Hansard Data Mining

Final report

Aaron Butler

Bipin Karki

Katherine Noack

Mahmoud Yousefi

November 1, 2019

Contents

[Acknowledgements 3](#_Toc23417274)

[Executive Summary 3](#_Toc23417275)

[Introduction 4](#_Toc23417276)

[Background 4](#_Toc23417277)

[Project Objectives 4](#_Toc23417278)

[Scope 4](#_Toc23417279)

[Deliverables 6](#_Toc23417280)

[Project Methodology 6](#_Toc23417281)

[Project Plan and Alterations 7](#_Toc23417282)

[Comparison of Original and Updated Project Plan 7](#_Toc23417283)

[Task Changes 8](#_Toc23417284)

[Deliverables 8](#_Toc23417285)

[Project Architecture 8](#_Toc23417286)

[Extract, Transform and Load (ETL) Process 10](#_Toc23417287)

[Dashboards 14](#_Toc23417288)

[Text Analytics 19](#_Toc23417289)

[Project Performance Assessment 26](#_Toc23417290)

[Performance against critical success factors 26](#_Toc23417291)

[Performance against schedule 27](#_Toc23417292)

[Performance against outputs 27](#_Toc23417293)

[Lessons Learned 27](#_Toc23417294)

[Recommendations and Future Work 28](#_Toc23417295)

[Conclusion 29](#_Toc23417296)

[References 30](#_Toc23417297)

[Appendix A: Decisions Made 32](#_Toc23417298)

[Appendix B: Database Schema 34](#_Toc23417299)

[Appendix C: Data Dictionary 35](#_Toc23417300)

[Appendix D: Hansard XML Record Structure 39](#_Toc23417301)

[Appendix E: Dashboard Wireframe Mockups 40](#_Toc23417302)

[Appendix F: Dashboard Charts 41](#_Toc23417303)

[Appendix G: Document Summary Example 45](#_Toc23417304)

[Appendix H: SSIS - Visual Basic script to read in the file names 46](#_Toc23417305)

[Appendix I: T-SQL scrips to check if the file name exists in DB (SSIS) 47](#_Toc23417306)

[Appendix J: SQL Command to truncate the final tables in SSIs 48](#_Toc23417307)

[Appendix K: SQL Command to join tables to get the HansardID in SSIS 49](#_Toc23417308)

[Appendix L: SSIS Variable Table 50](#_Toc23417309)

[Appendix M: List of Topics Generated by LDA 51](#_Toc23417310)

[Appendix N: Gantt Chart 52](#_Toc23417311)

# Acknowledgements

We would like to express our deepest appreciation to everyone involved in this project. A special gratitude to our clients Andrew Corrigan, James Baker, Aaron Steicke and all members of the Project Reference Group from the Auditor-General’s Department for providing regular feedback and an extraordinary working environment.

Furthermore, we would like to acknowledge with much appreciation the crucial role of our mentor Eric Lam for his support, comment, advice and all the effort he put into this project. We also want to thank our course coordinators for their support and guidelines about the nature of the project and its deliverables. Last but not least, we would like to thank the University of South Australia for providing this opportunity to work on a real-world scenario.

# Executive Summary

The Hansard Data Mining project is focused on providing the Auditor-General's Department (AGD) staff with an automated system to better understand, interrogate and summarise text data from South Australia Parliament discussions. The Hansard website keeps the official records of debates in an XML format which is the primary source of data for this project (Parliament South Australia, 2019).

The project was built and tested upon two years of data extracted from the Hansard website for three different proceeding types: “Bills”, “Question Time” and "Answers to Questions”. To better understand these records, multiple text analytics techniques were implemented such as text summarisation, sentiment analysis, and topic modelling. Several visualisation techniques were also used to create dashboards for AGD. However, several text analytics, such as topic modelling and word tree, were developed as a proof-of-concept and were not included in the dashboards.

Five dashboards were developed using Tableau Desktop and are currently being used by AGD to visualise trending discussion topics and to identify relevant Hansard records for their audits. The dashboards also allow users to view records that mention 338 different AGD clients. AGD was also provided with an Installation and Deployment guide that contains information on how to setup and run the automated processes, code, and dashboard(s) on their own systems and a user guide briefly describing how to use the dashboards.

This project will save AGD employees time and better interrogate parliament text compared to the current manual search used by AGD audit teams. Additionally, AGD intends to publish the dashboards for use by the entire department. This entire project from scraping Hansard records to visualisation has been deployed to AGD systems and is run automatically every seven days.

