READ ME FILE

Title: An AI-powered Tool for Central Bank Business Liaisons: Quantitative Indicators and On-demand Insights from Firms

Authors: Nicholas Gray, Finn Lattimore, Kate McLoughlin and Callan Windsor

Description

This 'read me' file contains general instructions on how to replicate the tools and results presented in RDP 2025-06. The code and data are structured for two main uses:

1. To support the creation of other textual analysis and retrieval systems (TARS), using artificial data as a placeholder to preserve the confidential nature of unaggregated liaison textual information.
2. To support replication of the paper's findings, including the construction of text-based indicators and their application in nowcasting exercises.

Coding languages

- Python: If you do not have Python installed, a fully working free scientific distribution, which includes all of the necessary packages, can be found here: https://www.anaconda.com/download/. The code has been written in Python 3.10. Required packages are listed in 'environment.yml' file in the 'backend' folder.
- R: If you do not have R installed, a fully working free scientific distribution, which includes all of the necessary packages, can be found here: https://cran.r-project.org/. The code has been written in R 4.4.0. Required packages are listed in requirements.txt files in the folders.
- The program 'RStudio' is needed to open and use the '.Rproj' file in each of the folders. In our work we used version 2024.04.0. For more information, see <https://www.posit.co/>.

Questions, comments and bug reports can be sent to windsorc and grayn at domain rba.gov.au.

Quick guide

- Download the zipped file 'rdp-2025-06-supplementary-information'.
- Unpack into desired directory.
- Ensure Python and R environments are set up using the provided environment files.
- Refer to the 'README' files in each folder for detailed instructions on running specific components.

Zip file contents

- 'rdp-2025-06-read-me.pdf' is this read me file.
- 'Data' contains all input data files used across each of the folders.
- 'backend' is a folder containing code for replicating a data extraction process like the tool described in the paper.
- 'frontend' is a folder containing code to create a dashboard like the tool described in the paper.
- 'Capabilities' is a folder containing code for building the text-based measure described in the paper.
- 'nowcasting' is a folder containing code for running the empirical exercise outline in Section 5 of the paper.

Data

This folder contains the data that is used by default in each of the other folders of the supplementary information. All the data for demonstrating how to build and use a tool as described in the paper has been artificially generated. As a result, using this data for replication will look different to results shown in the paper. This is to ensure the confidential nature of liaison discussions is maintained, whilst allowing for exploration of a tool that would use this confidential information within the RBA. The data for running the nowcasting exercise is real aggregate data and is similar to what was used in the results in the paper.

- 'rdp-2025-06-graph-data.xlsx' provides the data used to plot figures in the main paper in an excel format.
- 'Example Liaison Summary Note.docx': a Word document that provides input for the extraction step in the `backend` code. This document is artificially generated and does not contain any confidential information about real firms.
- 'Example_liaison_data.csv': provides input for the extraction step in the 'backend' code. This data is artificially generated and does not contain any confidential information about real firms.
- 'nowcasting_df.xlsx': provides aggregated time series data for nowcasting wage price index (WPI) growth. This is the same liaison data as the results in the paper, except for the gap measures that use confidential model estimates of the NAIRU and NAIRLU. Instead, piecewise representations of NAIRU and NAIRLU are provided that broadly reflect the real model estimates used in the paper.
- 'dictionary.xlsx': provides a list of keywords and their associated qualifiers in an excel format. These keywords underpin the construction of textual measures of topic exposure and tone. There are lists of keywords for the wages and labour topics, split into different tabs based on whether it is counting just topic words, or topic words with positive/negative qualifiers.
- 'LM_dictionary.xlsx': provides a guide for selecting language model outputs and setting thresholds to construct textual measure of topic exposure and tone in an excel format.
- 'uncertainty_dict.xlsx': provides a list of keywords that underpin the construction of the uncertainty measure in an excel format. This is the same set of keywords used to create the measure shown in Figure 8 of the paper.
- 'liaison.sqlite': provides an SQLite database which is queried in the code to extract liaison-like text data for app demo and index construction. This data is artificially generated and does not contain any confidential information about real firms. As a result, using this data for replication will look different to results shown in the paper.

backend

This folder contains code for building the Text Analysis and Retrieval System (TARS) database using artificial data reflective of real liaison information as a placeholder for the confidential liaison data used in the paper.

