Digitization of the Australian Parliamentary Debates (1901-2022)*

Lindsay Katz[†] Rohan Alexander[‡]

October 27, 2022

Public knowledge of what is said in parliament is a key tenet of democracy. Parliamentary dialogue is also critical for political science research. In Australia, following the UK tradition, the written record of what is said in parliament is known as Hansard. While the Australian Hansard has always been publicly available, it has been difficult to use it for the purpose of large-scale macro and micro-level text analysis because it has not been available as a dataset of sufficient quality to be credibly analysed with statistical models. Following the lead of the LiPaD project which achieved this for Canada, our project aims to provide a new, comprehensive, high-quality database that captures all proceedings of the Australian parliamentary debates from 1901 to the present day using Hansard. Our dataset will be publicly available and linked to other datasets such as election results. The creation of this dataset will enable the exploration of questions that are not currently possible to explore, serving as a valuable resource for both researchers and policymakers. This work will provide a thorough description of the creation and computational underpinnings of this database, followed by a discussion of some applications. To this point, we have completed this for 2011 to 2022.

1 Introduction

The official written record of parliamentary debates, formally known as Hansard, plays a fundamental role in capturing the history of political proceedings and facilitating the exploration of valuable research questions. Originating in the British parliament, the production of Hansard became tradition in a number of Commonwealth countries such as Canada, the United Kingdom, and Australia (Vice and Farrell 2017). Given the content and magnitude of these records,

[*] Code is available at: https://github.com/lindsaykatz/hansard-proj

[†] University of Toronto, lindsay.katz@mail.utoronto.ca

[‡] University of Toronto, rohan.alexander@utoronto.ca

they have significance, particularly in the context of political science research. In the case of Canada, a team of researchers at the University of Toronto have digitized Hansard from 1901 to 2019, an endeavor called the Linked Parliamentary Data (LiPaD) project (Beelen et al. 2017). Having a digitized version of Hansard enables researchers to perform advanced analyses on these records using text analysis tools and statistical modelling. Following the lead of the LiPaD project, in this paper we introduce a novel database for the Australian Hansard. This is composed of individual datasets for each sitting day in the House of Representatives, containing details on everything said in parliament in a form which can be readily used by researchers. With the development of tools for large-scale text analysis, this database will serve as a valuable resource for efficiently studying and exploring political behaviour in Australia over time.

The House of Representatives performs a number of crucial governmental functions, such as creating new laws and overseeing government expenditure (House of Representatives 2018, ch. 1). Sittings of the House follow a predefined order of business, regulated by procedural rules called standing orders (House of Representatives 2018, ch. 8). A typical sitting day in the Chamber has a number of scheduled proceedings including debates on government business, 90 second member statements, and Question Time (House of Representatives 2018, ch. 8). The Federation Chamber was created in 1994 as a subordinate debate venue of the Chamber. This allows for better time management of House business as its proceedings occur simultaneously with those of the Chamber (House of Representatives 2018, ch. 21). Sittings in the Federation Chamber are quite different to those of the Chamber in terms of their order of business and scope of discussion. Business matters discussed in the Federation Chamber are limited largely to intermediate stages of bill development, and the business of private Members (House of Representatives 2018, ch. 21). It is the recording and compilation of all of these proceedings on which Hansard is based, and it is essentially, but not entirely, verbatim.

A week or so after each sitting day, a transcript is available for download from the official Parliament of Australia website in both static PDF and extensible markup language (XML) form. The PDF is the official release, which is converted to typed text using Optical Character Recognition (OCR) technology (Sherratt 2016). This conversion allows for the production of the XML formatted transcript, though these are not always perfect conversions and are particularly flawed in the early decades of Hansard (Sherratt 2016). The PDF imposes formatting designed for humans to read with ease, whereas XML is designed for consistency and machine legibility. The nature of XML enables us to more easily use code to manipulate these records at scale, motivating our choice to develop our database solely using the XML formatted files. In cases where we were unsure on how to proceed with processing the XML, we defer first to the PDF, and then to the video recording of the proceeding if available.

The end goal for this work is to create a publicly available database of parsed Hansard transcripts from 1901 to present in a form which can be readily analyzed. More specifically, we aim to present each transcript in an ordered form with details on each speaker such as their political affiliation, the time stamp of their statement (where available), and whether they interjected (i.e. spoke out of turn). Thus far, this has been done for Hansard from mid-2011 to present.

We begin this paper with a description of what datasets for Australian Hansard already exist (Section 2). Next will be a detailed description of the contents of our database (Section 3), followed by a thorough discussion of how this database was created (Section 4).

2 Existing Datasets

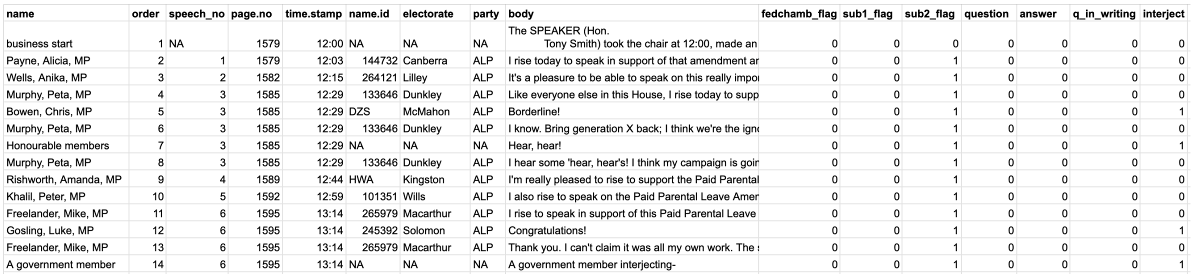
At present, the Hansard format that is available on the Parliament of Australia website is not accessible for large scale analysis. To this point, various researchers have had to create their own databases in an attempt to produce usable, complete Hansard data. For instance, Sherratt (2016) created an online, easy to read database of Hansard from 1901 to 1980 using the XML files. These data can be navigated by year, parliament, people, and bills (Sherratt 2016). As well, Leslie (2021) has released data on all Members of the Australian House of Representatives from 1945 to 2019, and is developing an `R` package for these data. Further, Alexander and Hodgetts (2021) created the `AustralianPoliticians` `R` package, which contains a number of datasets related to the political and biographical information of Australian federal politicians who were active between 1901 and 2021.

