CESSA WP 2025-02

社会経済研究のためのテキスト分析:
ロシア中央銀行通貨政策ガイドラインを例として

岡山 香
中村 靖

横浜国立大学

2025 年 7 月

Center for Economic and Social Studies in Asia (CESSA) Working Paper
Downloadable from:
https://www.econ.ynu.ac.jp/cessa/publication/workingpaper/

Center for Economic and Social Studies in Asia, Department of Economics
Yokohama National University

社会経済研究のためのテキスト分析:
ロシア中央銀行通貨政策ガイドラインを例として

Text Mining as a Tool to Gather Information for Socio-Economic Studies: A Text Analysis of Central Bank of Russia Monetary Policy Guidelines

岡山　香（横浜国立大学経済学部附属アジア経済社会研究センター助手）
中村　靖（横浜国立大学名誉教授）

Kaori OKAYAMA
Yokohama National University, Collage of Economics,
Center for Economic and Social Studies in Asia, Research Associate
Yasushi NAKAMURA
Yokohama National University, Emeritus Professor

2025 年 7 月

Abstract

社会経済研究情報をテキスト分析で得ることの有効性と問題点を評価するために，ロシア中央銀行が毎年発行する通貨政策ガイドラインを分析した．有益な情報を得ることができた一方，本格的分析には専門分野と対象文献に合致したテキスト分析用の各種辞書の整備が必要であることがわかった．対象となる専門文献と専門分野について知識を持つ専門家との協力のもとでテキスト分析経験を蓄積することが重要である．

We text-mined Monetary Policy Guideline documents issued annually by the Central Bank of Russia to evaluate the effectiveness and problems of using text analysis methods to obtain information for socio-economic studies. While we were able to obtain useful information, we found that it is a prerequisite to have dictionaries for text analysis that match the specialized field and the target literature. For making those dictionaries, it is important to accumulate experience in text analysis in cooperation with experts who have knowledge of the target literature and the specialized field.

Key words

Text mining, tidytext, Text analysis, Central Bank of Russia, Monetary Policy,

1. はじめに

経済活動全般のグローバル化の進展と世界情勢の予測困難な変化は，政治，社会，文化，国際関係，環境といった広範な要因を考慮した総合的アプローチによる経済分析の必要性を高めている(Graebner-Radkowitsch and Strunk, 2023)．総合的アプローチによる分析は，アカデミックな経済分析だけでなく，企業経営戦略の策定や政府の政策立案といった実践的分野において必要となっている(PWC, 2022).

横浜国立大学経済学部附属アジア経済社会研究センター(CESSA)は，貿易人材育成を使命のひとつとした横浜高等商業学校の伝統を引き継ぎ，世界各国，各地域の社会経済分析のための情報の収集，蓄積と研究支援をおこなってきた．しかし，情報のデジタル化とその伝達のオンライン化にともない，紙媒体を中心としてきた CESSA の研究支援サービスは時代遅れになったといわざるをえない．この間，CESSA はデジタル化，オンライン化に対応した新たな研究支援サービスの構築を検討してきた．

現在検討中の研究支援サービスのひとつが，研究者からのセミカスタムオーダーによるテキスト分析サービスの提供である．Web ページ，SNS 等により大量に供給されるデジタル文書データは，総合的アプローチによる経済分析の情報源として大きなポテンシャルを持っている．しかし，大量で多様なデジタル文書データをどのように収集，蓄積，利用するかは，容易に解決できる問題ではない．CESSA の限られた資源では，Internet Archive のように，デジタル情報を row data として網羅的に保存することはできない．そこで CESSA では，研究者側からの提案にもとづき特定テーマのデジタル文書データを一定期間にわたり収集，加工し，加工データを研究者に提供するとともに，データを蓄積するというサービスの提供を企画している．このサービス提供に向けて，CESSA は，デジタルリポジトリの構築，デジタル文書データの自動収集，テキスト分析といった要素技術について試験実装し，検討をおこなってきた．本研究は，それらの検討結果のうち，社会経済分析の専門分野のデジタル文書に定型的なテキスト分析を適用する場合の問題点と定型的なテキスト分析の結果がどの程度有益な研究資料となり得るのかを検討した結果を記録することを，目的としている．

分析対象として，ロシア中央銀行が毎年発行している「ロシア中央銀行通貨政策ガイドライン」英語版(MPG, 2009-2024)の 2010 年版から 2025 年版の 16 年分の pdf ファイルを用いている(以下，MPG)．ロシアの政策文書を例としている理由は，第 1 に，SNS や一般ニュースサイトからのデジタル文書ではなく，専門分野のデジタル文書をテキスト分析する場合の問題点の検討を目的としているためである．第2に，ウクライナ戦争勃発以降，通常チャンネルによるロシア公式情報入手が困難になっているため，政府機関等の Web サイトのスクレイピングにより情報を収集したいという研究者側からの要望があったためである．しかし，ロシアの政府諸機関 Web サイトへの接続は現時点では意図的停止や TLS 証明書失効に関する問題があるため，一般的利用例としては適切ではない．その中で，ロシア中央銀行の Web サイトは通常状態で利用可能だった．実際には，ロシアの Web サイトを含む Web サイトのスクレイピングや Web Application Programming Interface (API)を利用したデジタル文書の収集も試行的におこなっているが，これらについては別途まとめて別の研究ノートで報告する予定である．

CESSA の研究支援サービスは，いわゆる Global South への関心の高まりにあわせて，英語以外に，中国語，ポルトガル語，ロシア語，フランス語のデジタル文書への対応を予定し，さらに他のアジア各国言語への拡張も視野に入れている．これらの言語の社会経済分析用専門文献のテキスト分析

には，ドメイン知識を豊富に持つ研究者との密接な協力の下で，テキスト分析で用いる各種辞書を言語ごとに作成することが必要になる．本研究では英語文書に対するテキスト分析のみを扱い，多言語への拡張は今後の課題とする．本研究は，英語の場合でも，社会経済分析用専門文献のテキスト分析ではドメイン知識を持つ研究者との共同作業が不可欠であることを明確に示した．これは，本研究から得られた成果のひとつである．

以下では，tidytext パッケージを使った R の標準的なテキスト分析手法(Silge and Robinson, 2025)により MPG を分析する．本研究の以下の構成は次の通りである．次章では，文書ファイルを読み込み，トークンに分割し，tidytext によるテキスト分析に適した tidydata データ形式で保存する．続いて，頻度分析，TF-IDF 分析，単語間関係のグラフ分析の順に分析をすすめる．それぞれの分析においては，分析用コードの説明ではなく，分析手法を MPG に適用する際に実際に生じた方法的困難についての説明に重点をおく．最後に，MPG のテキスト分析から得られた経済社会分析に関する専門的文書のテキスト分析において今後検討すべき課題をまとめる．

本研究は R Notebook で作成されている．Markdown ファイルは MPG データとともに Appendix の URL からダウンロードできる．分析に利用した R のバージョンは R 4.5.0(R Core Team, 2025)である．以下の背景が灰色で字体が Consolas の部分は R のコードである．コード内では，特に利用が一般的ではないパッケージの関数の場合，何のパッケージの関数なのかを示すため，パッケージ名: パッケージの関数名の表記法を使っている．ただし，通常は，パッケージ名を含めて関数を指定する必要はないため，パッケージ名を含む表記方法を網羅的には使っていない．また，紙面の制約のため，印刷版のコードの書き方は必ずしもベスト・プラクティスにはしたがっていない．

2. MPG テキストの読み込みと tidytext 分析用 tibble 化

2.1 MPG のテキストデータ化

MPG は pdf 形式で公開されている．Acrobat で pdf 形式の MPG を読み込み，UTF-8 テキスト形式で export することで，図表，脚注，ページ番号等の文書 object は排除され，囲み記事(Box，Appendix)等にあるテキストデータもテキストのみ抽出された．他のケースではテキストデータ化に問題が生じる場合もあるが，それらの問題については別稿で報告する．

MPG のテキストデータ化の内容上の問題は，Appendix，囲み記事(Box)，脚注，図表のタイトル，図表注，図表出所などの本文以外の部分の文書情報を分析対象にいれるかどうかの判断だった．特に MPG の場合は，2010 年版から 2025 年版の間に文章のボリュームだけでなく，編集スタイルにも変遷があるため，判断は単純にはできなかった．2014 年版から主として統計表を掲載する目的で Appendix がつけられた．2017 年版からは，Appendix に加えて，Glossary と Abbreviations が掲載されている．2018 年版からは，統計表は Appendix の一部となり，ビジネスサーベイ，経済予測，その時々のトピックの解説が Appendix の主要な内容となった．2018 年版からは，MPG 全体のページ数が 130 ページを超える分量となったが，半分以上のページ数が Appendix にあてられている．これらに加えて Glossary と Abbreviations が掲載されている．2018 年以降は Appendix のかなりの部分が MPG テキスト分析の分析対象として意味のある内容になったといえる．

このような編集スタイル上の変遷がある中で，何をテキスト分析の対象として設定するかを一般的に決めることは難しい．今回は，試行分析として，十分な根拠はないままで，脚注，図表のタイトル，図表注，図表出所，Glossary，Abbreviations はテキスト分析の対象には入れなかった．一方，

Appendix，Box はそのときどきの経済情勢，通貨政策にかかわるテキスト情報を含むと判断し，分析対象に入れることとした．なお Appendix の統計表部分は，3 節でみる Stop word 排除の際に削除され，テキスト情報だけが残る．ただし，ロシア中央銀行定例理事会の日程等を含む，どちらかといえば形式的な内容である通貨政策イベントのクロノロジー等は分析対象として残っている．

2.2 tidy data の基本

本研究では，テキスト分析用のデータを tibble 形式で保存する．tibble 形式は，1 行を 1 レコードとし，列に属性情報を格納する表形式(データフレーム)の一種である．MPG のテキストファイルを読み込み，テキストを，テキストを構成する基本単位であるトークンに分解し，そのデータを tidy data の考え方にしたがって整形し，tibble 形式で保存する．tibble 形式で保存することで，tidyverse パッケージに含まれる各種サブパッケージおよび他のパッケージを使ったテキスト分析が容易におこなえる．tidyverse は，tidy data の考え方にもとづいて作られたデータを処理するためのパッケージ集である(Wickham, Çetinkaya-Rundel and Grolemund, 2025)．tidy data の考え方の基本は，tibble の各行が 1 つの observation を記録するものとし，その行の列(属性)のうち 1 列のみが observation の観測値を持つ構成とすることにある(tidyr, 2025)．ある 1 行の値列以外の列は，観測値を特定するための属性となる．例えば，1 つの行の列が日付，観測地，観測対象，気温であり，2025 年 6 月 23 日，横浜市，最高気温，32 度という 1 行(観測)があるとすれば，日付，観測地，観測対象の各列の値を 2025 年 6 月 23 日，横浜市，最高気温と指定すると tibble の値列の値(最高気温)を一意に特定でき，32 度とわかる．逆にみれば，この tibble の中には，値が 32 度以外の値，例えば 20 度であって，同時に日付が 2025 年 6 月 23 日，観測地が横浜，観測対象が最高気温の行は存在しない．

tidy data の考え方の 1 行を 1 つの observation とするという意味は，値列以外の属性列が全体としてリレーショナル・データベースにおける複合主キーとなることを意味していると思われる．しかし，tidy data は，値以外の属性列のセットが値を一意に指定するために必要で最小の属性のセット(正規化終了後の複合主キー)となっていることを要求してはいない(tidyr, 2025)．tidy data の目的はデータベースの構築ではなくデータ分析であるから，違いが生じるのは当然である．例えば，観測地列と観測地 ID 列が存在する場合である．この場合，観測地名と観測地 ID が 1 対 1 に対応しているかぎり，値を一意に指定するために観測地名列と観測地 ID 列のどちらかがあれば良く，属性列セットは値を一意に特定するための「最小の数」の列とはならない．このようなデータの構造は，実際の観測データとしてはよくあるだろう．特にデータを可視化する場合には，値列が複数あるようなデータ構造の方が都合の良い場合もある．上の例では，観測日列が 1 つあるのではなく，6 月 23 日，6 月 24 日，6 月 25 日というように日付の列が横長に並んでいる構造の方が時系列変化を可視化する場合には扱いやすいことがある．ただし，日付の列が複数ある場合は，取り出したい値の列名(日付)がわからなければ値を一意に取り出すことはできない．tidyverse パッケージを構成する tidyr パッケージは，観測値が複数ある構造を単一の値に変換する pivot_longer()関数やその逆をおこなう pivot_wider()関数といった関数を用意しており，データ構造の変換は容易である．ただし，基本形は tidy data であることと，現在取り扱っているデータ構造が tidy data の原則に従っているかを意識することが重要である．

2.3 テキストファイルの読み込み

MPG を各年で 1 つの UTF-8 エンコード txt ファイルとして保存する．発行年ごとの 1 つの pdf ファイルを 1 つの txt ファイルに変換することが簡単であるだけでなく，以下のコードでは tibble 化する

際にファイル名情報を保持するようにしているためである．ファイル名を適切に設定すれば，MPG の各版の情報を tibble の属性として自動的に格納できる．ここでは，ファイル名に発行年を入れ，それを tibble の issue 列に保持している． tibble の名前は tbl としている．

まず，tidytext パッケージの read_lines()関数を用いて，MPG テキストファイルの文字(MPG の場合基本的にアルファベットとその他の記号)を改行コードがあらわれるまで読み込む．読み込んだ文字全体を paste()関数で 1 つの文字列に結合する．これで 1 段落分が 1 つの文字列になる．この文字列(1 段落)を tibble の text の列に格納する．発行年情報は MPG のテキスト分析に重要な情報であるため，ファイル名を利用して issue 列に格納する．この処理により，各行が段落内容とその段落が掲載されていた MPG の版についての情報を保持している tibble が作成される．

以下のコードでは，tidyverse パッケージ(purrr サブパッケージ)の map_df()関数を用いて以上の処理をおこなっている．map_df()関数は，map()関数の戻り値をデータフレーム(tibble)形式に変える map()関数の wrapper 関数である．map()関数は第 1 引数で与えられたベクトル，配列，データフレーム等の要素ごとに第2引数で与えられた関数を並列で適用する関数である．ループを使うことで同等の結果を得ることができるが，R や python といったインタプリタ型の言語では並列処理する方が，処理速度が上がる．加えて，ループ制御を避けることはコードの書き間違えの減少につながる．

ここでは，map_df()関数中でデータに適用する一連の処理を R のラムダ式(無名関数)で指定している．ラムダ式は，処理したい内容を自身で定義するが，特に繰り返し利用を想定しないため関数名を与えず(無名)，独自の関数として保存する必要のない関数である．ラムダ式中の.x はラムダ式が受け取る引数を示している．map()関数が map()関数の第 1 引数をラムダ式に渡している．ラムダ式が受け取る引数が 2 つ以上の場合も，.y，.z 等で指定できる．以下のコードでは，tidyverse のパイプ演算子%>%が list.files()関数で読み込んだ MPG のファイルの 1 つを map_df()関数の第 1 引数として引き渡すので，このコードでは map_df()関数では第 1 引数が明示的に示されておらず，ラムダ式の定義に使われる.x だけがラムダ式の中にあらわれている．同じ内容を list.files()の結果作成されるファイルのパスのリストを list_f とし，パイプ演算子を使わなければ，

```
map_df(list_f, ~ tibble(text=readLines(.x), …)
issue=tools::file_path_sans_ext(basename(.x))))
```

となる．

MPG ファイル内の段落位置情報を tbl に保存するために，段落番号を格納した line_num 列を tbl に加える．オリジナルの MPG は段落番号を持っていないため，tbl の行番号で段落番号を代替する．tbl への段落の読み込みは各 issue の先頭から順番におこなわれるため，tbl の行番号はオリジナルの MPG 文書内の段落の順番を反映している．行番号を issue ごとに 1 から振りなおすために，group_by()関数を issue 列に適用し，各年次 MPG で tbl の行をグループ化したうえで，row_number()関数を用いて行番号をグループごとに取得し，これを段落番号として line_num 列に格納する．group_by()関数でグループ化すると，tibble に適用する関数はデフォルトで group ごとに別個に適用されるため，ここでは MPG の各 issue に別個に row_number()関数を適用することとなり，各 issue 行番号は自動的に 1 から振られる．group_by()関数でグループ化する目的は段落番号を振りなおすためだけであるから，処理後は ungroup()関数でグループ化を解除しておく．グループ化の解除を忘れると，tbl に対する以降の各種処理は基本的にグループごとに独立して適用されることになる．mutate()関数は段落番号を格納

した line_num 列を tbl に新たに付加している．tibble 形式データは既存データの保持を優先的目的としているから，tibble 内の値の操作には制約がある．一方，新しい情報で新しい列を追加すること，既存列を処理して新しい列を追加することは，mutate()関数で容易におこなえる．