# Introduction

The Auditor-General’s Department (AGD) contributes to public sector accountability in South Australia by providing independent assurance to the Parliament that government activities are conducted and accounted for properly and in accordance with the law (AGD, 2019). As part of their responsibilities AGD conducts and reports on special audits which requires reading through Hansard records to find relevant topics. To assist AGD in quickly identifying relevant information for their audits, several proof-of-concept dashboards were developed that allow staff to better interrogate and summarise Parliament discussions.

This report summarises the project background, methodology, deliverables, project performance assessment, and lessons learned for the Hansard Data Mining Project and concludes with several recommendations.

# Background

## Project Objectives

The objective of this project is to develop a proof-of-concept dashboard that will allow AGD to analyse unstructured data sources, focusing on Hansard records from South Australia’s Parliament. Hansard is a division of South Australia’s parliament that keeps the official record of Parliamentary debates and questions (Hansard, 2008). The main project outcomes identified by AGD was to:

* Consult with relevant stakeholders (including audit teams and executives) to confirm business requirements for Hansard analysis
* Develop a process to load Hansard records to a central database
* Develop a dashboard or other outputs to allow staff to search and summarise topics
* Update the dashboard based on iterative feedback, with suitable documentation and recommendations for future features developed as part of the handover

## Scope

This section describes what was considered in and out of scope for the project, and accepted scope changes. The project scope was initially described in the project plan and was created in collaboration with the project sponsor and project owners at AGD.

#### Inclusions

The following aspects of the project are considered within scope:

* Identification and analysis of customer requirements
* Build an automatic web scraping process to extract Parliament discussions from Hansard webpage
* Processing of text data scraped from Hansard webpage into a format suitable for storage in a database and later analysis.
* Manage the storage of the scraped data in a database
* Data analysed will be at least one year of Hansard Parliament Discussions
* Exploratory approaches for initial data analysis and visualisations
* Background research on text analysis, dashboard design and visualisation methods
* Text analysis of the collected data
* Search functionality for clients and topics over collected data
* Building the visualisation dashboard using a third-party application (such as Tableau Desktop)
* The project will be developed in R or Python, making use of existing packages and libraries
* Version control for the project will be maintained in GitHub
* Documentation of deliverables, including instructions on the deployment of the process to scrape, store and visualise Hansard data
* Process and products produced will be deployable by Auditor-General's Department

#### Exclusions

The following aspects of the project are explicitly out of scope:

* Advanced text analytics techniques. Due to time constraints, advanced techniques cannot be adequately developed or tested.
* Data sources outside of those listed in scope
* Predictions or forecasting from scraped data
* Web application development
* Evaluation of third-party software such as Tableau
* Evaluation of Hansard data accuracy. Due to time constraints this project will not investigate the accuracy of Hansard transcripts compared to the original discussions.

#### Approved Scope Changes

There were several approved changes to the scope described previously such as sentiment analysis. Initially this approach was excluded due to the limited time available to complete the project. However, due to some parts of the project being completed ahead of schedule, sentiment analysis was implemented and included in the dashboards as a proof-of-concept. Sentiment analysis is described in further detail in the Sentiment Analysis section.

Another approved scope change was to provide a short user guide for the dashboards. This was required due to the increasing number of features available in the dashboards and the use of regular expressions in the text search feature. Users cannot be expected to know or memorise regular expressions and therefore their use in text search required user documentation. Therefore, a list of common regular expressions, and examples of their use in dashboard text search, were provided in a user guide.

At the request of the Project Reference Group the reasons why certain decisions were made was documented in a table (Appendix A: Decisions Made) so that after the project concluded stakeholders could understand why these decisions were made. Additionally, one of the deliverables listed in the project plan was project closure documents to handover the project to the client. On consultation with the project owners at AGD it was determined that a Project Review and Closure report was not required. This report, an Installation and Deployment Guide and user guide was determined to be sufficient documentation for the project.

## Deliverables

The deliverables produced are consistent with the stated scope and deliverables in the project plan. These deliverables are described in more detail in the Deliverables section.

|  |  |
| --- | --- |
| Deliverable | Description |
| Installation and Deployment Guide | Contains instructions on how to setup and run the provided code, database, and dashboards on AGD systems. |
| User Guide | Contains an overview of dashboard features and common regular expressions that can be used for text search. |
| Decisions Made Documentation | Documentation of decisions made to improve stakeholder understanding post-project (Appendix A: Decisions Made). |
| Risk Register | Spreadsheet of project risks and their likelihood, potential impact, and mitigating actions. |
| Automated data scraping process | Automated process to scrape Hansard records from webpage using Python programming language. |
| Automated SSIS Process | Automated SSIS process to process scraped data and store in Hansard database. The SSIS process also triggers the data scraping weekly. |
| Two years of Hansard records in database | The last two years of Hansard records for Bills, Question Time, and Answers to Questions are stored in a SQL Server database. The database schema is shown in Appendix B: Database Schema and data dictionary In Appendix C: Data Dictionary. |
| Dashboards | Five dashboards providing visualisations, metrics and data tables of different aspects of the Hansard records in the database. |
| GitHub Project | All code, text analytics investigations, final documentation and dashboards were committed to a GitHub project to maintain version control. |
| Presentation to AGD Executives | Overview of project and demonstration of dashboards given to AGD Executive team on 4th November 2019. |