A number of papers exist which utilize components of Australian Hansard to explore various research topics. For example, Salisbury (2011) used online Hansard records to explore the occurrences of unparliamentary comments by members of the House, where The Speaker tells that member to withdraw their remark. Rasiah (2010) worked with Question Time data from Hansard transcripts during February and March of 2003, to investigate resistance of politicians in answering questions about Iraq. In their work, Fraussen, Graham, and Halpin (2018) use Hansard to quantify political prominence by investigating strategic mentions of interest groups by elected officials. Finally, the work of Alexander and Alexander (2021) offers a dataset of the Australian Hansard, along with an analysis of the effect of elections and changes in Prime Ministers upon topics mentioned in parliament. Alexander and Alexander (2021) created this database with the static PDF versions Hansard, using OCR to digitize these files into text which is suitable for analysis. This means there are considerable digitization errors especially in the first half of the dataset.

While there is evidently a growing body of literature on this topic, there is still no comprehensive database for Australian Hansard based on XML that spans from 1901 to the present day. Our work serves to bridge this gap.

3 Description of the Database

Our current database contains 670 comma-separated values (CSV) files, one for each sitting day of the House of Representatives from 10 May 2011 to 08 September 2022. We used two scripts to produce these files, which only have one subtle difference. Since the Federation Chamber was renamed in mid-2012 from the original title of Main Committee, the element under which all the Federation Chamber contents are nested in the XML was renamed as well (House of Representatives 2018, ch. 21). This difference in element name means that our original code for parsing Federation Chamber content would not work, as the specified node path would be incorrect. We accounted for this by creating a separate script with this name change. Our first script parses Hansard from 14 August 2012 to 08 September 2022, that being everything after the Federation Chamber name change, and the second script parses Hansard from 10 May 2011 to 28 June 2012. All output files in our database are formatted and processed in the exact same way. Figure 1 provides a snapshot of one of these files.



name	order	speech_no	page.no	time.stamp	name.id	electorate	party	body	fedchamb_flag	sub1_flag	sub2_flag	question	answer	q_in_writing	interject
business start	1	NA	1579	12:00	NA	NA	NA	The SPEAKER (Hon. Tony Smith) took the chair at 12:00, made an	0	0	0	0	0	0	0
Payne, Alicia, MP	2	1	1579	12:03	144732	Canberra	ALP	I rise today to speak in support of that amendment ar	0	0	1	0	0	0	0
Wells, Anika, MP	3	2	1582	12:15	264121	Lilley	ALP	It's a pleasure to be able to speak on this really impoı	0	0	1	0	0	0	0
Murphy, Peta, MP	4	3	1585	12:29	133646	Dunkley	ALP	Like everyone else in this House, I rise today to supp	0	0	1	0	0	0	0
Bowen, Chris, MP	5	3	1585	12:29	DZS	McMahon	ALP	Borderline!	0	0	1	0	0	0	1
Murphy, Peta, MP	6	3	1585	12:29	133646	Dunkley	ALP	I know. Bring generation X back; I think we're the ign	0	0	1	0	0	0	0
Honourable members	7	3	1585	12:29	NA	NA	NA	Hear, hear!	0	0	1	0	0	0	1
Murphy, Peta, MP	8	3	1585	12:29	133646	Dunkley	ALP	I hear some 'hear, hear's! I think my campaign is goir	0	0	1	0	0	0	0
Rishworth, Amanda, MP	9	4	1589	12:44	HWA	Kingston	ALP	I'm really pleased to rise to support the Paid Parental	0	0	1	0	0	0	0
Khalil, Peter, MP	10	5	1592	12:59	101351	Wills	ALP	I also rise to speak on the Paid Parental Leave Amer	0	0	1	0	0	0	0
Freelander, Mike, MP	11	6	1595	13:14	265979	Macarthur	ALP	I rise to speak in support of this Paid Parental Leave	0	0	1	0	0	0	0
Gosling, Luke, MP	12	6	1595	13:14	245392	Solomon	ALP	Congratulations!	0	0	1	0	0	0	1
Freelander, Mike, MP	13	6	1595	13:14	265979	Macarthur	ALP	Thank you. I can't claim it was all my own work. The :	0	0	1	0	0	0	0
A government member	14	6	1595	13:14	NA	NA	NA	A government member interjecting-	0	0	1	0	0	0	1

Figure 1: First 15 rows of CSV file for Hansard on 25 February 2020

Each row of the CSV file contains an individual statement, with the unique name ID, electorate, and party information of the speaker. Of course, for general statements such as "Honourable members", these variables cannot be specified. Each proceeding in the Chamber begins with the business start, where the time that the Speaker took the chair is documented, as well as the acknowledgment of the country and reading of the prayers as governed by the standing orders of the House (Department of the House of Representatives 2019, ch. 6). The Federation Chamber also begins with a business start, where it is stated at what time 5he Deputy Speaker takes the chair. For further clarification on each variable in the resulting CSV, Table 1 provides an overview of each variable in the database.

Table 1: Summary and description of variables in our database

Variable	Description
name	Name of speaker
order	Row number
speech_no	Speech number
page.no	Page number statement can be found on in official Hansard
time.stamp	Time of statement
name.id	Unique member identification code
electorate	Speaking member's electorate
party	Speaking member's party
body	Statement text
fedchamb_flag	Flag for Federation Chamber (1 if Federation Chamber, 0 if Chamber)
sub1_flag	Flag for sub-debate 1 contents (1 if sub-debate 1, 0 otherwise)
sub2_flag	Flag for sub-debate 2 contents (1 if sub-debate 2, 0 otherwise)
question	Flag for question (1 if question, 0 otherwise)
answer	Flag for answer (1 if answer, 0 otherwise)
q_in_writing	Flag for question in writing (1 if question in writing, 0 otherwise)
interject	Flag for interjection (1 if statement is an interjection, 0 otherwise)

The `name`, `page.no`, `time.stamp`, `name.id`, `electorate`, `party`, and `body` variables all came directly from the XML contents. In addition to these variables, we added a number of flags to enable easy filtering of statements. For example, adding the `fedchamb_flag` provides a clear distinction between the proceedings of the Chamber with those of the Federation Chamber. As well, the `sub1_flag` and `sub2_flag` variables allow us to keep track of where various statements are being parsed from in the XML document. Further details on this XML structure will follow in Section 4. The `question`, `answer`, and `q_in_writing` flags were added to identify statements belonging to Question Time, and the nature of these statements.[1] We also added a flag for interjections, which are defined as statements made during one members speech by another member who is not the Speaker or Deputy Speaker. Further, the `speech_no` variable allows us to keep track of the speech number that each statement and interjection belongs to. For example, in Figure 1, statements belonging to speech number 3 spans five rows, including two interjections. Having the speech number variable offers an easy way to group statements by speech or isolate specific speeches of interest. Finally, the `order` variable was added to keep track of the order in which everything was said, and was created after all statements and interjections were separated onto their own rows.