- 'README': the read me file for this folder and contains further details on replicating the backend process of building a TARS.
- 'TARS_Extraction.ipynb': a python notebook that demonstrates text extraction from a Word document structured like a typical liaison summary document discussed in Section 3 of the paper.
- 'TARS_Enrichment.ipynb': a python notebook that demonstrates text enrichment using a suite of language models. This includes tagging text with a set of pre-defined topics, sentiment and precise extractions of numerical quantities.
- 'TARSml.py': contain functions that leverage the language models pulled from the public Hugging Face repository.
- 'TARSutils.py': contains functions used to extract information from word documents, and to support the enrichment of text using language models.
- 'environment.yml': is a YAML file for setting up the Python environment.

frontend

This folder contains code to build the R Shiny dashboard application that uses the TARS database using artificial data reflective of real liaison information as a placeholder for the confidential liaison data used in the paper.

- ‘README’: is the read me file for this folder and contains further details on running the dashboard.
- ‘frontend.Rproj’: is an R project file. When opened, it will initiate an R workspace that allows for easier replication of code.
- ‘app.R’: this is the main file to run the dashboard.
- ‘Code’: is a subfolder that contains UI, server logic, user-defined functions, and word2vec model to run the dashboard. For more details on these files, refer to the ‘README’ file mentioned above.
- ‘markdown’: is within the ‘Code’ subfolder and contains R Markdown and PNG files for generating the landing page of the dashboard. For more details on these files, refer to the ‘README’ file mentioned above.
- ‘requirements.txt’: a dependency file that outlines which R packages (and their versions) are required for a working replication environment.

Capabilities

Contains code for constructing text-based indicators using a TARS that contains artificial data reflective of real liaison information as a placeholder for the confidential liaison data used in the paper.

- ‘README’: is the read me file for this folder and contains further details on building text-based measures outlined in the paper.
- ‘Capabilities.Rproj’: is an R project file. When opened, it will initiate an R workspace that allows for easier replication of code.
- ‘Dictionary_based_indices.R’: code that uses a dictionary of words to build textual measures of topic exposure and tone. This code was applied to the confidential liaison data to build the dictionary-based measures of wages topic exposure and tone shown in Figure 7 of the paper.
- ‘LM_based_indices.R’: code that uses the output of a language model to build textual measures of topic exposure and tone. This code was applied to the confidential liaison data to build the dictionary-based measures of wages topic exposure and tone shown in Figure 7 of the paper.
- ‘Uncertainty_index.R’: code that uses a dictionary of words to build textual measures of firm uncertainty. This code was applied to the confidential liaison data to build the uncertainty measure shown in Figure 8 of the paper.
- ‘Numerical_extraction_indices.R’: code that uses extracted numerical quantities to create aggregate series that can reflect economic measures. This version of the code specifically extracts aggregate measure of wages and price inflation and was used on confidential liaison data to build the numerical extraction of price inflation shown in Figure 9 of the paper.
- ‘Plot_measures.R’: contains code to create Figures 1, 4, 7–9 of the paper, using the graph data file in the ‘Data’ folder.
- ‘data_gen_utils.R’: contains utility function which are sourced and used in the above code. This is done automatically in each of the above scripts.
- ‘requirements.txt’: a dependency file that outlines which R packages (and their versions) are required for a working replication environment.

nowcasting

This folder contains code for the empirical nowcasting exercise using the same liaison data as the results in the paper, except for the gap measures that use confidential model estimates of the NAIRU and NAIRLU. As a result, using this data for replication may be slightly different to results shown in the paper.

- 'README': is the read me file for this folder and contains further details on running the nowcasting exercise.
- 'nowcasting.Rproj': is an R project file. When opened, it will initiate an R workspace that allows for easier replication of code.
- 'fit_predict.R': main script for running nowcasting. In this script, the data will be imported and split up into each of the nowcasting windows. Then each of these windows will be run through OLS and regularised regressions to build models that are used to make out-of-sample predictions for each window. Finally, each model's out-of-predictions are compared to the real values and the errors are compared to a baseline Phillips curve OLS specification. The output is the RMSE, and significance compared to the baseline, as well as each of the model files.
- 'Functions.R': utility functions sourced and used by the scripts to assist with building the data windows and regression models, as well as functions for analysing the results.
- 'analysis.R': script for evaluating and visualising nowcasting performance. This script takes the outputs from 'fit_predict.R' and the graph data file in the 'Data' folder as its input. This script will allow you to plot Figures 14, 15 and D1.
- 'requirements.txt' is a dependency file that outlines which R packages (and their versions) are required for a working replication environment.

28 August 2025