Table 1: Project Deliverables

# Project Methodology

The team worked using agile methodologies. While the project team did not conduct a daily “stand-up”, sprint-based planning sessions occurred fortnightly at the start of a development sprint. Outputs were also planned and reported weekly. Weekly meetings held with AGD kept the client up to date on progress and allowed for a forum for feedback.

#### Practical examples of why this approach was useful to the project

As the project was initiated the team looked at various solutions for the infrastructure. After being notified that the client had an Azure account an Azure Architecture was designed. Additional support was also given by Microsoft to establish a deployment plan. After some iterations the client advised that they intended to use an on-premise solution. Sprint planning allowed the team to respond at a fast pace and change the infrastructure to use SQL Server and began looking at alternate extract, transform and load (ETL) processes.

Apache Spark transformations were created and explored for the ETL process as was SSIS packages. Due to the complexity of semi-structured documents and the overall goodness-of-fit within the client's architecture SSIS modules were selected as the final solution for ETL. The agile process allowed the team to plan, build, test and execute important changes within a short time period.

Dashboard versions were packaged and emailed to James Baker (Project Owner) periodically so that they could be published at AGD for testing by the Project Reference Group. The Project Reference Group at AGD consisted of members from three different audit teams who provided user feedback and requirements (Local Government Audit Team, IT Audit Team and Performance Audit Team). Regular publishing of the dashboards allowed regular feedback and iterative development with new versions of the dashboard being published weekly from 2nd October 2019. One exception to this weekly publishing of dashboards was the week starting 14th October due to an upgrade of AGD’s Tableau Server being required

# Project Plan and Alterations

The entire project was developed under the road map of the initial project plan. However, there were several changes that were necessary to impose on the project plan as per client’s requirements.

## Comparison of Original and Updated Project Plan

During the development of the project, there were several changes made to the deliverables and timeline of the project. Initially, meetings with the client were arranged weekly in person or via conference call. But scheduling of meetings changed to weekly with the Project Planning Group and every fortnight with the Project Reference Group from AGD.

In the initial development phase, there were six sprints with each sprint divided for a specific task such as scraping, pre-processing, building database, text analysis, dashboard and handing over deliverables. Later, these sprints were divided into five equal length sprints with the aim of presenting a proof-of-concept visualisation to the client based on sample data at the end of each sprint. This resulted in tasks such as scraping, text analytics and ETL being completed across multiple sprints with the iterative development increasing functionality over time. The purpose of this was to deliver example dashboards as early as possible to get feedback from the Project Reference Group. Figure 1 shows the sprint dates of the development phase.

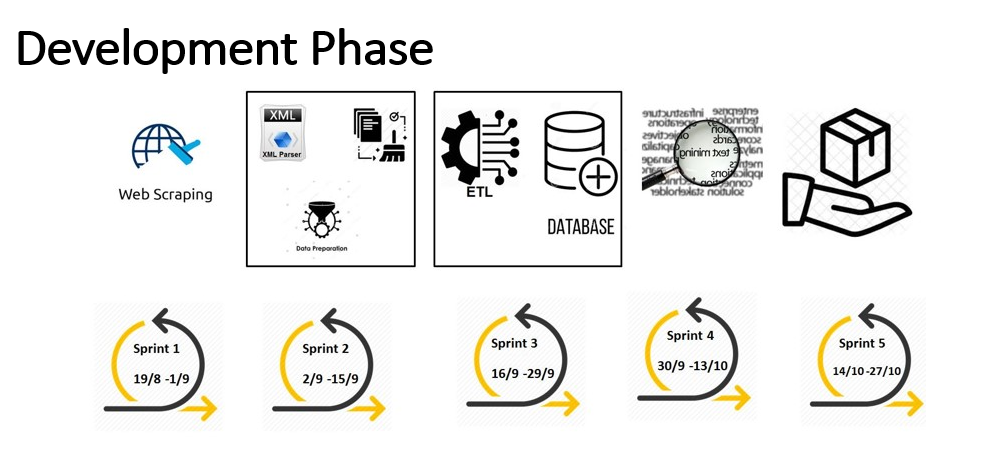


Figure : Development phase of Hansard data mining project

This led to the development of several Gantt chart versions based on client feedback. The final version of the Gantt chart is shown in Appendix N: Gantt Chart. The project was running ahead of schedule in early development sprints which allowed the project team to add and extend a few tasks.