ファイル名(ここでは出版年)や段落番号だけでなく，章見出しやページ数を属性として分析用 tibble データに保持したい場合があるかもしれない．オリジナルのテキストにおいて何らかのマーカにより章見出しやページ数が容易に特定できるならば，tibble 化の際にも，tibble 化後にも，それらの文書情報を付与することは容易である．例えば，章見出しに”Ch.”がついていれば，“Ch.”をマーカとして章見出しを発見し，付加するだけである．章見出しやページ数が容易に特定できる状態でなければ，手動で対処するしかない．MPG ではページ数や章見出しのスタイルが一定していないことと，ページ数，章見出し等の情報が MPG のテキスト分析において特別の意味があるとは考えられないことから，それらの情報を保存する処理はしなかった．一方，章見出しは tbl の 1 行であり，テキスト分析の対象になっている．Table 1 は tbl の最初の 3 行のみを示している．

```
# Create a tibble with the contents of all text files in the specified directory.
tbl <- list.files(path="./FINAL_MPG", pattern="*.txt", full.names=TRUE) %>%
       map_df(~ tibble(text=readLines(.x),
                       issue=tools::file_path_sans_ext(basename(.x))))
tbl <- tbl %>%
       group_by(issue) %>%
       mutate(line_num=row_number()) %>%
       ungroup()
# Display the first few rows of the tibble.
tab01 <- tbl %>%
  slice(1:4) %>%
  gt() %>%
  opt_table_font(font="Arial", size=px(10), weight="normal",
                 style="normal", color="black") %>%
  tab_header(title=md("Table 1. First 3 rows of tbl"),
             subtitle=md("Before tokenaization")) %>%
  tab_source_note(source_note=md("_Source_: Generated by authors.")) %>%
  tab_options(heading.title.font.size=px(12),
              heading.title.font.weight="normal",
              heading.subtitle.font.size=px(10),
              heading.subtitle.font.weight="normal",
              source_notes.font.size=px(8)) %>%
  opt_align_table_header(align="left") %>%
  cols_align(align="left", columns=everything())
tab01
```

Table 1. First three rows of tbl. Before tokenization.

text	issue	line_num
GSSMP 2010	GSSMP 2010	1
Medium-term monetary policy principles	GSSMP 2010	2
Over the next three years, the objective of the single state monetary policy will be, as before, to reduce inflation to 9-10% in 2010 and to 5-7% in 2012. While monetary policy will retain its anti-inflationary emphasis, efforts taken by the Bank of Russia in the field of monetary regulation in the short term will largely aim to minimise the negative impact of the global financial and economic crisis on the Russian economy and banking sector.	GSSMP 2010	3
The Bank of Russia shares the Russian government's assessments of the external and domestic conditions of the Russian economy, and consequently, the possible scenarios for its development in 2010 and in the period up to 2012. In the period ahead, the Bank of Russia will increase emphasis on keeping inflation on a downward trend. The Bank of Russia will use all monetary policy instruments at its disposal to restrain inflation.	GSSMP 2010	4

Source: Generated by authors.

2.4 トークン化

tbl の text 列に格納した 1 段落分のテキストをトークン化する．tidytext によるテキスト分析では，データ構造を tibble の 1 行(observation)につき 1 つのトークンだけを持つ tidy data 構造にすることが標準アプローチである．トークンは単語とは限らず，分析目的に合わせて，文，段落，章でも良い．トークンへの分割は tidytext パッケージの unnest_tokens()関数で簡単におこなえる．英文で単語をトークンとする場合，unnest_tokens()関数は対象となるテキストを空白，ハイフン他の記号をマーカとして分割する．

トークン化の際に検討すべきいくつかの分析内容に関連する問題がある．MPG の場合は，文書の性格上，複合語あるいは2単語以上のかたまりで意味をなす用語がかなり存在する．複合語の表記のゆれにどう対応するかという形式的問題と，分析目的からみて複合語を分割するかしないかという両方の問題がある．

MPG では，short-term や hyper-inflation といったハイフンを使った合成語がかなり存在するが，表記法は必ずしも一定していない．unnest_tokens()関数のデフォルトのままならば，hyperinflation は分解されず 1 トークンで，hyper-inflation あるいは hyper と inflation の間にスペースがあれば，hyper と inflation の 2 つのトークンに分割される．2 つのトークンに分割されても，単語の出現順情報は保持されているため，hyper の次のトークンとして inflation を検索し，元の hyperinflation に復元できる．ただし，注意すべき点は，複合語の存在が認識されている場合のみトークン化後に複合語を復元できることである．

hyperinflation という語の存在の可能性が事前に認識されており，かつ hyperinflation を 1 トークンとすることが分析上重要である，つまり hyperinflation と inflation は異なる意味を持つ単語であるとして分析することが重要であるという判断のもとでは，オリジナルテキストの前処理段階で，hyper-inflation や hyper inflation といった表記法を hyperinflation に変換する方法が容易であろう．この場合は，hyperinflation と inflation とは別の分析上の意味をもつ単語とすることになる．ただし，トークン化後に hyperinflation と inflation とを同じグループのトークンとして分析することは可能である．

本研究の段階では，複合語をどのように対処すべきか一般的な判断基準を与えることは難しい．

本研究ではMPGの経済学的分析を目的としていないため，MPG内で複合語の表記法が必ずしも統一されていないという形式的理由により，hyperinflationのように一続きで書かれている場合はそのままで1トークン，それ以外の場合はデフォルトのままトークンに分割する方法をとった．この場合，ハイフンを別の意味で使っていないかを確認することも必要になる．例えば，spelling markを使う言語では，spelling markとしてのハイフンがある．unnest_tokens()関数のデフォルトのままであればspelling markで単語が2つのトークンに分離される．MPGの場合は，spelling markはテキストファイル内で適切に排除されているため，この問題はなかった．

本格的な社会経済分析をおこなう場合には，合成語の分割についてテキスト分析にそれらの単語の分析上の意味の観点から検討する必要がある．ただし，そもそも各文献にどのような複合語が存在するのか，表記法は一貫しているのかを事前に調べる必要がある．トークンへの分割だけでも，分析対象分野と分析対象文献についての事前知識なしで判断することは難しいことがわかる．Table 2は，トークン化した結果を格納したtibbleであるtidy_tblの最初の5行を示している．なおunnest_tokens()関数でトークン化するとトークンはデフォルトですべて小文字に統一される．

```
# Create a tidytext tibble by tokenizing the text column into words.
tidy_tbl <- tbl %>% unnest_tokens(word, text, token="words")
# Display the first few rows of the tibble.
tab02 <- tidy_tbl %>%
  slice(1:10) %>%
  gt() %>%
  opt_table_font(font="Arial", size=px(10), weight="normal",
                 style="normal", color="black") %>%
  tab_header(title=md("Table 2. First five rows of tidy_tbl"),
             subtitle=md("Before deleting stop words")) %>%
  tab_source_note(source_note=md("_Source_: Generated by authors.")) %>%
  tab_options(heading.title.font.size=px(12),
              heading.title.font.weight="normal",
              heading.subtitle.font.size=px(10),
              heading.subtitle.font.weight="normal",
              source_notes.font.size=px(8)) %>%
  opt_align_table_header(align="left") %>%
  cols_align(align="left", columns=everything())

tab02
```

Table 2. First five rows of tidy_tbl. Before deleting stop words

issue	line_num	word
GSSMP 2010	1	gssmp
GSSMP 2010	1	2010
GSSMP 2010	2	medium
GSSMP 2010	2	term
GSSMP 2010	2	monetary
GSSMP 2010	2	policy
GSSMP 2010	2	principles
GSSMP 2010	3	over
GSSMP 2010	3	the
GSSMP 2010	3	next

Source: Generated by authors.

2.6 トークン化後のデータ構造

Table 2 が示すようにトークン化後の tidy_tbl は，tidy data の原則を満たしていない．tidy_tbl の issue 列と line_num 列の値を指定しても，word 列の値を一意に特定することはできない．それどころか，特定 issue の特定段落内に同じ単語があることはいくらでもある．Table 2 ではたまたまその事例は無いが，tidy_tbl は同一内容の行を多数持っているはずである．つまり，tidy_tbl はリレーショナル・データベースの第一次正規化もみたしていない．

実際には，tibble はデータ読み込み時のデータの順序情報を保持しているため，値(ここではトークン)を一意に特定することは可能である．unnest()関数で読み込み時の段落をトークン化した際に段落番号を line_num 列に保持したように，トークンの出現順を明示的に保存することは容易である．tibble には行番号が自動的に付与されるから，row_number()関数で tibble の行番号を取得して mutate()関数で新たな列に保存すればよい．ただし，テキスト分析においてトークンの順序情報は重要であるが，トークンに固定的な id を付与しても分析上あまり役に立つ場面はない．特に tidytext の多くの機能は，tibble の行が読み込み順であることを前提として，つまりトークンの相対的位置関係を使って自動処理してくれる．分析者が独自に開発した分析手法で必要になるという以外には，テキスト分析において分析者が特定トークンを一意に指定しなければならないケース，つまりトークンの絶対的位置を指定することが必要になる場合はほぼないと思われる．

以上のように，トークン化した tidy_tbl でも，データ分析の便宜上，tidy data の原則に従ったデータ構造になっているわけではない．それを補完する情報は，tibble が保持しているトークンの順序情報である．このデータ構造について認識していることはデータを操作する際に重要である．sort や group 等でオリジナルの tibble の行の順番を変更した場合は，その状態で分析を継続するのか，行の順番を元にもどす必要があるのかを注意する必要がある．

2.7 英語以外の言語のトークン化

欧文でも英語以外の言語，例えばロシア語やドイツ語などでは，空白文字だけではトークン化はできない．ロシア語では，空白文字だけでなく，ハイフンやアポストロフィーも単語区切りマーカとして使われる．ドイツ語では，複合名詞や分離動詞が多いため，合成語処理と同様の問題が生じる．日本語，中国語の場合は，空白文字をマーカとしてトークン化することはできない．これらの言語に対しては，R の tidytext パッケージの unnest_tokens()関数を使う前に，それぞれの言語用のトークン化処理をおこなう必要がある．日本語では RMeCabC パッケージ等，中国語では jiebaR パッケージ等を使う．日本語の場合は，例えば次のようになる．

```
tbl <- tibble(text) %>%
unnest_tokens(word, RMeCabC::RMeCabC(japanese_text))
```

一般的な文献に関しては英語以外の言語でもトークン化のためのパッケージが用意されている．しかし，専門用語や専門分野の独自の言い回しに対応したトークン化をおこなうことは，英語以上に検討すべき問題が多いと予想される．多言語における経済分析に適したトークン化処理については，次の 2.6 節の Stop words 除去，3.3 節の語形統一にも関連する．これらの問題は，多言語で社会経済専門文献のテキスト分析をおこなう場合の重要な検討課題である．

2.6 Stop words の除去

stop words は，テキスト分析において意味を持たないとされる日常的に多用される単語を指している．英語では，the や and 等の冠詞や接続詞がこれにあたる．tidytext は，多言語対応の stop words を

除去するための辞書を標準で用意している．また，tidystopwords や stopwrods といった多言語対応の stop words を提供する他のパッケージもある．

専門文献のテキスト分析では，標準的な stop words 辞書が持つ一般的な stop words に加えて分析対象に固有な stop words を追加するかを検討しなければならない．例えば，MPG は Central Bank of Russia を多数含んでおり，デフォルトでは central，bank, of, russia の 4 トークンに分離される．of は一般的な stop word として取り除かれるが，分析用データには数多くの central，bank，russia が残る．トークン化の前段階で Central Bank of Russia を CBR に置き換えておくといった処理をすることで CBR として残すことや，そもそも Central Bank of Russia を削除してしまうことができる．逆に，単語の出現順序情報を利用して，トークン化後に central, bank, russia と並ぶ 3 行を削除することも可能である．後者は，意図しない結果をもたらさないことを十分に確認する必要がある．CBR として残す場合，あるいは何も処理をしない場合は，cbr あるいは central，bank，russia という語が分析対象に多数含まれていることを意識していなければならない．

本研究では，Central Bank of Russia をオリジナルテキストの前処理として削除した．Central Bank of Russia がテキスト分析上の重要な意味を持つとは思われず，一方，トークンに分けた場合，bank がロシア中央銀行を意味する場合と他の銀行を意味する場合とを区別することが必要になると考えられたからである．このように，stop words の選択においても，分析対象の特性と分析目的にもとづいて stop words を検討することが必要になる．この問題の重要性を確認するために，次章は，標準的 stop words を利用した場合とカスタマイズした stop words を利用した場合でトークンの出現頻度分析をおこない，結果を比較する．

3. トークンの頻度分析と stop words の検討

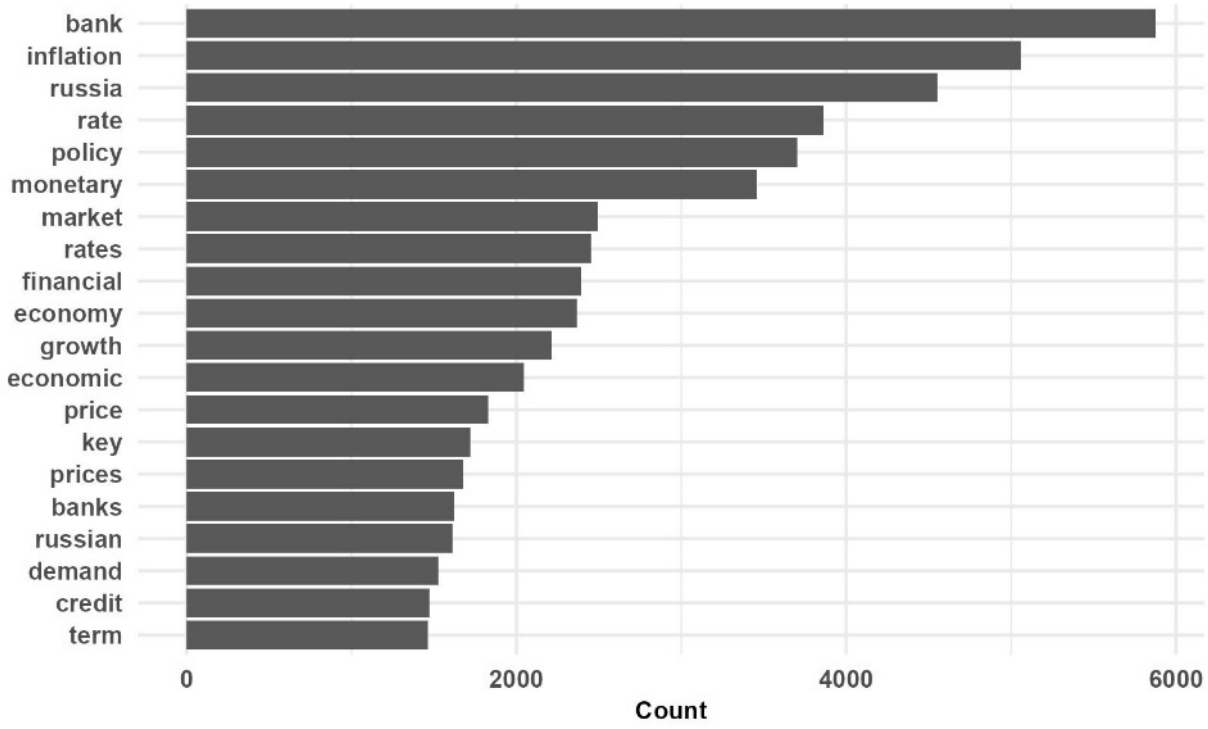
3.1 標準的な stop words 辞書による頻度分析

まず tidytext の標準 stop words 辞書で stop words を除去した場合のトークン頻度分析をおこなう．トークン化後のデータ形式が tidytext 化されていれば，tidytext 化された tibble と stop words 辞書とに anti_join()関数を適用することにより stop words を除去できる．anti_join(a,b,c)関数は，c で指定した列で 2 つの集合 a，b の積集合をとり，a 集合からその積集合を除いた集合を返す．以下のコードでは，tidytext の stop word の標準辞書から英語の stop words を取り出している．その後 anti_join()関数により，word 列に，stop words に含まれるトークンを word 列に持つ行を，tidy_tbl から削除している．Figure 1 は，標準 stop words を取り除いた後の 2010 年版(2009 年発行)から 2025 年版(2024 年発行)の MPG 全体について出現数上位 20 トークンを示している．

```
# Load the stop words from the tidytext package and filter for English stop
words.stop_words <- tidytext::stop_words %>% select(word)
# Delete the standard stop words from the tibble.
tidy_tbl <- tidy_tbl %>%
  dplyr::anti_join(stop_words, by="word")
g_top20 <- tidy_tbl %>%
  dplyr::count(word, sort=TRUE) %>%
  head(20) %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(n, word)) +
    geom_col() +
    labs(title="Figure 1. Top 15 tokens. All issues.",
         subtitle="after deleting standard stop words",
         caption="Source: generated by authors.", y=NULL, x="Count") +
```

```
    theme_minimal() +
    theme(plot.title=element_text(face="bold",size=10),
          plot.subtitle=element_text(face="bold",size=8),
          plot.caption=element_text(face="italic", size=8, color="black", hjust=0),
          axis.title.x =element_text(face="bold",size=8),
          axis.title.y =element_text(face="bold",size=8),
          axis.text.x =element_text(face="bold",size=8),
          axis.text.y =element_text(face="bold",size=8))
g_top20
```

Figure 1. Top 20 tokens in all issues. After deleting standard stop words.