[1] More on this in Section 4.1

3.1 Descriptive Statistics

A visual exploration of our database offers interesting insight into general characteristics of Hansard. First, we looked at the total number of sitting days in the House of Representatives for each year from 2000 to 2021. Figure 2 illustrates that the number of House sitting days tends to fluctuate over time, where within this time span the years 2013 and 2019 had the fewest sitting days (45), and 2014 had the most sitting days (76).

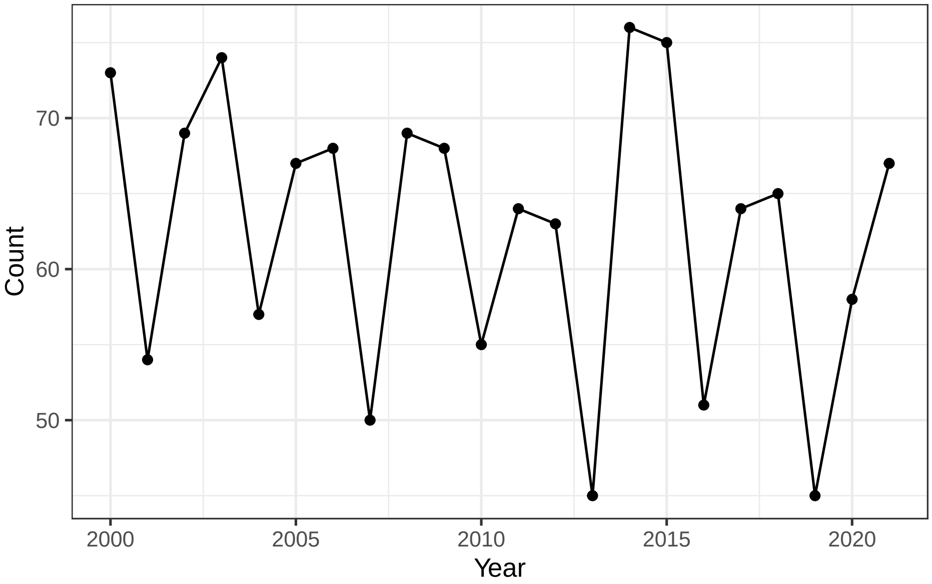


Figure 2: Total number of sitting days per year (2000-2021)

In Figure 3, we consider the average number of rows per sitting day per year, disaggregated by Chamber and Federation Chamber proceedings. We can see that 2014 has the highest average number of rows per day, and 2021 has the lowest average number of rows per day. There appears to be less fluctuation in the average number of rows per day in the Federation Chamber than there is in that of the Chamber.

Next we were interested in the percentage of rows in each Hansard which are flagged as interjections, using our definition of interjections mentioned above. We found that on average, approximately 18% of rows are characterized as interjections. Further, of all sitting days in our database, 25 June 2014 had the greatest percentage of rows being interjections, which was approximately 33%. The day with the smallest percentage of interjections was 23 March 2015, at approximately 3%. Based on Figure 4, there do not appear to be any significant fluctuations or patterns in the percentage of rows that are interjections across our time span.

Finally, Figure 5 summarizes the number of unique speakers, electorates, and parties found in Hansard over time. In all three panels, the data exhibit the same general pattern over time

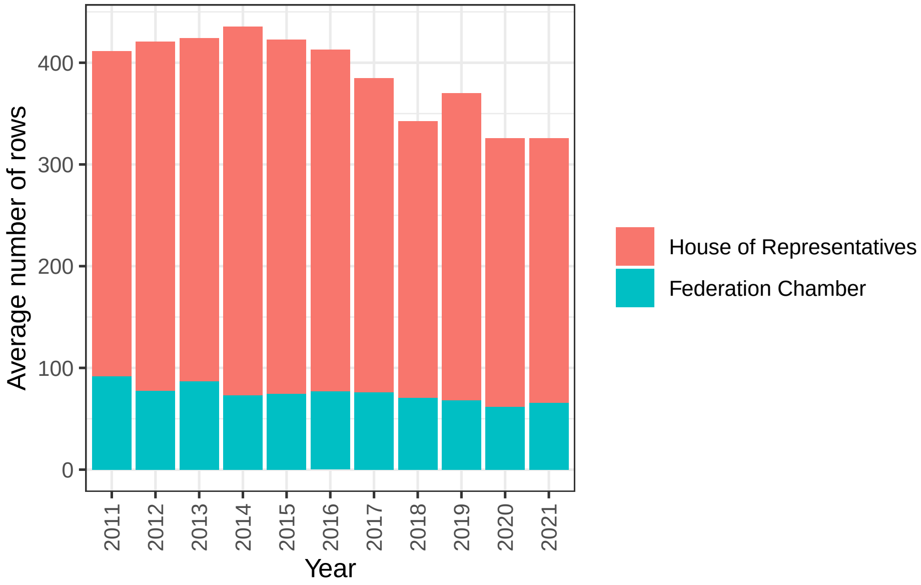


Figure 3: Average number of rows in Hansard per year, by Chamber and Federation Chamber

