color='purple')

sp500['Close'].plot(ax=ax, color='blue', secondary_y=True)
sp500['rolling_w_close'].plot(ax=ax, color='red', secondary_y=True)

ax.set_ylabel('Topic-uncertainty indexes  ', color= 'Orange')
plt.ylabel( "SP500 close index ", color='blue')
ax.set_xlabel('Time')

```

```
axvline('2019-05-03', color='red', ls="dotted")
axvline('2019-06-03', color='green', ls="dotted")
axvline('2019-07-26', color='red', ls="dotted")
axvline('2019-08-23', color='green', ls="dotted")
axvline('2019-09-19', color='red', ls="dotted")
axvline('2019-10-02', color='green', ls="dotted")
axvline('2020-01-17', color='red', ls="dotted")
axvline('2020-01-31', color='green', ls="dotted")
axvline('2020-02-19', color='red', ls="dotted")
axvline('2020-03-23', color='green', ls="dotted")
axvline('2020-03-25', color='red', ls="dotted")
axvline('2020-04-3', color='green', ls="dotted")

orange_patch = mpatches.Patch(color='orange',
 ↪  label='\'Climate change topic-uncertainty\' index')
green_patch = mpatches.Patch(color='purple', label='Mean 9
 ↪  days rolling window of the  \'climate change
 ↪  topic-uncertainty\' index ')

blue_patch = mpatches.Patch(color='blue', label='SP500
 ↪  close index ')
red_patch = mpatches.Patch(color='red', label='Mean 9 days
 ↪  rolling window of the SP500 close index ')

plt.legend(handles=[orange_patch, green_patch, blue_patch,
 ↪  red_patch],loc='center left', bbox_to_anchor=(0.0,
 ↪  0.89))

plt.savefig('Graph4_LDA_NYTimes_climate.png',
 ↪  bbox_inches='tight')
```

6.5 EGARCH: Estimation and Measures of Goodness of Fit

This sections shows part of the Rstudio code to estimate the Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) to analyze the effect of an increase in the topic-uncertainty indices in US financial markets from 8 January, 2019 to 1 May, 2020.

In particular, the following lines show the code of the measures of the ‘trade war’ uncertainty index for the first specification of the EGARCH model (Equations 5 and 6, and Table 9 of the paper).

```
 fit.spec <- ugarchspec(variance.model      = list(model =
  ↪  "eGARCH", garchOrder = c(1, 1) , external.regressors =
  ↪  tradewar_vix_num), mean.model = list( armaOrder = c(1,
  ↪  1), include.mean = TRUE , external.regressors =
  ↪  tradewar_vix_num), distribution.model = "norm")

fit  <- ugarchfit( spec = fit.spec ,  spx_num, mexsimdata=
 ↪  tradewar_vix_num , vexsimdata= tradewar_vix_num ,
 ↪  solver = "hybrid")
fit
```