Source: generated by authors.

次に，各年次別のトークンの頻度分析をおこなう．MPG は各年で文書量が変わるため，ある issue のトークン出現数をその issue のトークン総数で割ったトークンの出現率で比べる．Table 3 は，2010 年版(2009 年発行)から 2025 年版(2024 年発行)の各 MPG での出現比率上位 15 トークンを示している．なお，Table 3(B)作成用コードは(A)と同等のため省略しているが，Appendix の Markdown ファイルには含まれている．

```
freq_by_issue <- top_tokens %>%
  arrange(issue, desc(ratio)) %>%
  unnest(word) %>%
  select(issue, word) %>%
  group_by(issue) %>%
  mutate(row=row_number()) %>%
  pivot_wider(names_from=issue, values_from=word)

freq_by_issue_gt1 <- freq_by_issue %>%
  select(1:9) %>%
```

```
  gt() %>%
    opt_table_font(font="Arial", size=px(10), weight="normal",
                   style="normal", color="black") %>%
    tab_header(title=md("Table 3 (A). Top 20 tokens in 2010-17 (ratio)"),
               subtitle=md("After deleting standard stop words")) %>%
    tab_source_note(source_note=md("_Source_: Generated by authors.")) %>%
    tab_options(heading.title.font.size=px(12),
                heading.title.font.weight="normal",
                heading.subtitle.font.size=px(10),
                heading.subtitle.font.weight="normal",
                source_notes.font.size=px(8)) %>%
    opt_align_table_header(align="left") %>%
    cols_align(align="left", columns=everything()) %>%
    cols_label(`row`="Rank",
               `GSSMP 2010`="2010",
               `GSSMP 2011`="2011",
               `GSSMP 2012`="2012",
               `GSSMP 2013`="2013",
               `GSSMP 2014`="2014",
               `GSSMP 2015`="2015",
               `GSSMP 2016`="2016",
               `MPG_2017`="2017") %>%
  cols_align(align="left", columns=everything())
freq_by_issue_gt1
```

Table 3 (A). Top 20 tokens in 2010-17 (ratio). After deleting standard stop words.

Rank	2010	2011	2012	2013	2014	2015	2016	2017
1	bank	bank	bank	bank	bank	bank	bank	bank
2	russia	russia	russia	russia	russia	russia	russia	russia
3	credit	2010	2011	2012	rate	policy	2015	economic
4	2009	billion	billion	monetary	policy	inflation	policy	inflation
5	billion	credit	credit	billion	monetary	growth	rate	policy
6	roubles	rate	monetary	credit	2013	monetary	monetary	growth
7	foreign	roubles	roubles	rate	exchange	market	growth	rate
8	growth	monetary	january	market	inflation	rate	market	monetary
9	monetary	growth	institutions	russia’s	growth	2014	economic	financial
10	exchange	banking	september	financial	market	economic	financial	economy
11	banking	financial	banking	exchange	economic	rates	operations	rates
12	institutions	exchange	foreign	policy	liquidity	operations	inflation	conditions
13	january	institutions	rate	institutions	operations	foreign	liquidity	price
14	market	market	policy	2015	russia’s	exchange	foreign	2016
15	trillion	2011	growth	growth	2014	economy	rates	external

Table 3 (B). Top 20 tokens in 2018-25 (ratio). After deleting standard stop words.

Rank	2018	2019	2020	2021	2022	2023	2024	2025
1	inflation	inflation	inflation	bank	bank	bank	inflation	inflation
2	bank	bank	bank	inflation	inflation	inflation	bank	bank
3	russia	russia	rate	rate	policy	russia	policy	rate
4	rate	rate	policy	russia	russia	rate	monetary	policy
5	price	policy	monetary	policy	monetary	policy	rate	russia
6	financial	rates	russia	monetary	rate	monetary	russia	monetary
7	rates	monetary	growth	rates	2021	economy	economy	rates
8	market	market	rates	financial	economy	financial	financial	economy
9	growth	price	2019	economy	market	market	rates	market
10	economic	prices	financial	market	rates	banks	market	2024
11	expectations	growth	market	key	demand	2022	2023	demand
12	policy	financial	factors	economic	growth	economic	key	financial
13	economy	economic	economic	2020	key	key	target	banks
14	prices	economy	price	price	economic	russian	banks	key
15	level	expectations	economy	banks	price	rates	demand	growth

Source: Generated by authors.

Table 3 の単語をみると，第 1 に，Table 3 は MPG 文書の性格から多数含まれるであろうと考えられる単語から構成されているが，それらのうち分析上意味がある可能性の高い単語と意味が乏しいと考えられる単語が混在していることがわかる．前者は，bank，inflation，monetary，rate, market といった単語が該当する．後者は，2010，2011 といった年号，月名，billion といった単位，russia などの単語である．ただし，両者の境界はあいまいであり，各単語が分析上の意味を持っているかどうかを一律に判断することは困難である．例えば，inflation は単に定型的な事実としてのインフレ率報告のために使われている場合もあれば，インフレ激化が問題となっているという政策問題のコンテクストで使われている場合もあるだろう．

第 2 に，単数形，複数形，所有格，現在形，過去形といった単語の語形が別の単語としてカウントされている．例えば，bank は，banking, banks, bank's といった語形がある．テキスト分析において，ある単語の語形を別の単語として認識すべきであるのか，あるいは異なる語形の語も同一の単語として分析すべきかという問題も，一般的に答えることは難しい．語形をどう扱うべきかは文献の専門分野，性格だけでなく，単語ごとにも異なると考えられる．

3.2 Stop words の拡張と除去

この研究ノートでは，stop words の拡張は，もっぱら形式的な観点からのみおこなった．分析上意味を持たないことがほぼ自明な単語について，それらの異なる語形を含めて除去した．異なる語形のうち，単数形と複数形をわけることの意味が乏しいと考えられる単語について，トークンそのものを書き換える方法で統一した．

拡張した stop words で stop words を除去する方法として，第 1 に，標準 stop words 辞書に単語を追加したうえで，anti_join()関数を使えば，拡張した stop words を一挙に除去できる．以下のコードでは，この方法で西暦を意味する 2000 から 2025 を削除し，その結果を新たな tibble である tidy_ctm_tbl に格納している．ただし，以下のコードは手順を示すことが目的であり，次の数字の除去の際に西暦も削除されるため，重複して実行する必要はない．

この方法では西暦以外の意味をもつ 2000 といった数字も除去される．本研究では，数字は基本的に除去するため，この点は問題にならない．分析と分析目的によっては，西暦も数字も残す必要の

ある場合がありうる．これも，分析対象についての専門知識がないと stop words の除去が困難であることを示す例である．

```
# Add custom stop words to stop_words
years_lst <- as.character(2000:2025)
custom_stop_words <- bind_rows(tibble(word=years_lst, lexicon="custom"),
                               stop_words)

# Delete the standard stop words from the tibble.
tidy_ctm_tbl <- tidy_tbl %>%
  dplyr::anti_join(custom_stop_words, by="word")
```

拡張した stop words を除去する第 2 の方法として，トークン化後の tibble に対して filter()関数を使って除去する方法がある．filter()関数は tibble の行を一定条件で選択するための関数である．この方法では，行の選択条件として正規表現による条件を使えるため，ある語の語形も一括して除去できる．例えば，russia*によって russian，russia's 等を一括して除去することができる．正規表現を使うと除去すべき単語を柔軟に指定できるが，正規表現で指定されるすべての語を直感的に把握することは，容易ではない．除去すべきでない単語を誤って除去しないように十分注意する必要がある．

以下では，テキスト分析において意味が乏しいと思われる MPG 文書に特有な単語をそれらの語形を含めて除去する．ただし，ここでも分析に必要となる可能性を考慮して，ロシア中央銀行名か他の特定銀行名をほぼ意味する単数形の bank は除去したが，集合的に銀行部門を意味する banks は残した．

filter()関数で正規表現を使うためには，正規表現にマッチするかどうかを TRUE/FALSE で返す関数である grepl()関数を利用する．grepl()関数は，正規表現を 1 つの文字列(pattern)としてのみ受けとるため，正規表現によって複数の word を除去する場合は複数の正規表現を 1 つの文字列にまとめる必要がある．月名，ローマ数字，数字，期間を示す単語，数値の単位，MPG 文書の様式に関連する単語を除去するための正規表現パターンを分けて用意し，最後に ptterns_re として 1 つのパターンにまとめている．なおパターン中のフェンスは OR 演算を意味する．grepl()関数前に否定演算子!を付けることで，正規表現パターン patterns_re と word 列の値が一致しない tidy_ctm_tbl の行だけを選択できる．

```
# Create a regular expression pattern for the additional custom stop words
month_re <-
c("january*|february*|march*|april*|may*|june*|july*|august*|september*|october*|nove
mber*|december*")
numstr_re <-
c("^one|^two|^three|^four|^five|^six|^seven|^eight|^nine|^ten|^eleven|^twelve|^thirte
en|^fourteen|^fifteen|^sixteen|^seventeen|^eighteen|^nineteen|^twenty|^thirty|^forty|
^fifty|^sixty|^seventy|^eighty|^ninety|^ii|^iii|^iv|^v|^vi|^vii|^viii|^ix|^x|^xi|^xii
|^xiii|^xiv|^xv|^xvi|^xvii|^xviii|^xix|^xx|^i[a-e]|^ii[a-e]|^iii[a-e]|^iv[a-e]|^v[a-
e]")
num_re <- c("[0-9]|[0-9][0-9]|[0-9]\\.[0-9]|[0-9][0-9]\\.[0-9]|[0-9][0-9][0-9]|[0-
9][0-9][0-9][0-9]")
length_re <- c("^annual|^period|^quarter|^year|^month|^week|^day|^term|^time")
unit_re <- c("^hundred|^thousand|^million|^billion|^trillion|^p\\.|^pp$")
MPG_re <-
c("^mpg|^gssmp|^policy|^sector|^single|^russia|^central|bank|^econom|^rate$|^rouble|^
```

```
ruble|^factor|^issue$|^issues$|^variant|^measure|^yoy$|^mom$")
sp_style_re <- c("^e\\.g|^i\\.e|^etc|^figure|^table|^source|^note")

# Combine all patterns into a single vector
patterns_re <- paste(month_re, numstr_re, num_re, length_re, unit_re, MPG_re,
                     sp_style_re, sep="|")

tidy_ctm_tbl <- tidy_ctm_tbl %>%
  filter(!grepl(patterns_re, word, ignore.case=TRUE))
```

3.3 語形の統一

stop words の拡張に加えて，bank と banks，price と prices，import と imports のように，stop words ではないが，単数形と複数形をどのように扱うかが問題となる単語があった．一般的なテキスト分析では，単語の異なる語形あるいはある単語とその類似単語の処理についての考え方として，次のような考え方がある(Silge and Robinson, 2025).

1. Stemming
Stemming は，単語を語幹(root form)に統一する．例えば，“cats” を “cat” に，“running” を “run” に変換する．
2. Lemmatization
Lemmatization は，より洗練された語形統一法とみなされる．単語の意味と文脈を考慮して単語を辞書的な基本形に変換する．例えば，“better” を “good” に変える．適切に処理する限り，一般に Stemming よりも適切な単語変換が可能と考えられるが，作業量はあきらかに大きくなる．R では，textstem パッケージを使うことで Lemmatization をおこなうことができる．
3. 単数形への統一
名詞の場合は単数形に統一する場合が多い．MEG 試行分析でもちいている一般的なテキスト分析手法では，ある語の単数形と複数形を別のものとしたとしても，なぜ単語がその語形で使われているのかというレベルでの分析は結局できないからである．
4.異なる語形の別単語化
単数形と複数形の文章上での意味の違いを現在のテキスト分析手法では明確に分析できないとしても，単数形と複数形は異なる意味を持っている以上，異なる語として扱うという考え方である．これは，結局，第 3 点と同じ理由で異なる結論に至っていることになる．
5. ドメイン依存の処理
最後に，特定の分野，特定の文書，特定の分析目的によって，単語の複数形と単数形の意味の違いが重要であるかどうかを判定し，1-4 の方法で対応するという方法である．

5 を考慮すれば，語形統一には標準的ルールはないということになる．SNS のような多様なトピックと多様なスタイルが入り混じった文章が分析対象である場合は，ランダム化効果が働くため，複数形と単数形を区別しても統一しても分析結果にあまり大きな影響はないかもしれない．MPG のように，特定のトピックに関する特定の文書を分析する場合は，単数形と複数形の区別をどのようにおこなうかは，その分野固有の単語の使い方に加えて，文章スタイルや扱うトピックの時間的変化もあるため，機械的，画一的判断は難しい．

本研究では，語形統一が MPG のテキスト分析に与える影響が現段階では十分わからないため，標準的な語形統一辞書は使わず，prices，imports，rates の複数形だけを単数形に統一することにした．price，rate，import は単数形，複数形とともに頻出単語の上位に位置する．import は動詞である可能

性もあるが，本稿でおこなうテキスト分析では，importの名詞と動詞を区別することが分析に影響するとは考えにくい．price については，複数形の prices が物価の意味を持つ可能性があるが，やはり単数形への統一が分析上の問題となるとは考えにくい．加えて，MPG では，物価を明示的に示すときにはCPI，WPI等が使われることが多い．これらは直感的な判断に過ぎない．語形統一についても，分析対象と分析目的に応じた判断をするには，専門分野の文献に対するテキスト分析経験を蓄積することしかないように思われる．

tidytext に従った tibble データでは，単数形と複数形の統一は str_replace()関数あるいは case_when()関数によって簡単に処理できる．以下のコードでは，str_replace()関数で置き換えた値で tidy_ctm_tbl の word 列の該当する値を上書きしている．

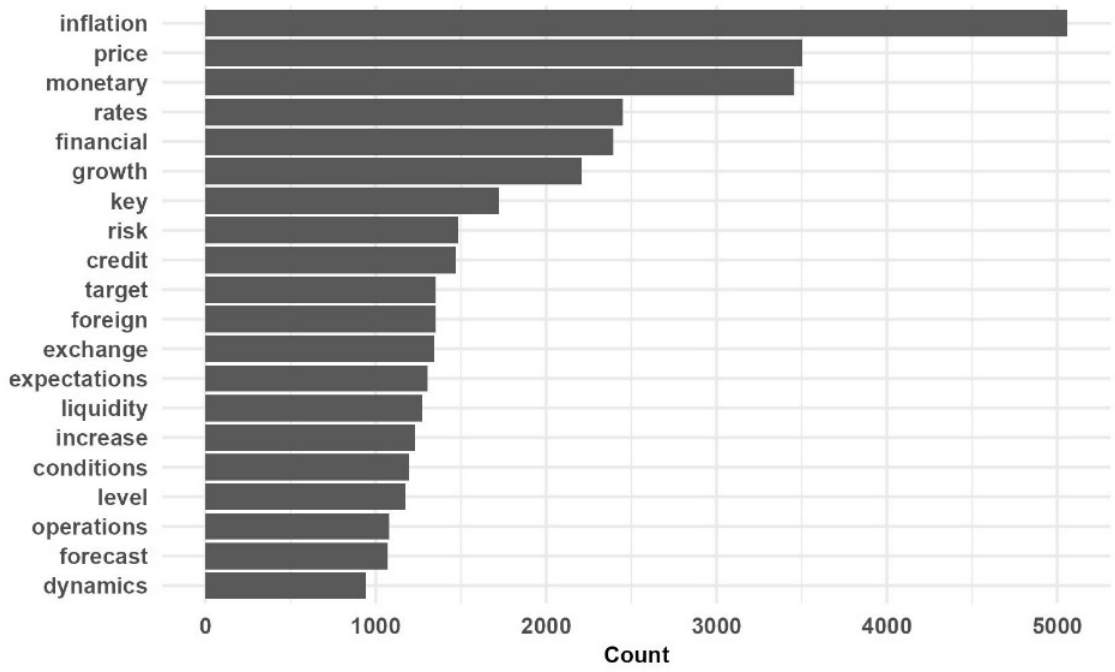
```
# Replace "prices" with "price" and "imports" with "import"
# ^, & indicate the exactl top and bottom of a word, respectively.
replace_patterns <- c("^exports$"="export",
                      "^imports$"="import",
                      "^prices$"="price",
                      "^repos$"="repo",
                      "^risks$"="risk",
                      "^targets$"="target")
tidy_ctm_tbl <-tidy_ctm_tbl %>%  mutate(word=str_replace_all(word, replace_patterns))
```

3.4 stop words 拡張と語形統一後のトークン頻度分析

Figure 2 は，stop word 追加と語形統一後の全 MPG 文書の出現数上位 20 トークンを示している．

```
# top 20 tokens count all issue
g_ctm_top20 <- tidy_ctm_tbl %>%
  dplyr::count(word, sort=TRUE) %>%
  head(20) %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(n, word)) +
    geom_col() +
    theme_minimal() +
    labs(title="Figure 3. Top 20 tokens. All issues.",
         subtitle="after processing extended stop words and words unification",
         caption="Source: Generated by authors.",
         y=NULL,
         x="Count",
         hadjust=0) +
    theme(plot.title=element_text(face="bold",size=10),
          plot.subtitle=element_text(face="bold",size=8),
          axis.title.x =element_text(face="bold",size=8),
          axis.title.y =element_text(face="bold",size=8),
          axis.text.x =element_text(face="bold",size=8),
          axis.text.y =element_text(face="bold",size=8),
          plot.caption=element_text(face="italic",size=8,hjust=0))
g_ctm_top20
```

Figure 2. Top 20 tokens in all issues. After processing extended stop words and words unification.