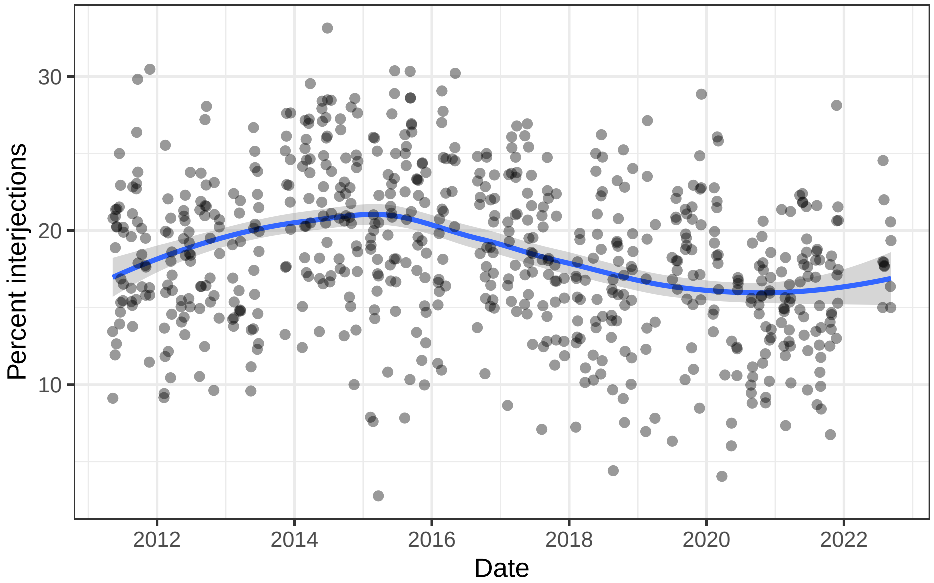


Figure 4: Percentage of rows in Hansard that are interjections, 10 May 2011 - 08 September 2022

for both the House of Representatives and the Federation Chamber, where the magnitude for the House data is consistently greater than that of the Federation Chamber. Panels A and B in Figure 5 illustrate that the number of unique names and electorates follow a very consistent trend across the entire time frame. Panel C shows that the number of unique parties has increased slightly since 2018.

4 Creation of the Database

The approach to parsing contents of an XML document is heavily dependent on its tree structure. As such, to create this database, we started by looking at a single Hansard XML transcript. Doing so enabled us to identify the various components of interest in the document, and how each one can be parsed according to its corresponding structural form. For example, the beginning of a speech may be followed by various interjections and continuations, all of which are structured as unique sub-elements containing further nested sub-elements within them. Parsing was performed in `R` using the `XML` package. Focusing on one transcript also allowed us to ensure that all key components of the transcript were being parsed and captured with as much detail as possible. Further, since we are looking at such a wide time span of documents, there are expectedly a number of changes in the way they are formatted. Some changes are as subtle as a differently named child-node, while others are as extensive as a completely different nesting structure. Smaller changes were accounted for as we became aware of them, and embedded into the final scripts in a way that would not cause issues for parsing more current Hansards with slightly different formatting. As mentioned in Section 1, the significant changes in XML structure of Hansard preceding 2011 necessitates much further development of our current scripts, which we are actively working to complete.

Figure 6 provides an example of the beginning of an XML file for Hansard. The XML structure begins with a root or parent element `"hansard"` (highlighted in blue), followed by a child element `"session.header"` (highlighted in yellow) with sub-child elements with information such as the date, and parliament number, which are all highlighted in pink. Next there is a child element containing everything that takes place in the Chamber, `"chamber.xscript"`, which is also highlighted in yellow in Figure 6. On sitting days where the Federation Chamber met as well, a child element exists for the Federation Chamber too. This allows for the distinction of proceedings of the Chamber and Federation Chamber. As mentioned in Section 3, each proceeding begins with the business start. The structure of this business start can be seen between the nodes highlighted in green in Figure 6. The content we parse from the business start are shown highlighted in orange in Figure 6. Next comes the debates and nested sub-debates, which follow the business start. Broadly, the speeches within Hansard transcripts are structured with a debate node, containing a sub-debate 1 child-node which has a sub-debate 2 child-node nested within it. That said, sometimes sub-debate 2 is not nested within sub-debate 1. Each of these three elements as well as their respective sub-elements contain