## Task Changes

Communication with different stakeholders in the Project Reference Group resulted in a variety of requirements. Initially, the scope of the project was to scrap and analyse two proceeding types, “Bills” and “Answers to Questions” but later the “Question Time” proceeding type was also added to the scope. Another change in task was to scrape two years of data from Hansard website instead of only one year. Other approved scope changes were discussed previously in Background.

# Deliverables

This section describes the main deliverables of the project and the techniques that were used in developing these products.

## Project Architecture

After several meetings with the AGD project team, the high-level architecture of the project was decided as follows (Figure 2):

1. Web scraping module completed using Selenium, an open-source web-based automation tool, and the programming language Python
2. SQL Server Integration Service (SSIS) used for the extract, transform and load (ETL) tool. The main reason for this decision was that the client already had SSIS installed in their organisation and had experience working with SSIS
3. The client also had Microsoft SQL Server at AGD and therefore was selected to store the Hansard data
4. Text analytics performed using R and Python
5. Hansard dashboards and visualisations developed in Tableau as per client's requirements

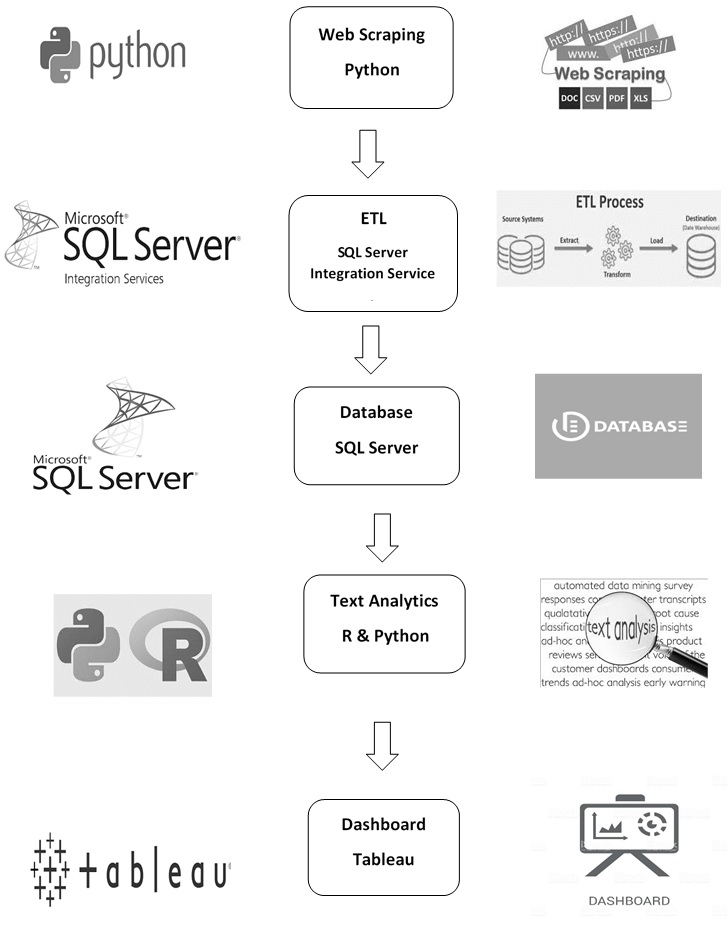


Figure : High-Level Hansard Data Mining Project’s Architecture

**Web Scraping**

A major objective of this project was to retrieve data from the Hansard website. Hansard keeps the record of individual debates of parliament in individual webpages and XML files. The number of files stored in the Hansard website is more than sixty thousand. To access individual files of interest, and to perform the text analytics and visualisation, it was required to build an automatic web scraping system. Web scraping is a method for automatically extracting data from a webpage using software that simulates human web browsing behaviour (Chaulagain et al., 2017).

The web scraping process for this project was done in two stages. In the first stage, the Hansard records for proceeding types “Answers to Questions”, “Bills” and “Question Time” were extracted for two years (7/08/2012 to 1/08/2019) from the Hansard website in XML format. Within these two years Hansard had stored 8,832 records for the three relevant proceeding types. In the second stage, the web scraper was set to automatically download records on a weekly basis on the client’s server environment. As of 30th October 2019, the database contains 9,962 records from 2/08/2017 to 17/10/2019.