Source: Generated by authors.

Table 4 は，MPG 各年版で出現率の高い 10 トークンを示している．ただし，Table 4(B)作成用コードは印刷版では省略している．

```
token_ctm_counts <- tidy_ctm_tbl %>%
  dplyr::count(issue, word, sort=TRUE)
# total number of token by issue
total_ctm_tokens <- tidy_ctm_tbl %>%
  count(issue) %>%
  rename(total=n)
# token number/ total
token_ctm_ratios <- token_ctm_counts %>%
  left_join(total_ctm_tokens, by="issue") %>%
  mutate(ratio=n / total)
# top 20 ratios by issue
# slice_head() is used to select the exactly top 20 tokens by ratio for each issue.
# slice_max() can select more than 20 if there are ties.
top_ctm_tokens <- token_ctm_ratios %>%
  group_by(issue) %>% arrange(desc(ratio)) %>% slice_head(n=10)
freq_ctm_by_issue <- top_ctm_tokens %>%
  arrange(issue, desc(ratio)) %>%
  unnest(word) %>%
  select(issue, word) %>%
  group_by(issue) %>% mutate(row=row_number()) %>%
  pivot_wider(names_from=issue, values_from=word)
# make gt tables
freq_ctm_by_issue_gt1 <- freq_ctm_by_issue %>%
  select(1:9) %>%
  gt() %>%
  opt_table_font(font="Arial", size=px(10), weight="normal", style="normal",
                 color="black" ) %>%
  tab_header(title=md("Table 4. (A) Top 10 tokens by issue in 2010-17"),
```

```
            subtitle=md("with extended stop words and words unification")) %>%
  tab_source_note(source_note=md("_Source_: Generated by authors.")) %>%
  tab_options(heading.title.font.size=px(12),
              heading.title.font.weight="normal",
              heading.subtitle.font.size=px(10),
              heading.subtitle.font.weight="normal",
              source_notes.font.size=px(8)) %>%
  opt_align_table_header(align="left") %>%
  cols_align(align="left", columns=everything()) %>%
  cols_label(`row`="Rank",
             `GSSMP 2010`="2010",
             `GSSMP 2011`="2011",
             `GSSMP 2012`="2012",
             `GSSMP 2013`="2013",
             `GSSMP 2014`="2014",
             `GSSMP 2015`="2015",
             `GSSMP 2016`="2016",
             `MPG_2017`="2017") %>%
  cols_align(align="left", columns=everything())
freq_ctm_by_issue_gt1
```

Table 4. (A) Top 10 tokens by issue in 2010-17. with extended stop words and words unification

Rank	2010	2011	2012	2013	2014	2015	2016	2017
1	credit	credit	credit	monetary	monetary	inflation	monetary	inflation
2	foreign	monetary	monetary	credit	exchange	growth	growth	price
3	growth	growth	institutions	financial	inflation	monetary	price	growth
4	monetary	financial	foreign	exchange	growth	price	financial	monetary
5	exchange	exchange	growth	institutions	liquidity	rates	operations	financial
6	institutions	institutions	financial	growth	operations	operations	inflation	rates
7	financial	foreign	increase	rates	target	foreign	liquidity	conditions
8	currency	risk	exchange	foreign	rates	exchange	foreign	external
9	net	operations	account	operations	financial	fx	rates	stability
10	federal	money	risk	price	price	currency	credit	liquidity

Table 4. (B) Continued.

Rank	2018	2019	2020	2021	2022	2023	2024	2025
1	inflation	inflation	inflation	inflation	inflation	inflation	inflation	inflation
2	price	price	price	monetary	monetary	monetary	monetary	monetary
3	financial	rates	monetary	price	price	price	price	price
4	rates	monetary	growth	rates	rates	financial	financial	rates
5	growth	growth	rates	financial	growth	key	rates	financial
6	expectations	financial	financial	key	key	rates	target	key
7	level	expectations	key	risk	financial	exchange	key	growth
8	risk	key	risk	growth	liquidity	foreign	growth	target
9	monetary	level	target	credit	impact	conditions	risk	increase
10	credit	risk	scenario	target	expectations	target	foreign	level

Source: Generated by authors.

より視覚的にトークンの出現頻度を把握するために，テキスト分析でよく使われる wordcloud チャートを作成する．Figure 3 は，2010 年版, 2015 年版, 2020 年版, 2025 年版 MPG において出現率が高い上位 40 トークンを抽出し，出現率が高いほど大きく表示している．

```
wcld_lst <- c("GSSMP 2010", "GSSMP 2015", "MPG_2020", "MPG_2025")
wcld <- list()
wcld_com_lst <- list() # use this later for patchwork
```

```r
set.seed(42) # For reproducibility

for (i in seq_along(wcld_lst)){
  tmp_tbl <- tidy_ctm_tbl %>%
    filter(issue==wcld_lst[[i]]) %>%
    count(word, sort=TRUE) %>%
    slice_max(n, n=40, with_ties=FALSE)

  wcld[[i]]<- ggplot(tmp_tbl, aes(label=word, size=n)) +
    theme_minimal() +
    geom_text_wordcloud_area(eccentricity=1) +
    scale_size_area(max_size=16) +
    labs(title=wcld_lst[[i]], y=NULL, x=NULL, hjust=0.1) +
    theme(plot.title=element_text(face="bold.italic",size=8))

  wcld_com_lst <- c(wcld_com_lst, paste0("wcld[[",i,"]]"))
}
# to show all plots in one plot
wcld_com <- stringr::str_c(wcld_com_lst, collapse="+")

wcld4 <- eval(parse(text=wcld_com)) +
  plot_layout(nrow=2, ncol=2) +
  plot_annotation(title="Figure 4. Top 40 tokens in each year (count)",
                  caption="Source: generated by authors.",
                  theme=theme(plot.title=element_text(size=10,
                                face="bold", color="black", hjust=0),
                              plot.caption=element_text(size=8,
                                face="italic", color="black", hjust=0)))
wcld4
```

Figure 3. Top 40 tokens in each year (count)

Source: generated by authors.

3.5 感情分析

stop words を除去し，語形を統一したトークンにより感情分析を試行的におこなう．感情分析はテキストの感情的なトーンを評価するための手法である．感情分析をおこなうためには，ある単語が positive か negative かを振り分ける感情辞書が必要となる．本研究では，tidytext パッケージで標準的に提供されている感情辞書を使った．感情辞書の単語の振り分けは，文，段落，文章全体のコンテクスト中の単語の意味は考慮していない．コンテクスト中での単語の意味を考慮しないことの制約はここでは問題にしないとしても，特定専門分野に一般的な感情辞書を使うと的外れな結果になる可能性が高い．MPG の場合，次の感情分析 word cloud が示すように，risk，debt，decline 等の語が，テキストが否定的感情のものであるという評価に強く貢献しているという結果になる．通貨政策と金融情勢について客観的に記述することを目的とする MPG の性格上，これらの語が否定的感情を意味するとは限らず，中立的あるいは肯定的意味で使われている場合もある．感情分析は多くの分野

で応用されているが，専門分野のテキスト分析においては分野に適した感情辞書の選択と構築が重要な課題である(Akdogan and Anbar, 2024)．ここでおこなう感情分析が試行にとどまるのは，MPG 分析に適した感情辞書を用いていないからである．社会経済分析用の感情辞書を整備することは今後の課題である．

Figure 4 は，tidytext の comparison.cloud()関数を使って，positive な単語と negative な単語を色分けし，出現率の高い 80 トークンを表示している．利用した感情辞書は bing である．bing は positive, negative の度合いを定量化せずに，単語を単に positive, negative のどちらかに振り分ける．R で一般的に使われる作図用パッケージ ggplot2 を使った場合と異なり，comparison.cloud()関数で作成した plot はオブジェクトとしては保存できない．comparison.cloud()関数で作成した図は，jpeg()関数等の R の標準的なイメージファイル保存用の関数で，ファイルとして保存できる．ggplot2 パッケージの geom_text_wordcloud_area()関数を使っても wordcloud チャートを簡単に作ることができ，作成した図を object として保存できる．しかし，Figure 4 のような positive 語と negative 語を織り交ぜて対比する wordcloud チャートについては comparison.cloud()関数で作成する方が容易である．

```
jpeg(filename="fig05.jpg", width=120, height=120, units="mm", quality=100, res=300)
sent_tbl1 <- tidy_ctm_tbl
sent_tbl1 %>%
  inner_join(get_sentiments("bing")) %>% count(word, sentiment, sort=TRUE) %>%
  acast(word ~ sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors=c("red1", "green4"), max.words=80, title.size=1,
                   title.colors=c("red1", "green4"),
                   title.bg.colors=c("grey90", "grey90"))
title(main="Figure 4. Top 80 sentiment tokens",
      line=3, col.main="black", font.main=1, cex.main=0.9, adj=0, outer=FALSE)
mtext("Source: Generated by authors.",
      side=1, adj=0, padj=0, line=4, cex=0.6, col="black", font=3)
dev.off()
```

Figure 4. Sentiment cloud.

Source: generated by authors.

3.5 Term frequency - inverse document frequency (TF-IDF) 分析

個々の単語の出現頻度あるいは出現率のみで単語の意味上の重要性を評価することは困難である．TF-IDF 分析は，文書中の単語の重要性を複数文書間で比較することを可能にする．MPG の場合は，TF-IDF により各出版年 MPG を分析することで，注目された経済問題と重点が置かれた政策課題の時間的変遷を分析できる．

TF-IDF では，特定の単語の重要性を，その単語の 1 文書内の出現頻度(TF)にその単語が全体の文書集合においてどれだけ一般的であるか(IDF)というウェイトを付けて評価する．ある文書集合 D 中の文書 d 内のトークン t の $\text{TF-IDF}_{d,t}$ は，

$$(TF-IDF)_{d,t} = TF_{d,t} \cdot IDF_t = \frac{Freq_{d,t}}{T_d} \cdot ln\frac{N_d}{N_{d,t}}$$

となる．ここで，$Freq_{d,t}$ は文書 d 内のトークン t の出現数，T_d は文書 d 内のトークンの総数，N_d は文書集合 D に含まれる文書の総数，$N_{d,t}$ は文書集合 D に含まれるトークン t を含む文書の総数である．TF-IDF は，単語が特定の文書内でどれだけ頻繁に出現するか(Term Frequency, TF)と，全体の文書集合においてその単語がどれだけ一般的であるか(Inverse Document Frequency, IDF)の組み合わせである．tidytext パッケージの bind_tf_idf()関数を使うことで簡単に TF-IDF を計算できる．Table 5 は，tidy_ctm_tbl から TF-IDF を計算した結果の 10 行のみを示している．

```
# Count the number of tokens by issue and word
ctm_iss_wrds_tbl <- tidy_ctm_tbl %>% dplyr::count(issue, word, sort=TRUE)
# ungroup() is not needed, because the result of count() is stored in the new tibble.

# total number of token by issue
ctm_ttl_wrds <- ctm_iss_wrds_tbl %>% group_by(issue) %>% summarize(total=sum(n))

#ctm_iss_wrds_tbl: issue, word, n, total of words in each issue
ctm_iss_wrds_tbl <- left_join(ctm_iss_wrds_tbl, ctm_ttl_wrds)
## Joining with `by = join_by(issue)`
# Calculate TF-IDF for the tidy_ctm_tbl
tf_idf_tbl <- ctm_iss_wrds_tbl %>% bind_tf_idf(word, issue, n) %>%
  arrange(desc(tf_idf)) %>% rowid_to_column(var="Rank")

gt_tf_idf <- slice(tf_idf_tbl, 1:10) %>%
  gt() %>%
  opt_table_font(font="Arial", size=px(10), weight="normal") %>%
  tab_header(title=md("Table 5. Top 20 TF-IDF tokens in all issues")) %>%
  tab_source_note(source_note=md("_Source_: Generated by authors.")) %>%
  tab_options(heading.title.font.size=px(12),
              heading.title.font.weight="normal",
              heading.subtitle.font.size=px(10),
              heading.subtitle.font.weight="normal",
              source_notes.font.size=px(8)) %>%
  opt_align_table_header(align="left") %>%
  cols_align(align="left", columns=everything()) %>%
  cols_align(align="left", columns=everything()) %>%
  fmt_number(columns=c(tf,idf,tf_idf), decimals=5)
gt_tf_idf
```

Table 5. Top 20 TF-IDF tokens in all issues

Rank	issue	word	n	total	tf	idf	tf_idf
1	MPG_2022	pandemic	101	18387	0.00549	1.16315	0.00639
2	MPG_2020	chart	89	16426	0.00542	0.98083	0.00531
3	MPG_2021	pandemic	81	18317	0.00442	1.16315	0.00514
4	MPG_2025	lcr	36	25116	0.00143	2.77259	0.00397
5	MPG_2023	controls	42	20429	0.00206	1.67398	0.00344
6	MPG_2023	digital	101	20429	0.00494	0.69315	0.00343
7	GSSMP 2015	fx	47	5199	0.00904	0.37469	0.00339
8	MPG_2025	subsidised	59	25116	0.00235	1.38629	0.00326
9	GSSMP 2011	nda	10	4567	0.00219	1.38629	0.00304
10	MPG_2024	digital	102	23403	0.00436	0.69315	0.00302

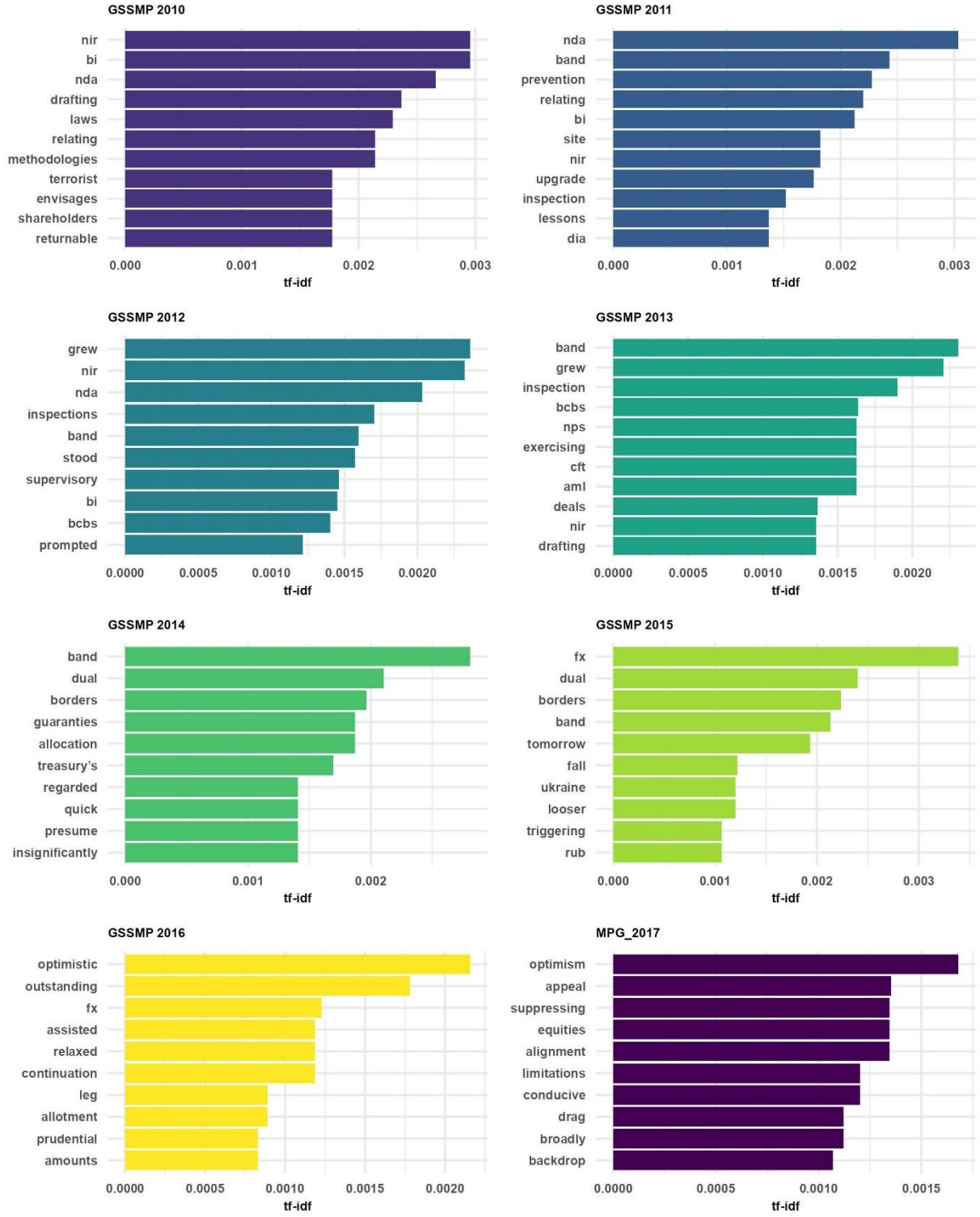
Source: Generated by authors.