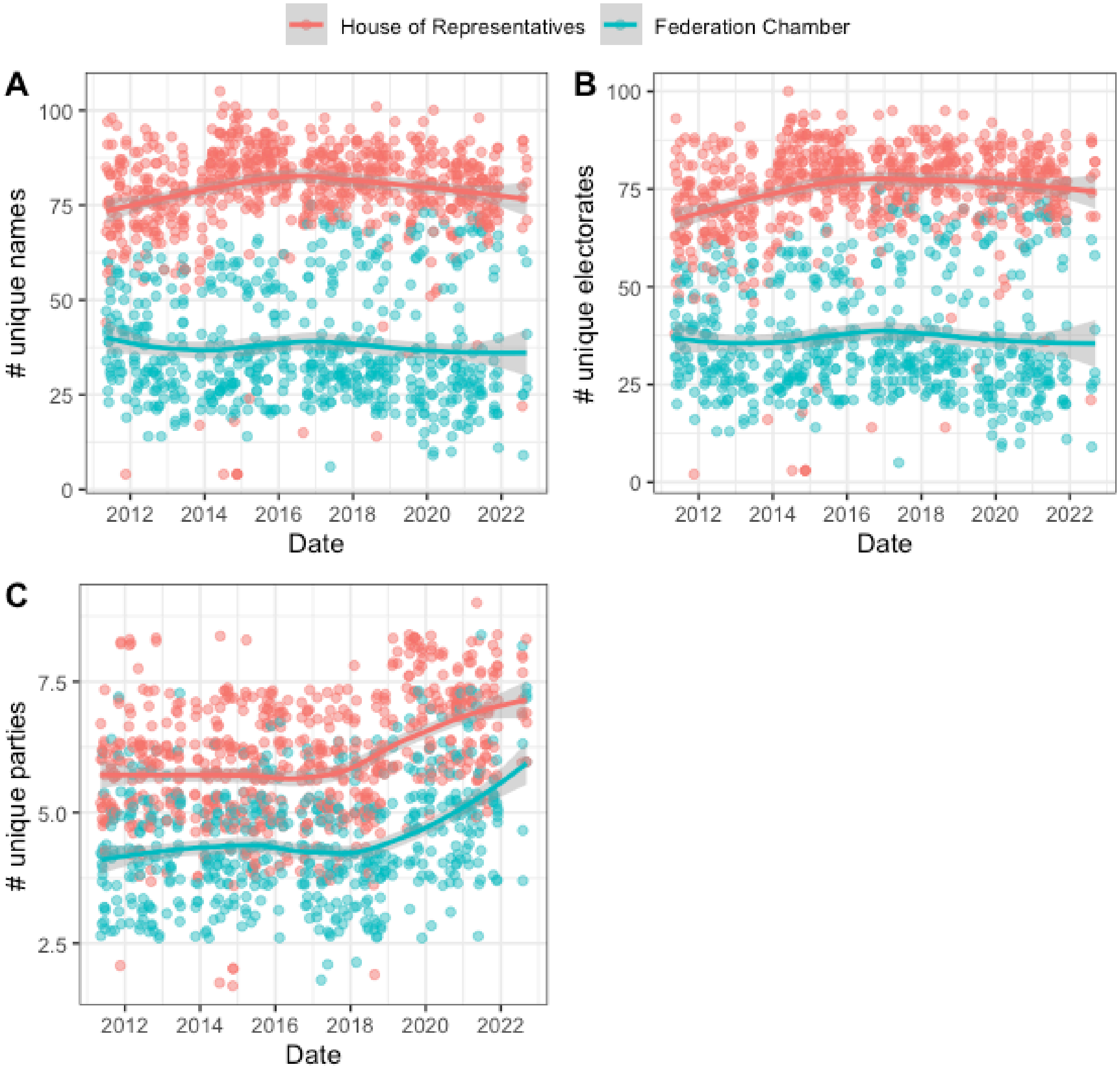
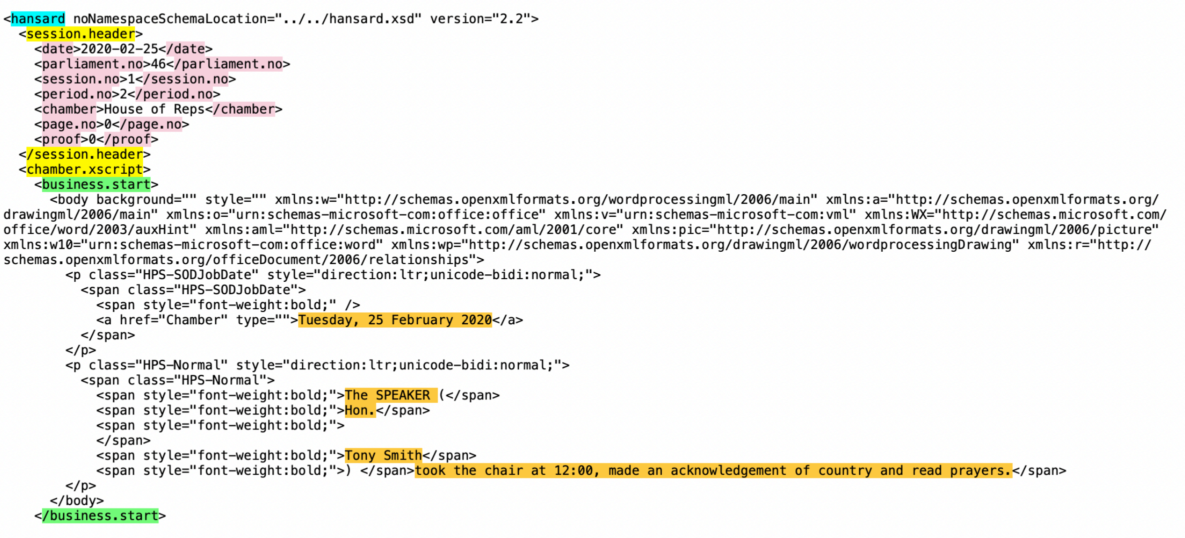


Figure 5: Number of unique names (A), electorates (B), and parties (C) found on each sitting day.

important information on the title of the debate or sub-debate, who is speaking, and what is being said. As mentioned, in many cases speeches are interrupted by interjections, which are embedded in these sub-elements as well. The final key distinct component of Hansard is Question Time, in which question and answer elements are classified differently than general debate text. Sometimes questions and answers are nested within debate or sub-debate child nodes, and other times they have their own child node. More detail on the processing of Question Time will follow in Section 4.1.



```xml
<hansard noNamespaceSchemaLocation="../../hansard.xsd" version="2.2">
  <session.header>
    <date>2020-02-25</date>
    <parliament.no>46</parliament.no>
    <session.no>1</session.no>
    <period.no>2</period.no>
    <chamber>House of Reps</chamber>
    <page.no>0</page.no>
    <proof>0</proof>
  </session.header>
  <chamber.xscript>
    <business.start>
      <body background="" style="" xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main" xmlns:a="http://schemas.openxmlformats.org/
drawingml/2006/main" xmlns:o="urn:schemas-microsoft-com:office:office" xmlns:v="urn:schemas-microsoft-com:vml" xmlns:WX="http://schemas.microsoft.com/
office/word/2003/auxHint" xmlns:aml="http://schemas.microsoft.com/aml/2001/core" xmlns:pic="http://schemas.openxmlformats.org/drawingml/2006/picture"
xmlns:w10="urn:schemas-microsoft-com:office:word" xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing" xmlns:r="http://
schemas.openxmlformats.org/officeDocument/2006/relationships">
        <p class="HPS-SODJobDate" style="direction:ltr;unicode-bidi:normal;">
          <span class="HPS-SODJobDate">
            <span style="font-weight:bold;" />
            <a href="Chamber" type="">Tuesday, 25 February 2020</a>
          </span>
        </p>
        <p class="HPS-Normal" style="direction:ltr;unicode-bidi:normal;">
          <span class="HPS-Normal">
            <span style="font-weight:bold;">The SPEAKER (</span>
            <span style="font-weight:bold;">Hon.</span>
            <span style="font-weight:bold;">
            </span>
            <span style="font-weight:bold;">Tony Smith</span>
            <span style="font-weight:bold;">) </span>took the chair at 12:00, made an acknowledgement of country and read prayers.</span>
        </p>
      </body>
    </business.start>
```