Figure 3 depicts the flow of the web scraper created to extract files from the Hansard webpage. Initially the client provides details of the URL and other information such as dates and proceeding types based on user requirements to scrape-hub. Scrape-hub is then activated which controls and optimises the program. Scrape-hub initiates the scraping engine, in our case it is a selenium web driver which opens a webpage, automates it to the required status and parses it using a request from the “beautiful soup” library. “Beautiful soup” is a Python library for pulling data out of HTML or XML files. All code was written in Python and the process automated on AGD systems. The scraper efficiently handles when there is no XML file to be downloaded and checks for any duplicate files. The automated scraper extracts the XML and stores in the database every week without user input.

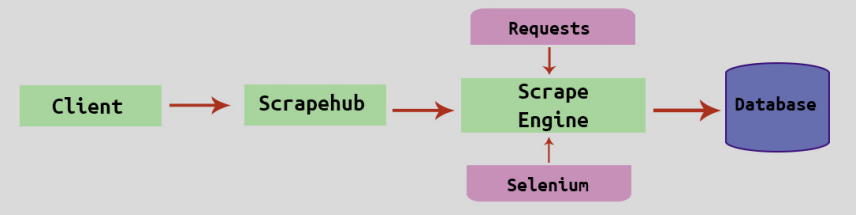


Figure 3: Web Scraping (Chaulagain et al., 2017)

## Extract, Transform and Load Process

A SSIS package was developed to automate the whole ETL process, from scraping the data to storing the record data and text analysis results into a SQL Server database. When designing the ETL processes, SSIS variables were used instead of hard coding to facilitate the handover process to the client. The package consists of four modules:

1. Scraping Module
2. Extract, transform and load the XML files into staging tables
3. Load data into target tables
4. Store the text analytics data into the database