Figure 6(A)，6(B)は，各年の TF-IDF 上位 10 トークンを示している．ただし，Figure 6(B)を作成するコードは印刷版では省略している．この研究の目的の 1 つは，stop words の設定や語形統一の方針を決定するために予備的検討をおこなうこととであり，ロシア金融経済の本格的分析は目的としていない．しかし，TF-IDF 分析結果だけでもクリミア併合，コロナ・パンデミック，ウクライナ戦争という大きなショックを反映していると思われる興味深い結果が得られている．同時に，Figure 6(A), 6(B)は，分析データが sa，ap といった分析上意味のない単語を含んでいることも示している．

```
gg_tf_idf_tbl <- tf_idf_tbl %>%
  group_by(issue) %>%
  slice_max(order_by=tf_idf, n=10, with_ties=TRUE) %>%
  ungroup() %>% arrange(issue, desc(tf_idf))

gg_tfidf1 <-list()  # list to store ggplots
gg_tfidf2 <-list()  # list to store ggplots
gg_tfidf1_cmd <-list() # list tp store text commands for patchwork
gg_tfidf2_cmd <-list() # list tp store text commands for patchwork
big_h <- ceiling(length(issue_list)/2)

for (i in seq_along(issue_list)) {
  j <- issue_list[[i]]
  isu_tfidf <- gg_tf_idf_tbl %>% dplyr::filter(issue==j)

  g <- isu_tfidf %>%
    ggplot(aes(x=tf_idf, y=fct_reorder(word, tf_idf, .fun=sum, .desc=FALSE),
fill=issue)) +
    theme_minimal() +
    geom_col(show.legend=FALSE, fill=vi8_cols[(i %% 8)+1]) +
    labs(title=j, x="tf-idf", y=NULL)+
    theme(plot.title=element_text(face="bold",size=6),
          plot.subtitle=element_text(face="bold",size=6),
          axis.title.x =element_text(face="bold",size=6),
          axis.title.y =element_text(face="bold",size=6),
          axis.text.x =element_text(face="bold",size=6),
          axis.text.y =element_text(face="bold",size=6))

  if (i<=big_h){
    gg_tfidf1[[i]] <- g
```

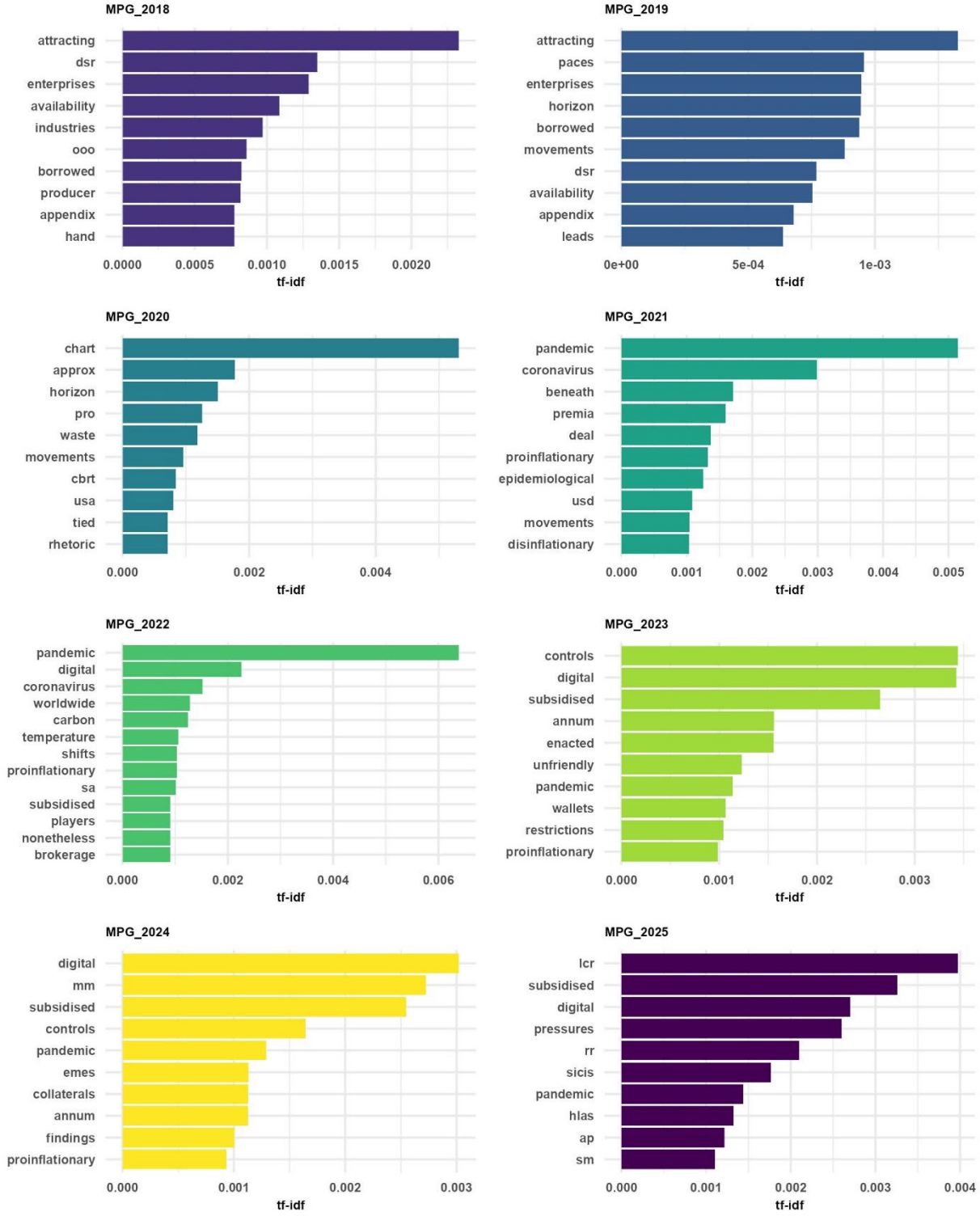
```
    gg_tfidf1_cmd <- c(gg_tfidf1_cmd, paste0("gg_tfidf1[[",i,"]]"))
  }else{
    gg_tfidf2[[i]] <- g
    gg_tfidf2_cmd <- c(gg_tfidf2_cmd, paste0("gg_tfidf2[[",i,"]]"))
  }
  g <- ggplot() #initialization
}

gg_tfidf1_eva <- stringr::str_c(gg_tfidf1_cmd, collapse="+")
gg_tfidf2_eva <- stringr::str_c(gg_tfidf2_cmd, collapse="+")

gg_tfidf_iss1 <- eval(parse(text=gg_tfidf1_eva)) +
  plot_layout(nrow=4, ncol=2) +
  plot_annotation(title="Figure 6 (A). Top 10 TF-IDF tokens in 2010-17",
                  caption="Source: generated by authors.",
                  theme=theme(plot.title=element_text(size=10,
                                face="bold", color="black", hjust=0),
                              plot.caption=element_text(size=8,
                                face="italic", color="black", hjust=0)))
gg_tfidf_iss1; gg_tfidf_iss2
```

Figure 5 (A). Top 10 TF-IDF tokens in 2010-17.



Source: generated by authors.

Figure 5 (B). Top 10 TF-IDF tokens in 2018-25.



Source: generated by authors.

4. N-grams 分析

n-grams 分析は，テキストデータ内の連続した n 個の単語の組み合わせを分析する手法である．これまでの 1 トークン(単語)による分析に対して，文章中のある単語の意味を同じ文章中の他の単語との関係を考慮して分析できる．Artificial Intelligence を用いた大規模言語モデル(LLM)における self-attention 機構のもっとも基本的な形態とみることができる．

n-grams は n 個のトークンを 1 連のトークンとして扱うことを意味している．例えば，“I love R”という文章の 2-grams は”I love”と”love R”の 2 つの 2-grams になる．3-grams は”I love R”の 1 つとなる．問題は，n-grams の n をどのように設定するかである．n が小さいと，文章中のトークン(2-grams)の出現頻度を把握しやすいが，コンテクストを考慮する度合いは小さい．一方，n が大きいと，より広い範囲のコンテクストを考慮することが可能になるが，データ量が増大し，計算負荷も高くなる．ここでは，2-grams(bigram)分析のみを MPG に対しておこなう．

tidytext における n-grams 分析では，まず unnest_tokens()関数のオプションで(ngram および n=2)を指定し，連続する 2 単語を 1 つのトークンとしてトークン化する．Table 6 は MPG を bigram トークン化した tibble(MPG_bigram)の最初の 10 行を示している． MPG_bigram は，連続する 2 単語をスペースをはさんで結合したトークンを，bigram 列に格納している．

```
MPG_bigram <- tbl %>%
  unnest_tokens(bigram, text, token="ngrams", n=2) %>% filter(!is.na(bigram))
gt_mpg_bgr_tbl <- slice(MPG_bigram, 1:10) %>%
  rowid_to_column(var="id") %>% gt() %>%
  opt_table_font(font="Arial", size=px(10), weight="normal") %>%
  tab_header(title=md("Table 6. Bigram table for all issues"),
             subtitle=md("before processing stop words and word forms")) %>%
  tab_source_note(source_note=md("_Source_: Generated by authors.")) %>%
  tab_options(heading.title.font.size=px(12),
              heading.title.font.weight="normal",
              heading.subtitle.font.size=px(10),
              heading.subtitle.font.weight="normal",
              source_notes.font.size=px(8)) %>%
   opt_align_table_header(align="left") %>%
   cols_align(align="left", columns=everything()) %>%
   cols_align(align="left", columns=everything())
gt_mpg_bgr_tbl
```

Table 6. Bigram table for all issues. Before processing stop words and word forms

id	issue	line_num	bigram
1	GSSMP 2010	1	gssmp 2010
2	GSSMP 2010	2	medium term
3	GSSMP 2010	2	term monetary
4	GSSMP 2010	2	monetary policy
5	GSSMP 2010	2	policy principles
6	GSSMP 2010	3	over the
7	GSSMP 2010	3	the next
8	GSSMP 2010	3	next three
9	GSSMP 2010	3	three years
10	GSSMP 2010	3	years the

Source: Generated by authors.

続いて，bigram トークンを再び 2 つに分割する．つまり， bigram トークン内の単語の順序情報を tibble に保持したうえで bigram トークンを前後の単語に分ける．そのうえで，bigram トークンから，stop words を除去し，語形を統一する．以下のコードでは，1 つの bigram を構成する前方の単語を word1 列に，後方の単語を word2 列に格納した tibble をつくる．次に，word1 列と word2 列のそれぞれに stop word 除去と語形統一の処理をおこない，その結果を MPG_bigram_sep_fltrd 名の tibble に格納している．続いて，unite()関数を使って MPG_bigram_sep_fltrd の word1 列と word2 列の値を結合し，bigram 列に格納する．以上で，MPG_bigram_united は stop words を含まず，語形統一処理をした単語だけから構成される bigram のみを持つ．Table 7 は stop words 除去と語形統一処理をした後の MPG_bigram_united の最初の 10 行を示している．

```
MPG_bigram_sep_fltrd <- MPG_bigram %>%
  separate(bigram, into=c("word1", "word2"), sep=" ") %>%
  filter(!word1 %in% custom_stop_words$word) %>%
  filter(!word2 %in% custom_stop_words$word) %>%
  filter(!grepl(patterns_re, word1)) %>%
  filter(!grepl(patterns_re, word2)) %>%
  mutate(word1=str_replace_all(word1, replace_patterns),
         word2=str_replace_all(word2, replace_patterns))
MPG_bigram_united <- MPG_bigram_sep_fltrd %>%
  unite(bigram, word1, word2, sep=" ") %>%
  rowid_to_column(var="id")
gt_mpg_bgrunt_tbl <- slice(MPG_bigram_united, 1:10) %>% gt() %>%
  opt_table_font(font="Arial", size=px(10), weight="normal", style="normal",
                 color="black") %>%
  tab_header(title=md("Table 7. Bigram tokens table of all issues"),
             subtitle=md("after processing stop words and word forms.")) %>%
  tab_source_note(source_note=md("_Source_: Generated by authors.")) %>%
  tab_options(heading.title.font.size=px(12),
              heading.title.font.weight="normal",
              heading.subtitle.font.size=px(10),
              heading.subtitle.font.weight="normal",
              source_notes.font.size=px(8)) %>%
   opt_align_table_header(align="left") %>%
   cols_align(align="left", columns=everything()) %>%
   cols_align(align="left", columns=everything())
gt_mpg_bgrunt_tbl
```

Table 7. Bigram tokens table of all issues. After processing stop words and word forms.

id	issue	line_num	bigram
1	GSSMP 2010	3	reduce inflation
2	GSSMP 2010	3	anti inflationary
3	GSSMP 2010	3	inflationary emphasis
4	GSSMP 2010	3	emphasis efforts
5	GSSMP 2010	3	monetary regulation
6	GSSMP 2010	3	negative impact
7	GSSMP 2010	3	global financial
8	GSSMP 2010	4	government’s assessments
9	GSSMP 2010	4	domestic conditions
10	GSSMP 2010	4	increase emphasis

Source: Generated by authors.