Figure 6: Snapshot of beginning of XML file for Hansard on 25 February 2020

Once code was written to parse all components, it was organized into individual scripts. The first script includes everything from the session header element, the second contains everything from the debate element, and the third and fourth contain everything from the sub-debate 1 and 2 elements, respectively. The fifth script contains debate interjection information, and the sixth contains content from Question Time. The final script is a compilation of everything. The next step was to further develop these scripts to produce tidy data sets from each parsed element, where each statement is separated onto its own row with details about the speaker, and rows are placed in chronological order. This first involved correcting the variable classes and adding a number of indicator variables to differentiate where statements came from, such as Chamber versus Federation Chamber or sub-debate 1 versus sub-debate 2. The next key task stemmed from the fact that the raw text data were not separated by each statement when parsed. In other words, any interjections, comments made by The Speaker or Deputy Speaker and continuations within an individual speech were all parsed together as a single string. As such, the name, name ID, electorate and party details were only provided for the person who's turn it was to speak. Splitting up these speeches in a way which would be generalizable across sitting days required much thought and effort. Section 4.2 will provide further details on the intricacies of this task.

When beginning to run these scripts on transcripts from other sitting days, it became clear that not every sitting day contains every possible XML element. For example, some days did not have sub-debate 2 content, and some days did not have a Federation Chamber proceeding. To improve the generalizability of these scripts, if-else statements were embedded within the code wherever an error might arise due to a missing element. For example, the entire Federation Chamber block of code is wrapped in an if-else statement for each script, so that it only executes if what the code attempts to parse actually exists in the file. Once the script ran without error for a few recent years of Hansard, we continued to work backwards until extensive changes in tree structure made our script incompatible with parsing earlier XML files. Specifically, the earliest date for which it works is 10 May 2011. Before writing a new script for parsing earlier Hansard, we decided to prioritize cleaning and finalizing what we have been able to parse. As such we continued building our scripts, fixing any issues we noticed in the resulting datasets, and splitting up any additional sections of the parsed text onto separate rows where necessary. Specifically, we added a section of our script to separate out general stage directions. More information on this separation will be provided in Section 4.3.

After completing the scripts, they were formatted as functions which take a file name argument. The final script which parses everything from Hansard produces one CSV file with the data on all proceedings for each sitting day (example shown in Figure 1).

4.1 Question Time

A key characteristic of the Australian parliament system is the ability for the executive government to be held accountable for their decisions. One core mechanism by which this is achieved is called Question Time. This is a period on each sitting day in the Chamber where members of the House can ask ministers two types of questions: questions in writing which are written in advance, or questions without notice which are asked verbally in the Chamber and are responded to in real time (House of Representatives 2021). Parsing the components of Question Time required a slightly different approach than that of the debate speech, because of its unique structure in the XML document. Sometimes, questions in writing are included directly in the `"chamber.xscript"` child node, with sub-child nodes called `"question"` and `"answer"` to differentiate the two. However, in other times, questions in writing are embedded in their own child node called `"answers.to.questions"` outside of `"chamber.xscript"`. Difficulty arose when trying to order these questions and answers correctly within the rest of the debate text, because unlike all other statements made which have both an associated page number and time stamp, these generally only have a page number. Time stamps have been a key variable for which we have arranged parsed text in chronological order, especially since many statements tend to be on the same page, meaning we cannot rely solely on page number to capture correct ordering. To approach this issue, we merged all parsed questions to be in a single dataframe, and all parsed answers in another dataframe. We then arranged each dataframe by page number, and merged the two dataframes in such a way that each question would be followed by its corresponding answer.

4.2 Interjections

As mentioned, the text was structured and parsed in such a way that various interjections and comments which happened during a speech were not separated onto individual rows. Below is a simple example extracted from Hansard on 30 November 2021, where Mr. van Manen is interrupted by The Speaker who states that the time for members' statements has concluded.

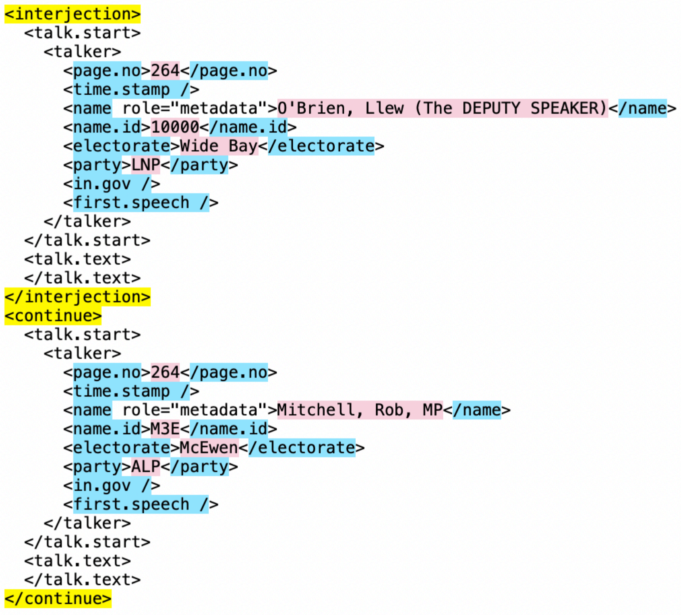
> "Mr VAN MANEN (Forde—Chief Government Whip) (13:59): It's a great pleasure to share with the House that Windaroo Valley State High School has qualified for the finals of the Australian Space Design Competition, to begin in January next year. The competition is regarded as the premier STEM competition for high school students and is recognised by universities around the country. The students are required to respond to industry-level engineering and requests for tender for design and—The SPEAKER: Order! In accordance with standing order 43, the time for members' statements has concluded."

As previously discussed, we wanted to have each statement on its own row, with the correct name, name ID, electorate and party information on the individual speaking. We approached this task in a number of steps. Once all parsed text from the XML was merged into one dataframe called `main`, we added a `"speech_no"` variable. As noted in Section 3, this was done to keep track of which speech each interjection, comment, or continuation belonged to as we separated these components onto their own rows.

The next step was to extract all the names and titles preceding these interjections, comments and continuations. This would enable us to then separate the speeches in the correct places using these names and titles in combination with regular expressions, which are patterns of characters that can be used to search bodies of text. We completed this extraction process with a few intermediate steps, due to the large number of name styles and interjection types that had to be accounted for, each requiring their own unique regular expression format.