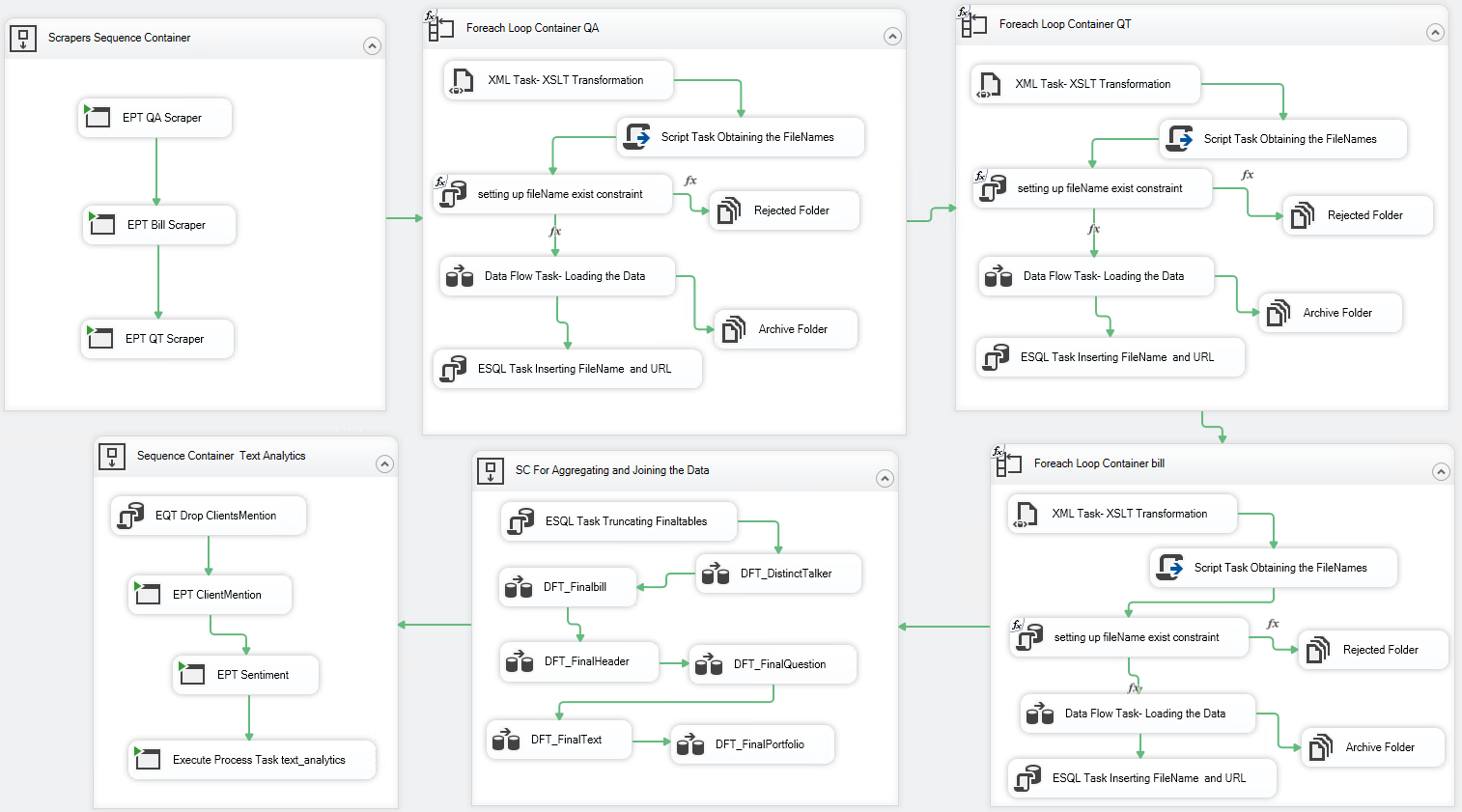


Figure : Hansard SSIS package overview

#### 1. Scraping module

This module uses three Executive Process Task containers within a Sequence Container to run different Python scrapers for the three proceeding types. The destination folder to download the XML files is the input folder for the next modules.

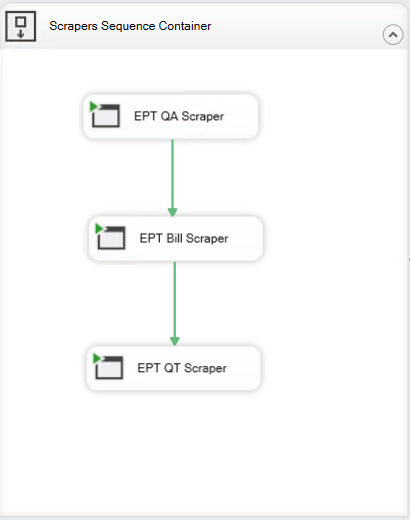


Figure : Scraping Module

#### 2. Extract, transform and load XML files into staging tables

In this module, a For-each Loop container was used to iterate through the source folder containing the XMLs scraped from the Hansard website. First, XML Task containers were used to transform the XML structures using XSLT (eXtensible Stylesheet Language Transformations) operations to a more readable structure. The original XML record had numerous XML elements which were not relevant to this project (e.g. page numbers) and therefore did not need to be extracted and stored in the database. The XSLT was designed so that it could be applied to all 15 proceeding types currently available in Hansard, not just the three proceeding types used in this project. This enables the XSLT to be applied, without modification, to additional proceeding types that may be added to the project in the future.

Then the transformed XMLs were loaded into relevant staging tables (Header, Question, Bill, Portfolio, Talker and Text) using a Data Flow Task container. The unique XML file names have been added into each of the staging tables using the Derived Column container to be able to join the tables together.

This module also avoids loading duplicate files with a mechanism to properly archive and reject duplicate files. First, the mechanism gets the file names for each XML and with the help of user parameters and T-SQL scripts in an Execute SQL Task, the mechanism checks whether the file name already exists in the database or not. If it exists, the file will be moved to the rejected archive folder and renamed otherwise it loads data into the staging tables. After each file is loaded, they are moved to the Archive folder and renamed with the date they were loaded into the database (e.g. [ProcessedOn\_2019\_9\_25]HANSARD-11-34945.xml). The same process is followed for the rejected duplicate files. They will be moved to the Reject folder and renamed (e.g. RejectedOn\_2019\_10\_19]HANSARD-11-34945.xml).

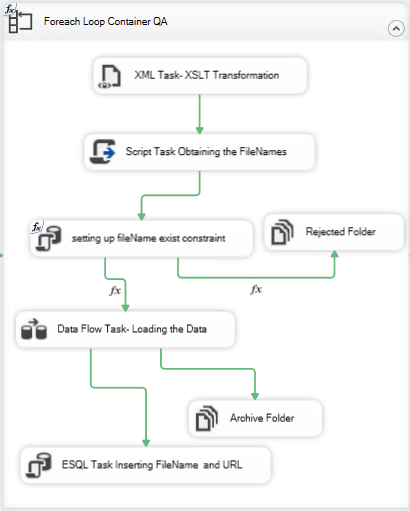


Figure : Load data into staging table module

#### 3. Load data into target tables

In this module, it has endeavoured to aggregate the data by eliminating redundant data as well as mainly changing the FileName variable with HansardID to match the database schema (Appendix B: Database Schema). To do this, we first add an Execute SQL Task container to truncate all the final tables (DistinctTalker, FinalBill, FinalHeader, FinalQuestion, FinalText and FinalPortfolio). This is to make sure no duplicate record are loaded into these tables.