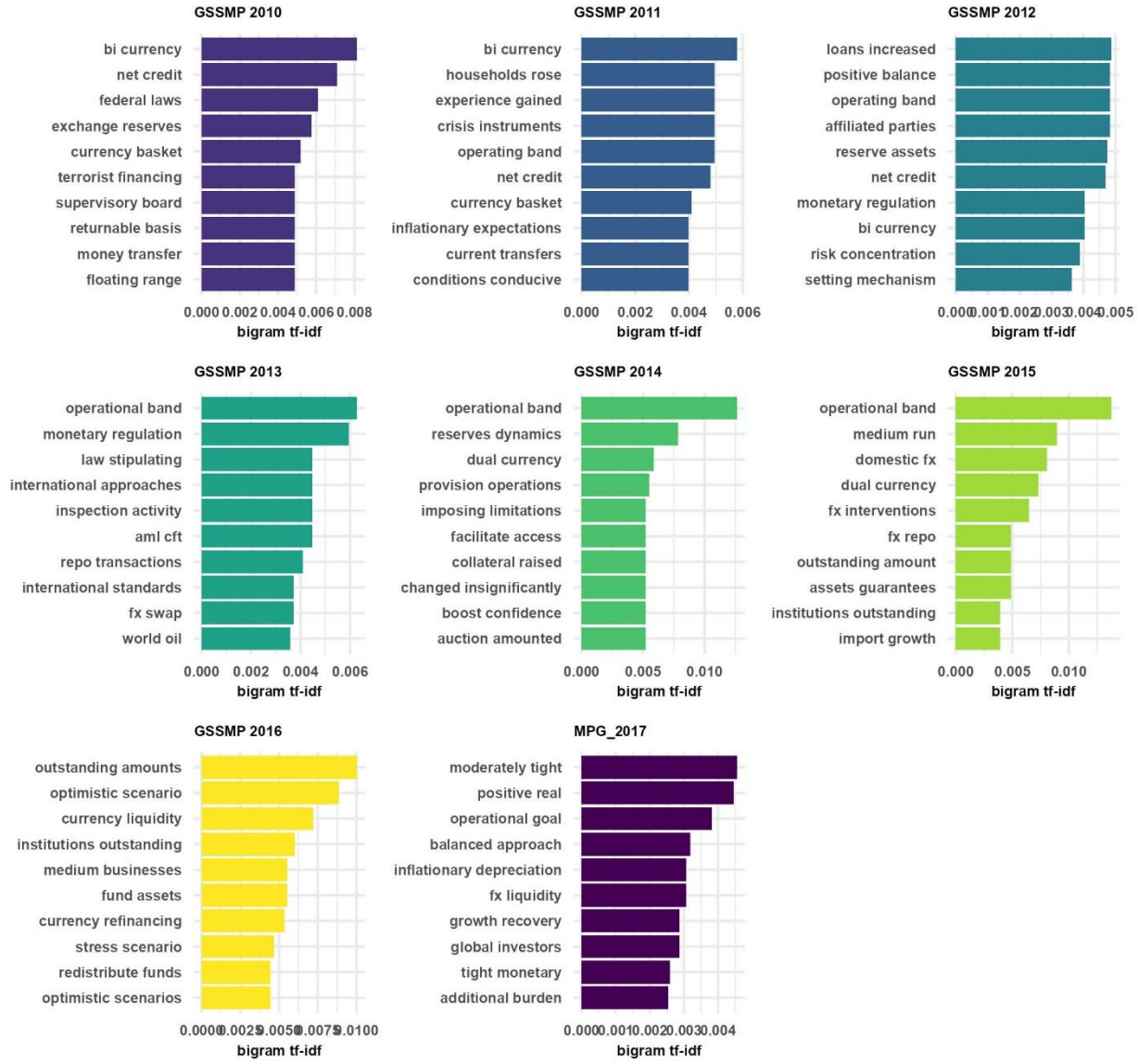
4.1 bigram トークンの出現頻度の分析

bigram トークンを準備すれば，単語トークンと同じ方法で bigram トークンの出現頻度を分析できる．Figure 6 (A) (B)は，MPG 各年版の TF-IDF 上位 10bigram トークンを示している．

```
MPG_bigram_tf_idf <- MPG_bigram_united %>% dplyr::count(issue, bigram) %>%
  bind_tf_idf(bigram, issue, n) %>% arrange(desc(tf_idf))
gg_bg_tfidf1 <- list()
gg_bg_tfidf2 <- list()
gg_bg_tfidf1_cmd <- list()
gg_bg_tfidf2_cmd <- list()
big_h <- ceiling(length(issue_list)/2)

for (i in seq_along(issue_list)) {
  j <- issue_list[[i]]
  isu_bg_tfidf <- MPG_bigram_tf_idf %>% dplyr::filter(issue==j) %>%
                    head(n=10) %>% arrange(desc(tf_idf))
  gbg <- ggplot(isu_bg_tfidf,
                aes(x=tf_idf, y=fct_reorder(bigram, tf_idf))) +
         theme_minimal()+
         geom_col(show.legend=FALSE, fill=vi8_cols[(i %% 8)+1]) +
         labs(title=j, x="bigram tf-idf", y=NULL)+
         theme(plot.title=element_text(face="bold",size=6),
               axis.title.x =element_text(face="bold",size=6),
               axis.title.y =element_text(face="bold",size=6),
               axis.text.x =element_text(face="bold",size=6),
               axis.text.y =element_text(face="bold",size=6)
               )
  if (i<=big_h){
    gg_bg_tfidf1[[i]] <- gbg
    gg_bg_tfidf1_cmd <- c(gg_bg_tfidf1_cmd,
                          paste0("gg_bg_tfidf1[[",i,"]]"))
  }else{
    gg_bg_tfidf2[[i]] <- gbg
    gg_bg_tfidf2_cmd <- c(gg_bg_tfidf2_cmd,
                          paste0("gg_bg_tfidf2[[",i,"]]"))
  }
  gbg <- ggplot() # initialize gbg
}

# make the comand string
gg_bg_tfidf1_eva <- stringr::str_c(gg_bg_tfidf1_cmd, collapse="+")
gg_bg_tfidf2_eva <- stringr::str_c(gg_bg_tfidf2_cmd, collapse="+")
# To adjust 8 plots to 9 panels (3 x 3)
gg_bg_tfidf1_eva <- paste0(gg_bg_tfidf1_eva,"+plot_spacer()")
gg_bg_tfidf2_eva <- paste0(gg_bg_tfidf2_eva,"+plot_spacer()")
gg_bg1<-eval(parse(text=gg_bg_tfidf1_eva)) +
  plot_layout(nrow=3, ncol=3) +
  plot_annotation(title="Figure 6 (A). Top 10 TF-IDF bigram tokens in 2010-17",
                  caption="Source: generated by authors.",
                  theme=theme(plot.title=element_text(size=8,
                              face="bold", color="black", hjust=0),
                              plot.caption=element_text(size=6, face="italic",
                                                        color="black", hjust=0)))
```

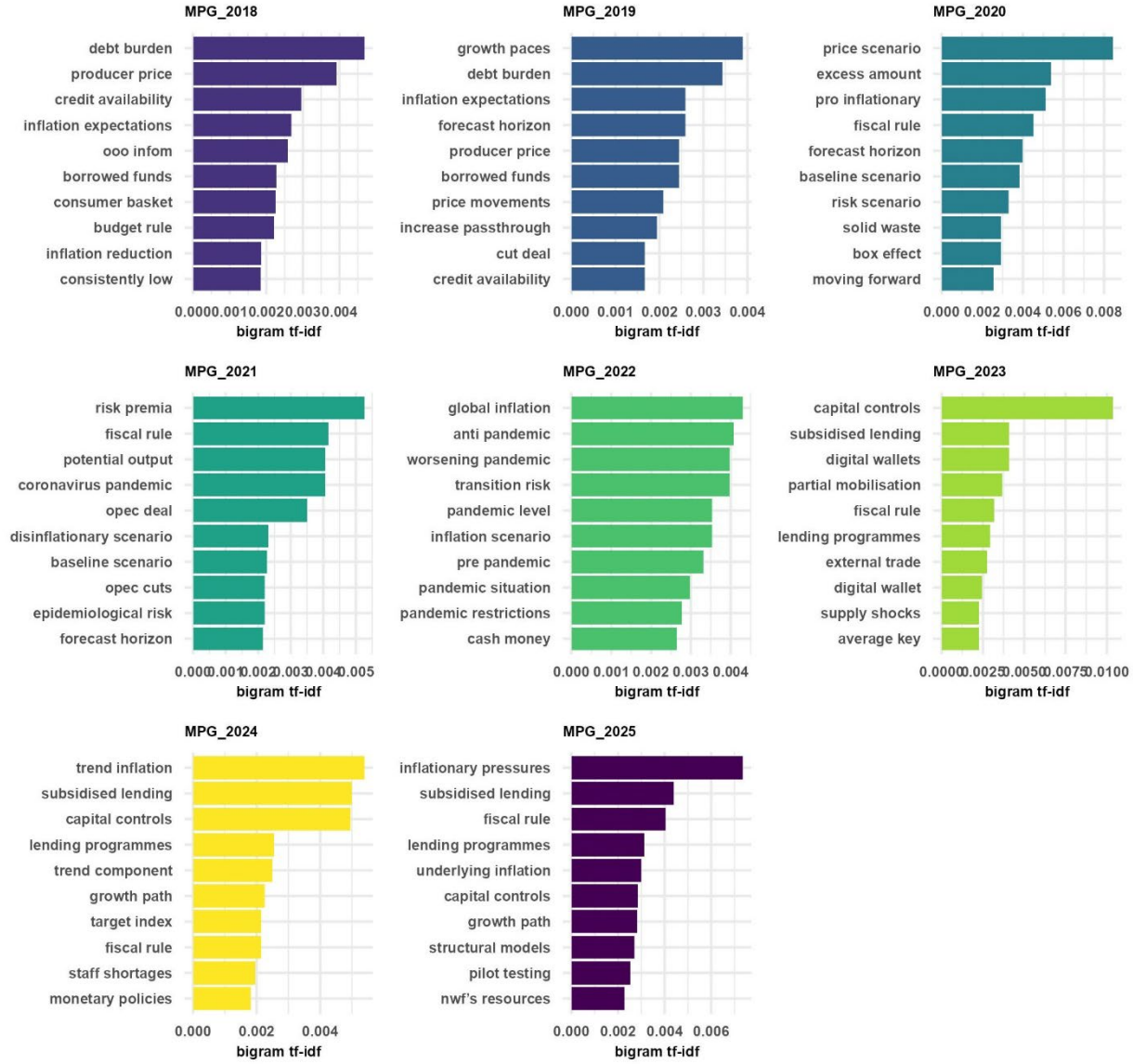
```
gg_bg2 <- eval(parse(text=gg_bg_tfidf2_eva)) +
  plot_layout(nrow=3, ncol=3) +
  plot_annotation(title="Figure 6 (B). Top 10 TF-IDF bigram tokens in 2018-25",
                  caption=md("Source: generated by authors."),
                  theme=theme(plot.title=element_text(size=8, face="bold",
                                                      color="black", hjust=0),
                              plot.caption=element_text(size=6, face="italic",
                                                        color="black", hjust=0)))
gg_bg1; gg_bg2
```

Figure 6 (A). Top 10 TF-IDF bigram tokens in 2010-17.



Source: generated by authors.

Figure 6 (B). Top 10 TF-IDF bigram tokens in 2018-25.



Source: generated by authors.

4.2 bigram トークンによる感情分析

bigram トークンを使うと，ある程度文脈を考慮した感情分析ができる．典型的な方法は，否定語に続く単語をみる方法である．否定語とそれに続く語を感情分析の対象とすると，それらの語を含む文書全体のトーンが一定程度判定できる．つまり，inflationary という単語そのものは多くの場合 negative な文脈で使われると思われるが，non inflationary であれば肯定的文脈で使われている場合が多いだろう．否定語とそれに続く語を組み合わせてみることで，それらを含む分のトーンがわかる．

ここでは，感情辞書として AFINN を使って bigram 感情分析をおこなう．AFINN は，単語の感情的なトーンを positive から negative へ 5 から-5 の値を付けて評価する感情辞書である．標準 stop words 辞書は否定語を含んでいるため，4.1 で用いた bigram トークンは否定語を含んでいない．したがって，否定語を排除していない bigram トークンを持つデータを作成する．標準 stop words 辞書から否定語

を除いたカスタム辞書を用意すれば，他の処理は 4.1 の処理と同様である．Table 8 は以上の処理をおこなった tibble である MPG_bigram_ng_sep_fltrd の最初の 10 行を表示している．

```
# stop_words including negations
negation_words <- c("isn't","wasn't","aren't","weren't", "don't", "doesn't",
"didn't", "not", "no", "non", "never", "without")
ctm_stop_words_ng <- custom_stop_words %>% filter(!word %in% negation_words)

MPG_bigram_ng_sep_fltrd <- MPG_bigram %>%
  separate(bigram, into=c("word1", "word2"), sep=" ") %>%
  filter(!word1 %in% ctm_stop_words_ng$word) %>%
  filter(!word2 %in% ctm_stop_words_ng$word) %>%
  filter(!grepl(patterns_re, word1)) %>%
  filter(!grepl(patterns_re, word2)) %>%
  mutate(word1=str_replace_all(word1, replace_patterns),
         word2=str_replace_all(word2, replace_patterns))

gt_ng_bg_tbl <- slice(MPG_bigram_ng_sep_fltrd, 1:10) %>%
  gt() %>%
  opt_table_font(font="Arial", size=px(10), weight="normal",style="normal") %>%
  tab_header(title="Table 8. Bigram tokens including negation words",
             subtitle="all issues after processing stop words and word forms.") %>%
  tab_source_note(source_note=md("_Source_: Generated by authors.")) %>%
  tab_options(heading.title.font.size=px(12),
              heading.title.font.weight="normal",
              heading.subtitle.font.size=px(10),
              heading.subtitle.font.weight="normal",
              source_notes.font.size=px(8)
             ) %>%
   opt_align_table_header(align="left") %>%
   cols_align(align="left", columns=everything())
gt_ng_bg_tbl
```

Table 8. Bigram tokens including negation words. All issues after processing stop words and word forms.

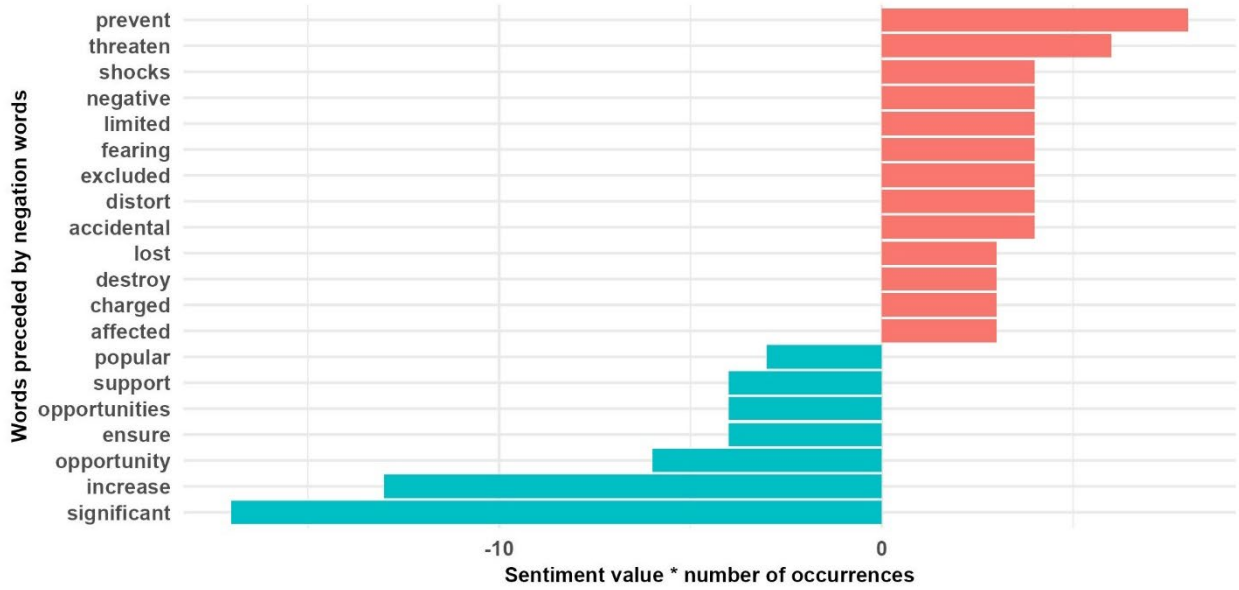
issue	line_num	word1	word2
GSSMP 2010	3	reduce	inflation
GSSMP 2010	3	anti	inflationary
GSSMP 2010	3	inflationary	emphasis
GSSMP 2010	3	emphasis	efforts
GSSMP 2010	3	monetary	regulation
GSSMP 2010	3	negative	impact
GSSMP 2010	3	global	financial
GSSMP 2010	4	government's	assessments
GSSMP 2010	4	domestic	conditions
GSSMP 2010	4	increase	emphasis

Source: Generated by authors.

否定語を含む bigram トークンと AFINN 感情辞書を使って感情分析をおこなう．Figure 8 は，否定語の次に来る単語の感情値とその単語の出現数との積を感情への貢献度とし，貢献度の絶対値が大きな 20 の bigram トークンを示している．positive な感情にもっとも大きく貢献した語は prevent の否定になっている．AFINN 感情辞書では prevent は negative な感情を持つ単語であるが，prevent が否定語の後に来ているため bigram 全体では positive な感情を持つトークンになる．

```
AFINN <- get_sentiments("afinn")
MPG_bigram_ng_sentiment <- MPG_bigram_ng_sep_fltrd %>%
  filter(word1 %in% negation_words) %>%
  inner_join(AFINN, by=c(word2="word")) %>%
  count(word2, value, sort=TRUE)

gg_ng_sentiment <- MPG_bigram_ng_sentiment %>%
  mutate(contribution=-(n * value)) %>%
  arrange(desc(abs(contribution))) %>%
  head(20) %>%
  mutate(word2=reorder(word2, contribution)) %>%
  ggplot(aes(-(n * value), word2, fill=n * value > 0)) +
  theme_minimal() +
  geom_col(show.legend=FALSE) +
  labs(title="Figure 7. Top 20 contributions of bigram to sentiment",
       subtitle="all issues.",
       caption="Note: bigrams those have large absolute values of
contributions.\nSource: generated by authors.",
       x="Sentiment value * number of occurrences",
       y="Words preceded by negation words") +
  theme(plot.title=element_text(face="bold",size=10, hjust=0),
        plot.subtitle=element_text(face="bold",size=8, hjust=0),
        plot.caption=element_text(face="italic",size=8, hjust=0),
        axis.title.x =element_text(face="bold",size=8),
        axis.title.y =element_text(face="bold",size=8),
        axis.text.x =element_text(face="bold",size=8),
        axis.text.y =element_text(face="bold",size=8))
gg_ng_sentiment
```

Figure 7. Top 20 contributions of bigrams to sentiment



Note: bigrams those have large absolute values of contributions.
Source: generated by atuhors.