In many of the XML files, below the full speech including all interjections and continuations, there is a series of child-nodes which exist to capture the structure of interruptions in that speech. Figure 7 provides an example of this, where the speech was interrupted by a comment from The Deputy Speaker, and then the member continued their speech. Looking at the elements highlighted in blue, it is clear that these child-nodes do not contain the actual text for the interjection or continuation. As mentioned, this text is embedded within the speech above it. However, as shown by the content highlighted in pink in Figure 7, we were able to extract useful details on the individual interjecting which we could use later on. Making use of this structure, we extracted names and information of all individuals that were categorized within the XML as interjections. We stored this as a dataframe called `"interject"`.[2]

[2] We decided not to include this data in our final database, as it is all embedded in our resulting datasets which have a flag for interjections.



```
<interjection>
  <talk.start>
    <talker>
      <page.no>264</page.no>
      <time.stamp />
      <name role="metadata">O'Brien, Llew (The DEPUTY SPEAKER)</name>
      <name.id>10000</name.id>
      <electorate>Wide Bay</electorate>
      <party>LNP</party>
      <in.gov />
      <first.speech />
    </talker>
  </talk.start>
  <talk.text>
  </talk.text>
</interjection>
<continue>
  <talk.start>
    <talker>
      <page.no>264</page.no>
      <time.stamp />
      <name role="metadata">Mitchell, Rob, MP</name>
      <name.id>M3E</name.id>
      <electorate>McEwen</electorate>
      <party>ALP</party>
      <in.gov />
      <first.speech />
    </talker>
  </talk.start>
  <talk.text>
  </talk.text>
</continue>
```

Figure 7: Snapshot of XML structure with interjection and continuation from 03 February 2021 Hansard

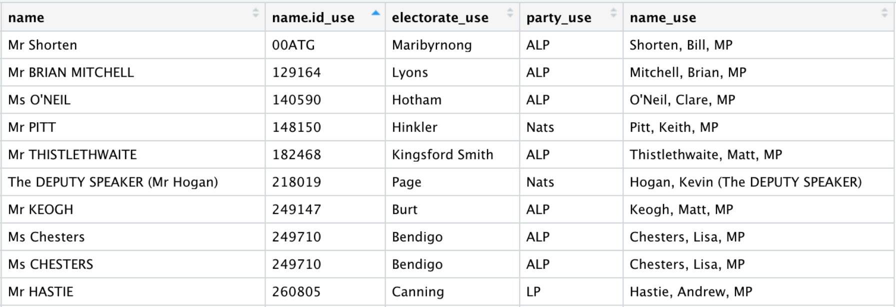
We then created lists using both the `interject` and `main` dataframes to capture all the names of individuals who spoke that day. We added the names of all Members in a number of unique formats, due to the frequent variation in how names are transcribed in Hansard. When a Member interjects or continues a speech, the usual form of their name is a prefix followed by their first name or first initial and/or last name. There is also variation in the capitalization of these names.[3] Another source of variation is in individuals with more than one first name, as sometimes only their first first name is written, while other times their entire first name is written. Additionally, some surnames have punctuation, and some surnames have specific capitalization such as "McCormack", where even in full capitalization, the first "c" remains lower case. This variation demands careful consideration when writing regular expression patterns to search for in the text. In these lists we also accounted for any general interjection statements transcribed that were not attributed to an individual, such as "An opposition member interjecting-".

Having these lists enabled us to extract the names of Members and their associated prefixes as they exist in the text, by searching for exact matches with regular expression patterns. We then used these extracted names to split all the speeches up, using a number of regular expressions with lookarounds. A lookaround can be added to a regular expression pattern to enhance the specificity of matches. These were used to ensure that the text was not being split

[3] Sometimes when someones first name is included, only their last name is capitalized, while sometimes their full name is capitalized, or other times neither are capitalized.

in the wrong places, such as places where Members were simply being named in the statement of another Member.

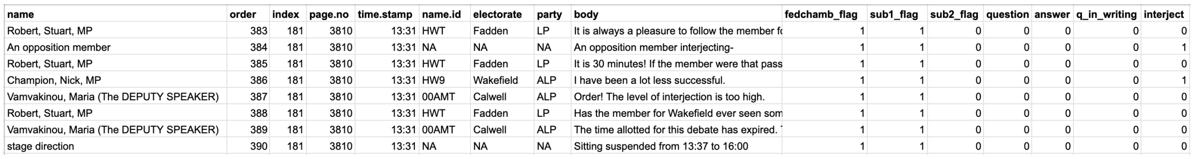
Once all interjections, comments and continuations were successfully split onto their own rows, we added a general order variable to the dataset based on row number, to keep track of the order in which everything was said. The next step was to fill the name, name ID, electorate and party variables with the correct data for each row. To do so, we created a lookup table, which contained the unique incomplete form in which the name was transcribed, and the corresponding full name, name ID, electorate, and party for that individual Figure 8. We used the main dataset from the `AustralianPoliticians` package in the creation of this lookup table (Alexander and Hodgetts 2021).



name	name.id_use	electorate_use	party_use	name_use
Mr Shorten	00ATG	Maribyrnong	ALP	Shorten, Bill, MP
Mr BRIAN MITCHELL	129164	Lyons	ALP	Mitchell, Brian, MP
Ms O'NEIL	140590	Hotham	ALP	O'Neil, Clare, MP
Mr PITT	148150	Hinkler	Nats	Pitt, Keith, MP
Mr THISTLETHWAITE	182468	Kingsford Smith	ALP	Thistlethwaite, Matt, MP
The DEPUTY SPEAKER (Mr Hogan)	218019	Page	Nats	Hogan, Kevin (The DEPUTY SPEAKER)
Mr KEOGH	249147	Burt	ALP	Keogh, Matt, MP
Ms Chesters	249710	Bendigo	ALP	Chesters, Lisa, MP
Ms CHESTERS	249710	Bendigo	ALP	Chesters, Lisa, MP
Mr HASTIE	260805	Canning	LP	Hastie, Andrew, MP