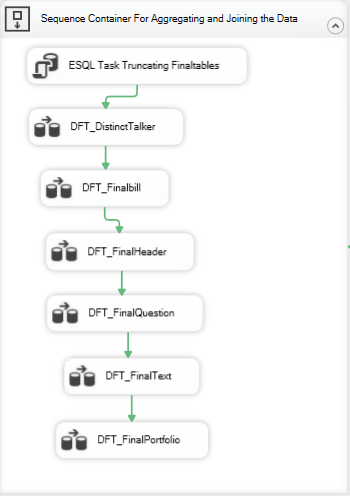


Figure : Hansard SSIS package - Loading data into target table module

DistinctTalker: In TalkerStaging table, there are several duplicate talkers exist which they needed to be removed. For this purpose, we have used the Sort Transformation container, the records with duplicate TalkerID have been removed, and then created a table called DistinctTalker.

For the rest of the target tables, we have joined the corresponding staging tables with HANSARDFilesInfo to replace the HansardID with the FileName. This has been accomplished with the help of a SQL Joint statement in an OLE DB Source container.

#### 4. Store text analytic data into the database

This module automatically runs R and Python text analytics code and stores the results in the database. This data can then be used by the Tableau dashboards. The text analytics executed by this project are:

* Client Mentions
* Document Summary
* Word Count
* Key Words
* Audit Team Key Terms
* Sentiment

The client mentions text analytics captures which records mention AGD clients within the record text. The analytical code has been scripted in R, and using an Execute Process Task container, it has been added to the SSIS package to insert the client mentions into the ClientMentions table. The code looks up a list of clients, and their name variations, in each text and returns the formal name of the clients. The list of clients has been stored in an Excel spreadsheet which AGD can update as required.

Sentiment analysis has also been written in R and calculates the sentiment score of the text at the record level (Hansard ID). The sentiment code is integrated into SSIS using an Execute Process Task container which inserts the sentiment scores into the HANSARDFilesInfo table for each record.

Document summary and key words were written in Python and is executed by SSIS to add columns to the HANSARDFilesInfo for each record. Word count is executed by SSIS to add as a column to the FinalText table for each record text fragment. Whether record text mentions audit team key terms has been added to the SSIS package to insert the term mentions into the KeyTerms table

#### Deployment and automation of the package

After designing and testing the package, it was deployed into SQL Server Management Studio. As per the client’s request, the package has been scheduled to run automatically (weekly) on Saturday at 12 AM using the SQL Server Agent. It updates the database with records from the last seven days.

## Dashboards

Five proof-of-concept dashboards were developed using the interactive visualisation software Tableau (Tableau, 2019). Tableau can query different data sources such as relational databases and spreadsheets and can generate a variety of visualisations. This visualisation software was used because it is currently in use at AGD which will make the transition to AGD systems easier. The purpose of these dashboards is to allow auditors to analyse Hansard records as part of their audit process. The dashboard functionality requested by the Project Reference Group included:

* Free text search of Hansard records, including being able to search for multiple terms simultaneously in a record
* Automated searching of key terms identified by three audit teams
* Provide link to Hansard record on website
* Be able to identify relevant records without having to read the entire record text
* Search for records within specified date ranges
* Sentiment of discussion in a record

Additional requirements and improvements were identified over multiple meetings with the Project Reference Group as the dashboards were developed. The dashboard requirements and their current progress status were documented in a spreadsheet which was available to all members of the project team and Project Reference Group.

The five dashboards developed using Tableau Desktop 2019.2 were:

* Summary Dashboard
* Record Search Dashboard
* Audit Team Dashboard
* Subject Overview Dashboard
* Client Overview Dashboard

#### Summary Dashboard

The Summary Dashboard (Figure 8) gives an overview of all Hansard records available in the database. Metrics are displayed on the left side of the dashboard and measure: Number of Records; Subjects Discussed; Most Popular Subject (and its number of mentions); and Most Discussed Bill (and its number of mentions). A Bill name can appear in both the most discussed subject and most discussed Bill metrics as a Bill can be recorded in metadata as the subject of a record as well as the Bill discussed. However, because these metrics search different metadata, they can display different values. For example, for Bills the “Education and Children’s Services Bill” is mentioned in 36 records as the subject, but only 35 records as the Bill being discussed.