Figure 7 は，興味深い分析内容を示すとともに，標準的な AFINN 感情辞書では MPG の感情分析をおこなうことの困難さがわかる．Figure 7 では，否定的な感情にもっとも貢献している bigram は significant を否定する内容の bigram である．AFINN 感情辞書は significant を positive な感情を持つ単語としているが，MPG では significant は単に統計的に有意といった意味で事実の叙述のためだけに使われている場合も多いであろう．否定的感情に 2 番目に大きく貢献している bigram は increase を否定する内容の bigram である．これも，AFINN 感情辞書が increase を positive な感情を示す単語としているからであるが，MPG では increase の否定が negative な内容を示しているとは言い難い．increase は，多くの場合は単なる事実の叙述に使われているだろう．あるいは，not increase の次の語が inflation であるような場合は，実際には positive なトーンであるというべきだろう．また，not increase の次の語が interest rate であれば，そこから文全体のトーンが positive なのか negative なのかを判定することは困難である．

さらに，MPG の分析結果をみると，否定語との bigram で AFINN により感情値が与えられている語は 44 種類しかなく，否定語との bigram の総数も 80 程度にしかならない．これは，公式政策文書の文章スタイルとして MPG が否定語をあまり使っていないためと考えられる．本格的に分析するにはこれらの点を考慮して分析を設計することが必要になる．

4.3 bigram のグラフ表示

bigram トークンを利用してある単語が次にどの単語とつながるかを分析できる．まず bigram を 2 単語に分解した状態で保持している MPG_bigram_ng_sep_fltrd から，各単語の列(word1，word2)に対して dplyr::count()関数により word1 のある単語と word2 のある単語の組がいくつあるかを計算し，その結果を新たな tibble (ここでは MPG_bigram_count) に格納する．MPG_bigram_count は，すべての word1 と word2 の組合せについて，word1 を起点ノード，word2 を終点ノードとするエッジの数(デフォルトの列名 n)を保持している．MPG_bigram_count のデータ形式は，igraph パッケージで用いるエッジリスト型データ構造に対応している(Holz, 2025)．Table 9 は，MPG_bigram_count の最初の 10 行を示している．

```
MPG_bigram_count <- MPG_bigram_sep_fltrd %>%
  count(word1, word2, sort=TRUE)
gt_bgcnt_tbl <- slice(MPG_bigram_count, 1:10) %>%
  gt() %>%
  opt_table_font(font="Arial", size=px(10), weight="normal", style="normal",
                 color="black") %>%
  tab_header(title=md("Table 9. Bigram graph data"),
             subtitle=md("all issues after processing stop words and word
forms.")) %>%
  tab_source_note(source_note=md("_Source_: Generated by authors.")) %>%
  tab_options(heading.title.font.size=px(12),
              heading.title.font.weight="normal",
              heading.subtitle.font.size=px(10),
              heading.subtitle.font.weight="normal",
              source_notes.font.size=px(8)) %>%
   opt_align_table_header(align="left") %>%
   cols_align(align="left", columns=everything()) %>%
   cols_align(align="left", columns=everything())
gt_bgcnt_tbl
```

Table 9. Bigram graph data. All issues after processing stop words and word forms.

word1	word2	n
inflation	expectations	886
credit	institutions	682
foreign	currency	573
inflation	target	449
transmission	mechanism	424
baseline	scenario	392
price	growth	359
foreign	exchange	343
inflation	targeting	335
financial	stability	330

Source: Generated by authors.

続いて，単語間の関係を MPG の出版年次ごとに可視化する．igraph パッケージを使って MPG_bigram_count をグラフ分析に適したデータ構造に変換し(Csardi and Nepusz, 2006)，ggraph パッケージで可視化する (Silge and Robinson, 2024)．ggraph は，ggplot2 の拡張パッケージであり，+演算子で ggplot2 の機能をそのまま使うことができる(Pedersen, 2024).

Figure 8 (A) (B) (C) (D)は，各年版の MPG における単語間のネットワークを示している．矢印は，矢印の始点の単語の次に矢印の終点の単語がくることを意味している．矢印の濃淡は，始点と終点という単語間の関係が観察される頻度が高いほど濃くなっている．MPG 各年版のグラフは単語間の関係が 10 回以下の関係は表示していない．Figure 9 は，同様に MPG 全体でみた単語間ネットワークを示している．Figure 9 では単語間の関係が 50 回以下の関係は表示していない．Figure 8，Figure 10 の両グラフは，省略した単語の関係があるため近似的ではあるが，単語間のマルコフ・チェーンを可視化している．

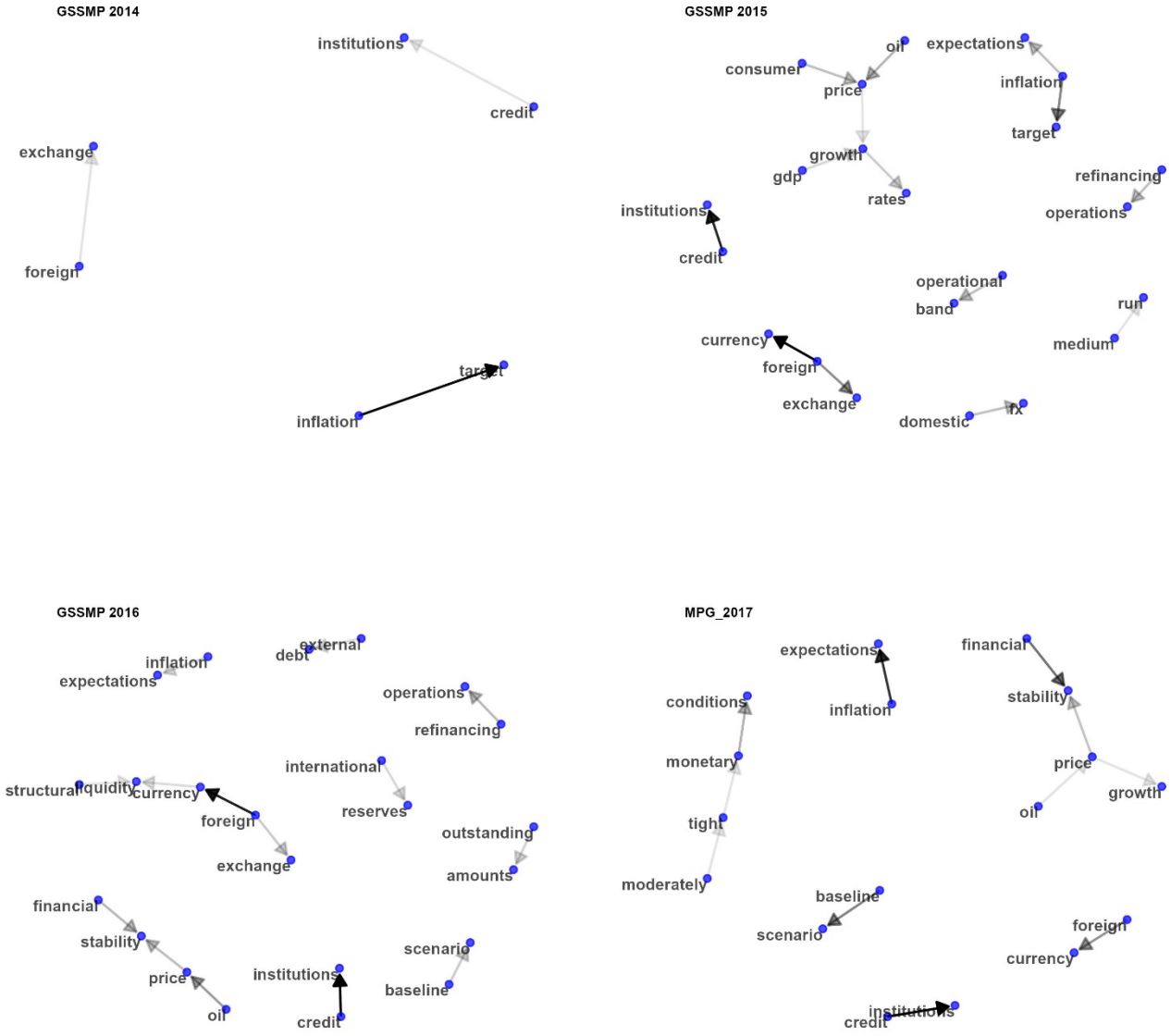
```
set.seed(42) # just for reproducibility
gg_bg_nw <- list()
gg_bg_nw1 <- list(); gg_bg_nw2 <- list(); gg_bg_nw3 <- list(); gg_bg_nw4 <- list()
gg_bg_nw1_cmd <- list(); gg_bg_nw2_cmd <- list(); gg_bg_nw3_cmd <- list()
gg_bg_nw4_cmd <- list()

# design the arrow in figures
a <- grid::arrow(type="closed", length=unit(.3, "cm"))
for (i in seq_along(issue_list)) {
  j <- issue_list[[i]]
  grph_tmp_tb <- MPG_bigram_sep_fltrd %>%
    filter(issue==j) %>% count(word1, word2, sort=TRUE) %>%
    filter( n>10 ) %>% igraph::graph_from_data_frame()

  gnw <- ggraph(grph_tmp_tb, layout="fr") +
    geom_edge_link(aes(edge_alpha=n), show.legend=FALSE,
                   arrow=a, end_cap=circle(.15, 'cm')) +
    geom_node_point(color="blue", alpha=0.7, size=2) +
    geom_node_text(aes(label=name), size=2, vjust=1, hjust=1) +
    labs(title=j) +
    theme_void() +
    coord_cartesian(clip="off") + # preventing labels from being cut-off
    theme(plot.margin=margin(0.5, 0.5, 0.5, 0.5, "cm")) +
    theme(plot.title=element_text(face="bold",size=6))
```

```
  if (i<=4){
    gg_bg_nw[[i]] <- gnw
    gg_bg_nw1_cmd <- c(gg_bg_nw1_cmd, paste0("gg_bg_nw[[",i,"]]"))
  } else if (i<=8){
    gg_bg_nw[[i]] <- gnw
    gg_bg_nw2_cmd <- c(gg_bg_nw2_cmd, paste0("gg_bg_nw[[",i,"]]"))
  } else if (i<=12){
    gg_bg_nw[[i]] <- gnw
    gg_bg_nw3_cmd <- c(gg_bg_nw3_cmd, paste0("gg_bg_nw[[",i,"]]"))
  } else {
    gg_bg_nw[[i]] <- gnw
    gg_bg_nw4_cmd <- c(gg_bg_nw4_cmd, paste0("gg_bg_nw[[",i,"]]"))
  }
  gnw <- ggplot()
}
# command string
gg_bg_nw1_eva <- stringr::str_c(gg_bg_nw1_cmd, collapse="+")
gg_bg_nw2_eva <- stringr::str_c(gg_bg_nw2_cmd, collapse="+")
gg_bg_nw3_eva <- stringr::str_c(gg_bg_nw3_cmd, collapse="+")
gg_bg_nw4_eva <- stringr::str_c(gg_bg_nw4_cmd, collapse="+")
gg_bgnw1<-eval(parse(text=gg_bg_nw1_eva)) +
  plot_layout(nrow=2, ncol=2) +
  plot_annotation(title="Figure 8 (A). Words Networks. 2010-13",
                  caption="Source: generated by authors.",
                  theme=theme(plot.title=element_text(size=8,
                                face="bold", color="black", hjust=0),
                              plot.caption=element_text(size=6,
                                face="italic", color="black", hjust=0)))
gg_bgnw2<-eval(parse(text=gg_bg_nw2_eva)) +
  plot_layout(nrow=2, ncol=2) +
  plot_annotation(title="Figure 8 (B). Words Networks. 2014-17",
                  caption="Source: generated by authors.",
                  theme=theme(plot.title=element_text(size=8,
                                face="bold", color="black", hjust=0),
                              plot.caption=element_text(size=6,
                                face="italic", color="black", hjust=0)))
gg_bgnw3<-eval(parse(text=gg_bg_nw3_eva)) +
  plot_layout(nrow=2, ncol=2) +
  plot_annotation(title="Figure 8 (C). Words Networks. 2018-21",
                  caption="Source: generated by authors.",
                  theme=theme(plot.title=element_text(size=8,
                                face="bold", color="black", hjust=0),
                              plot.caption=element_text(size=6,
                                face="italic", color="black", hjust=0)))
gg_bgnw4<-eval(parse(text=gg_bg_nw4_eva)) +
  plot_layout(nrow=2, ncol=2) +
  plot_annotation(title="Figure 8 (D). Words Networks. 2022-25",
                  caption="Source: generated by authors.",
                  theme=theme(plot.title=element_text(size=8,
                                face="bold", color="black", hjust=0),
                              plot.caption=element_text(size=6,
                                face="italic", color="black", hjust=0)))
gg_bgnw1; gg_bgnw2; gg_bgnw3; gg_bgnw4
```

Figure 8 (A) Words networks. 2010-2013.

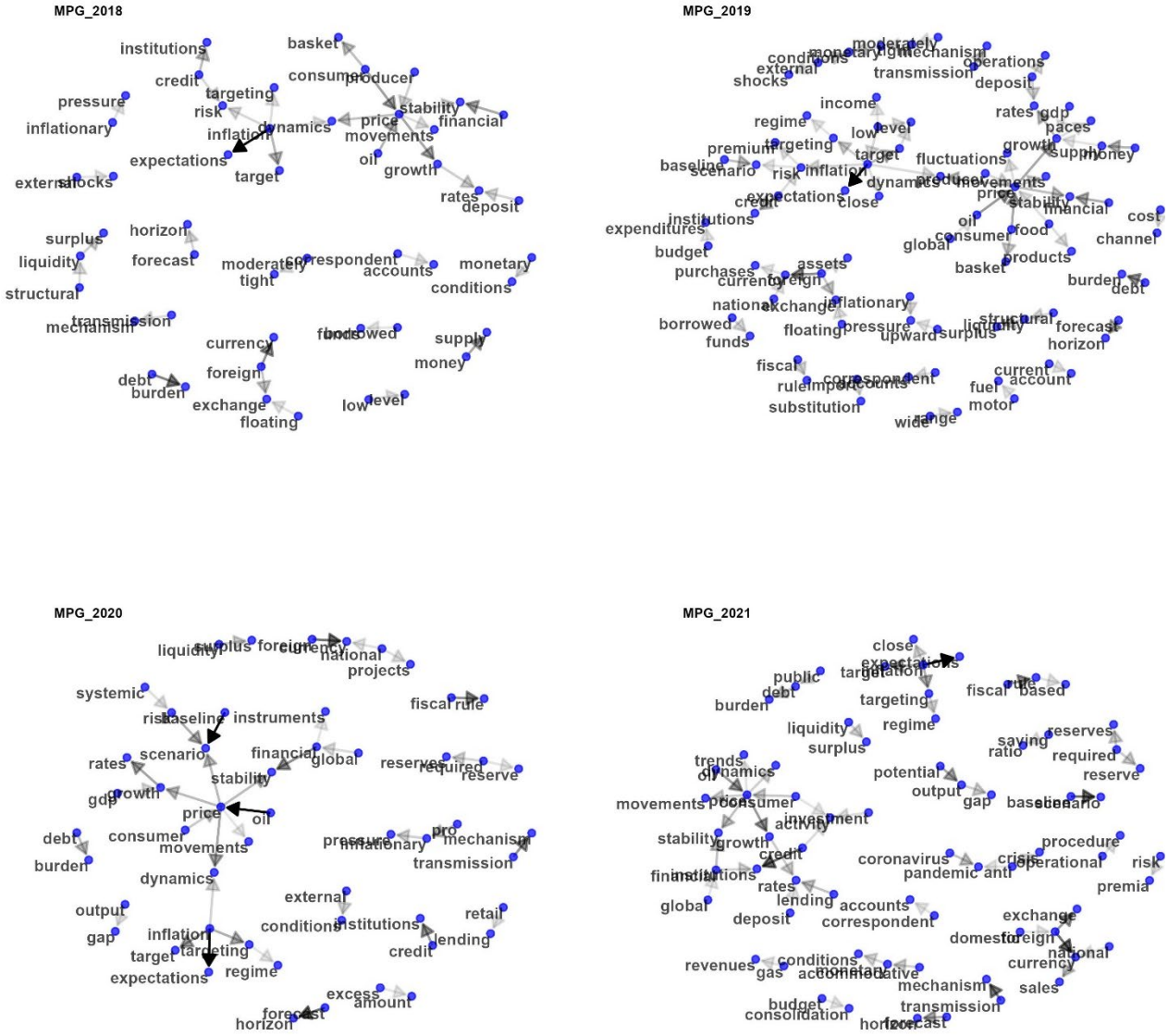
Source: generated by authors.