Figure 8: First 10 rows of lookup table from 10 May 2018 Hansard processing

Next, we merged our main dataframe with the lookup table to replace any incomplete names with their full names, and to fill in any gaps in the table with available name ID, electorate and party information. Finally, we were able to add a flag for interjections. Grouping our data by the speech number, we defined an interjection as a statement made by anyone who is not The Speaker, The Deputy Speaker, or the Member whose turn it was to speak. Figure 9 provides an example of a Federation Chamber proceeding with interjections. Evidently, statements made by the original speaker Stuart Robert or by The Deputy Speaker Maria Vamvakinou are not flagged as interjections.



name	order	index	page.no	time.stamp	name.id	electorate	party	body	fedchamb_flag	sub1_flag	sub2_flag	question	answer	q_in_writing	interject
Robert, Stuart, MP	383	181	3810	13:31	HWT	Fadden	LP	It is always a pleasure to follow the member f	1	1	0	0	0	0	0
An opposition member	384	181	3810	13:31	NA	NA	NA	An opposition member interjecting-	1	1	0	0	0	0	1
Robert, Stuart, MP	385	181	3810	13:31	HWT	Fadden	LP	It is 30 minutes! If the member were that pass	1	1	0	0	0	0	0
Champion, Nick, MP	386	181	3810	13:31	HW9	Wakefield	ALP	I have been a lot less successful.	1	1	0	0	0	0	1
Vamvakinou, Maria (The DEPUTY SPEAKER)	387	181	3810	13:31	00AMT	Calwell	ALP	Order! The level of interjection is too high.	1	1	0	0	0	0	0
Robert, Stuart, MP	388	181	3810	13:31	HWT	Fadden	LP	Has the member for Wakefield ever seen som	1	1	0	0	0	0	0
Vamvakinou, Maria (The DEPUTY SPEAKER)	389	181	3810	13:31	00AMT	Calwell	ALP	The time allotted for this debate has expired.	1	1	0	0	0	0	0
stage direction	390	181	3810	13:31	NA	NA	NA	Sitting suspended from 13:37 to 16:00	1	1	0	0	0	0	0

Figure 9: Example of speech with interjections from 21 November 2016 Hansard

4.3 Stage Directions

As mentioned, one of the final components added to our script was to separate general stage directions out from statements made by members. Stage directions are general statements included in the transcript to document happenings in parliament. Examples of stage directions are "Bill read a second time", "Question agreed to" or "Debate adjourned". It was unclear to us from the XML and PDF who exactly these statements were attributed to. For further clarification, we watched portions of the video recording for some sitting days, and noticed that where these statements are documented in Hansard, they are not explicitly stated in parliament. For example, when The Deputy Speaker says "The question is that the bill be now read a second time", members of the House take a vote, and if the majority is in favour, they proceed reading the bill the second time. This vote and second reading is not explicitly transcribed, rather what is written is: "Question agreed to. Bill read a second time". For this reason, we filled the name variable for these statements with "stage direction". Note that stage directions were not flagged as interjections. These stage directions are not defined differently from the regular debate speech in the XML, meaning we had to manually create a list of stage directions to separate out of the speeches. We have been building upon this list of stage directions as we work backwards in parsing Hansard.

5 Conclusion

Using a number of computational tools, we have created a novel database for Australian Hansard entirely based on XML data. Despite the public availability of Hansard, it is not functional in its current form for analysis at scale. Our work will enable individuals from various disciplines to perform broad spectrum research on Australian parliamentary proceedings without requiring a strong computational background.

Our database provides a CSV file for each sitting day of Hansard from 10 May 2011 to 08 September 2022. In our work, we have embedded a number of additional variables to the resulting datasets and accounted for variation in the way that proceedings were initially transcribed. This ensured that users of our database can easily filter through transcripts as needed to achieve their distinct research goal.

The vastness of our database and the significance of the content within it enables the exploration of an endless variety of research questions. For example, one might be interested in investigating the frequency of certain words or phrases, such as those relating to Covid-19. Another area of interest would be frequency of interjections, and the gender of individuals both being interrupted and doing the interrupting. This database would also act as a valuable resource for research using Natural Language Processing tools.

References

Alexander, Rohan, and Monica Alexander. 2021. "The Increased Effect of Elections and Changing Prime Ministers on Topics Discussed in the Australian Federal Parliament Between 1901 and 2018." *arXiv Preprint arXiv:2111.09299*.

Alexander, Rohan, and Paul A. Hodgetts. 2021. *AustralianPoliticians: Provides Datasets about Australian Politicians*. https://CRAN.R-project.org/package=AustralianPoliticians.

Beelen, Kaspar, Timothy Alberdingk Thijm, Christopher Cochrane, Kees Halvemaan, Graeme Hirst, Michael Kimmins, Sander Lijbrink, et al. 2017. "Digitization of the Canadian Parliamentary Debates." *Canadian Journal of Political Science/Revue Canadienne de Science Politique* 50 (3): 849–64.

Department of the House of Representatives. 2019. *Standing Orders as at 19 Septmeber 2019*. Commonwealth of Australia.

Fraussen, Bert, Timothy Graham, and Darren R Halpin. 2018. "Assessing the Prominence of Interest Groups in Parliament: A Supervised Machine Learning Approach." *The Journal of Legislative Studies* 24 (4): 450–74.

House of Representatives. 2018. Edited by D. R. Elder and P. E. Fowler. *House of Representatives Practice*. 7th ed. Australian Government - Department of the House of Representatives.

———. 2021. *A window on the house: Practices and procedures relating to question time*. Parliament of Australia.

Leslie, Patrick. 2021. *Members of the Australian House of Representatives (1945-2019)*. https://github.com/palesl/AustralianHouseOfRepresentatives.

Rasiah, Parameswary. 2010. "A Framework for the Systematic Analysis of Evasion in Parliamentary Discourse." *Journal of Pragmatics* 42 (3): 664–80.

Salisbury, Christopher. 2011. "'Mr Speaker, i Withdraw...': Standards of (Mis) Behaviour in the Queensland, Western Australian and Commonwealth Parliaments Compared via Online Hansard." *Australasian Parliamentary Review* 26 (1): 166–77.

Sherratt, Tim. 2016. "Documentation: Historic Hansard." *Historic Hansard*. http://timsherratt.org/digital-heritage-handbook/docs/historic-hansard/#.

Vice, John, and Stephen Farrell. 2017. *The History of Hansard*. House of Lords Library; House of Lords Hansard.