Figure 8 (B) Words networks. 2014-2017.



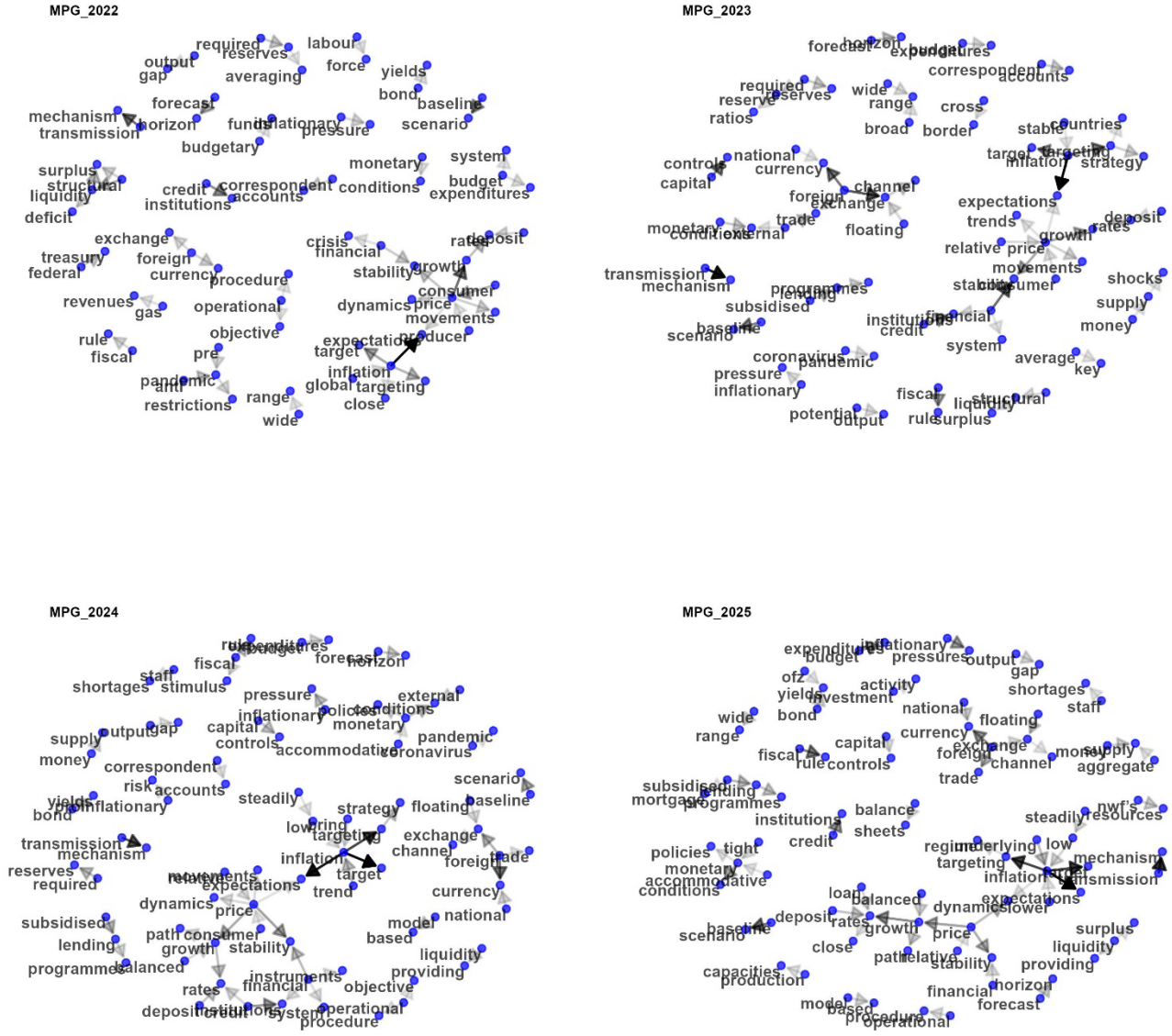
Source: generated by authors.

Figure 8 (C) Words networks. 2018-2021.



Source: generated by authors.

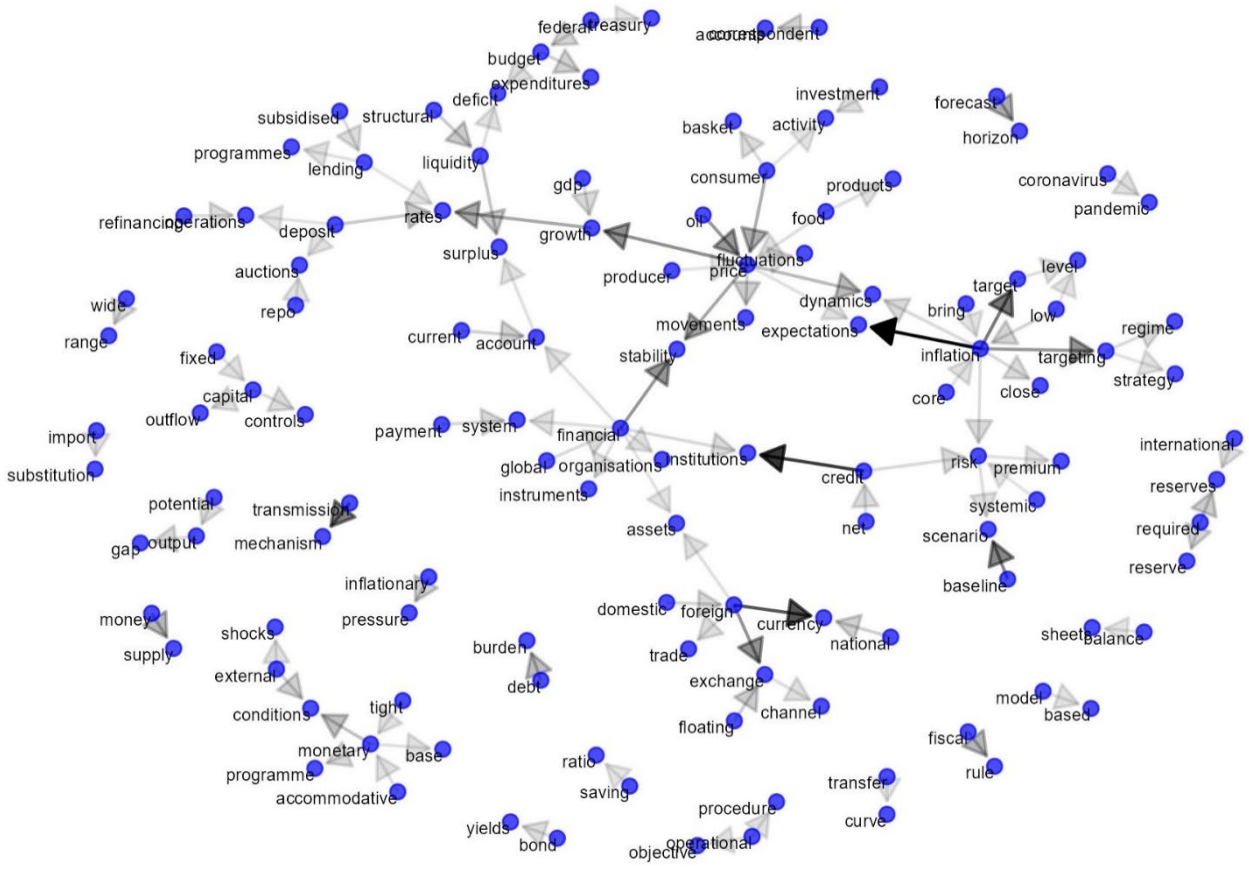
Figure 8 (D) Words networks. 2022-2025.



Source: generated by authors.

```
all_bg_nw <- MPG_bigram_sep_fltrd %>%
  count(word1, word2, sort=TRUE) %>%
  filter( n>50 ) %>%
  igraph::graph_from_data_frame() %>%
  ggraph(layout="fr") +
  geom_edge_link(aes(edge_alpha=n), show.legend=FALSE,
                     arrow=a, end_cap=circle(.15, 'cm')) +
  geom_node_point(color="blue", alpha=0.7, size=2) +
  geom_node_text(aes(label=name), size=3, vjust=1, hjust=1) +
  theme_void() +
  coord_cartesian(clip="off") + # preventing labels from being cut-off
  labs(title="Figure 9. Words network. All issues.",
       caption="Source: generated by authors.") +
  theme(plot.margin=margin(0.5, 0.5, 0.5, 0.5, "cm")) +
  theme(plot.title=element_text(size=10, face="bold", color="black", hjust=0),
        plot.caption=element_text(size=8, face="italic", color="black", hjust=0))
all_bg_nw
```

Figure 9. Words network. All issues.



Source: generated by authors.

Figure 8 と Figure 9 は内容的にも興味深い結果を示している．MPG において price と inflation は常に中心的な単語であり，price を中心とする単語のクラスターと inflation を中心とするクラスターがみられる．そのクラスターをみると，price は経済成長と関連性が強い一方，inflation は経済成長とはあ

まり関連がない．コロナ・パンデミックはトピックとしてあらわれている一方，クリミア併合やウクライナ戦争やそれらと直接関連するような sanction 等の単語はグラフ上にはあらわれない．クリミア併合やウクライナ戦争とそれにともなう経済制裁は，ロシアの経済活動と通貨政策にコロナ・パンデミック以上に大きな影響を与えるはずであるが，MPG がこれらに直接に言及することはほぼ無いことがわかる．2018 年には external shock，2019 年には import substitution がグラフ上にあらわれるものの，継続的なトピックスにはなっていない．変化を示唆するかもしれないものとしては，2023 年以降は，foreign exchange を中心とするクラスターが price，inflation を中心とするクラスターと並んであらわれてきたようにみえる．また，fiscal，budget, ofz (国債)といった国家予算関連の単語が目につくようになってきている．2024 年，2025 年には，stuff および subsidies と関連して shortage があらわれている．これらは，クリミア併合とウクライナ戦争にともなう経済状況の変化を間接的に反映しているかもしれないが，詳細を知るにはより本格的な分析が必要である．

なお，2018 年から word network のグラフが密になるが，これは MPG のボリュームが 2018 年版から増えたことによる．2017 年版までは 35-60 ページ程度だった MPG は，2018 年版が 130 ページを超え，それ以降 150 ページ前後になっている．テキスト分析では，このような外形的な文書スタイルの変化にも注意する必要がある．

Figure 8 と Figure 9 は，本試行分析の問題点も示している．簡単にみてとれるように，グラフは分析上意味がないと思われる単語間の関係もピックアップしている．例えば，balance と sheet や transmission と mechanism のように，MPG 中では事実上 1 単語であるものが含まれている．これらは 1 単語として事前処理して問題ないであろう．一方，inflation target や inflation expectation のように，1 単語に変換すべきかどうか現段階では判断が困難な単語もある．このような専門用語をどのように取り扱うべきかは，今後，多様な分野，国，時期の社会経済専門文献の分析経験の蓄積を通じて検討していくしかないように思われる．

これらのテキスト分析の技術的問題に加えて，MPG の試行分析結果はより本質的な問題もあきらかにした．クリミア併合，ウクライナ戦争が通貨政策に重大な影響を与えないとは考えられないが，MPG にはほぼ言及がない．現在のロシアのように多かれ少なかれ情報統制されている国では，テキスト分析によって社会経済研究の情報を得ることに本質的な限界がある．特定のバイアスをもった文献を対象にしてテキスト分析をおこなった結果は，やはり特定のバイアスを持つことになる．ここでも分析対象分野の専門知識がどれだけあるかが，テキスト分析によって収集した社会経済情報の質を決定することになる．

むすび

本研究では，ロシア中央銀行の公式刊行物である「通貨政策ガイドライン」の 2010 年版から 2025 年版を対象として標準的テキスト分析をおこなった．この研究の動機は，いわゆるグローバスサウスの社会経済分析用の情報を収集するためにテキスト分析手法を用いることのツールの有用性とツール利用上の問題点を評価することだった．本研究の結果，多国，多言語の社会経済分野の専門文献にテキスト分析を適用することによって情報を収集するために検討すべき一連の課題があきらかになった．それらの課題をまとめると次のようになる．

1. 分析対象となる文献の編集スタイル，全体構成とその時間的変化の確認と対応方法の決定．

2. 各文献中でテキスト分析の対象とすべき範囲の検討とそれに応じた前処理の検討．

3. 専門分野と各言語に応じた stop words のリストアップ.

4. 専門用語，特に合成語のリストアップとその取り扱い方法の検討.

5. 専門分野に対応した感情辞書の作成.

これらの課題に対処する 1 つの方法としてニューラルネットワークによる機械学習で対応することが考えられる．しかし，例えば今回の MPG では何を判断させ，何をもって教師とするのかという問題や専門文献であるためデータ量が限られるといった問題があり，機械学習方式の利用は容易ではないと思われる．これらの課題は，各専門分野の研究者とともに多国・多言語の専門文献のテキスト分析経験を蓄積していく中で解決していくしかないように思われる．

Reference

The Central Bank of the Russian Federation (MPG), 2009. *Guidelines for the Single State Monetary Policy in 2010 and for 2011 and 2012*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2010. *Guidelines for the Single State Monetary Policy in 2011 and for 2012 and 2013*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2011. *Guidelines for the Single State Monetary Policy in 2012 and for 2013 and 2014*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2012. *Guidelines for the Single State Monetary Policy in 2013 and for 2014 and 2015*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2013. *Guidelines for the Single State Monetary Policy in 2014 and for 2015 and 2016*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2014. *Guidelines for the Single State Monetary Policy in 2015 and for 2016 and 2017*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2015. *Guidelines for the Single State Monetary Policy in 2016 and for 2017 and 2018*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2016. *Monetary Policy Guidelines for 2017-2019*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2017. *Monetary Policy Guidelines for 2018-2020*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2018. *Monetary Policy Guidelines for 2019-2021*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2019. *Monetary Policy Guidelines for 2020-2022*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2020. *Monetary Policy Guidelines for 2021-2023*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2021. *Monetary Policy Guidelines for 2022-2024*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2022. *Monetary Policy Guidelines for 2023-2025*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2023. *Monetary Policy Guidelines for 2024-2026*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

The Central Bank of the Russian Federation (MPG), 2024. *Monetary Policy Guidelines for 2025-2027*, https://www.cbr.ru/eng/about_br/publ/ondkp/.

Csardi G, Nepusz T，2006. The igraph software package for complex network research, *InterJournal*, *Complex Systems*, 1695. https://igraph.org.

Graebner-Radkowitsch, Claudius, and Strunk, Birte, 2023. Degrowth and the Global South: The twin problem of global dependencies, Ecological Economics, *Ecological Economics*, 213, https://doi.org/10.1016/j.ecolecon.2023.107946.

Holtz, Yan, 2025. *Network chart with R and igraph from any type of input*, https://r-graph-gallery.com/257-input-formats-for-network-charts.html.

Pedersen, Thomas, 2024. *ggraph: An Implementation of Grammar of Graphics for Graphs and Networks*. R package version 2.2.1, https://ggraph.data-imaginist.com.

Silge, Julia and Robinson, David, 2025. *Text Mining with R: A Tidy Approach*, https://www.tidytextmining.com.

Wickham, Hadley, Çetinkaya-Rundel, Mine, and Grolemund, Garrett, 2025. *R for Data Science*, 2nd. ed., https://r4ds.hadley.nz/. Posit, 2025. *tidyr, tidy data*, https://tidyr.tidyverse.org/articles/tidy-data.html.

PwC コンサルティング合同会社(PWC), 2022.諸外国における産業政策の機軸, https://www.pwc.com/jp/ja/knowledge/thoughtleadership/2022/assets/pdf/industrial-policy.pdf.

R Core Team, 2025. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Appendix

R markdown の最後のコードチャンクは，利用したパッケージの名前とバージョンを記録した package_list.csv ファイルを生成している．R markdown の最初のコードチャンクは，package_list.csv を利用して必要なパッケージが存在しない場合は自動的にインストールするコードである．

R markdown のソースは MPG データ，package_list.csv とともに次にある: https://osi.org/ hxnqk/