Several charts can be displayed in the centre of the dashboard depending on the selection made using the “Select View” drop down menu. The bar charts and heat maps that can be displayed include:

* Top 50 Subjects (by number of Hansard Records and Days Discussed)
* Top 50 Bills (by number of Hansard Records)
* Top 50 Portfolios (by number of Hansard Records)
* Top Clients (by number of Hansard Records)
* Subjects Discussed over Time

See Appendix F: Dashboard Charts for screenshots of charts that are displayed in the Summary Dashboard. Hansard records can be filtered by Date and Proceeding Type (e.g. Answers to Questions; Bills; Question Time) to narrow records of interest.

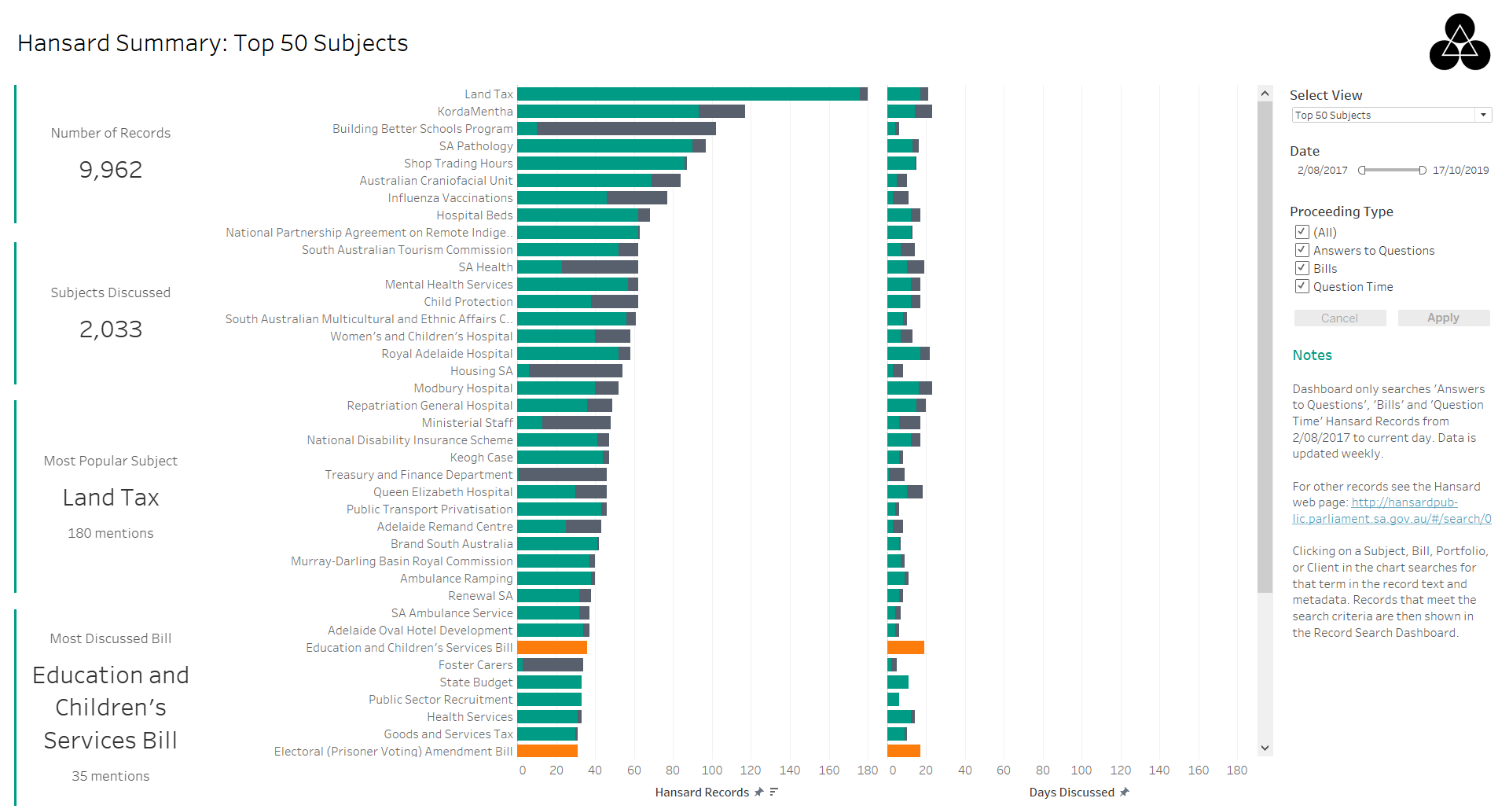


Figure 8: Summary Dashboard

Clicking on a Subject, Bill, Portfolio or Client in one of the charts or metrics searches for that term in the Hansard record text and metadata. Records that meet the search criteria are then shown in the Record Search Dashboard (Figure 9).

#### Record Search Dashboard

The Record Search Dashboard displays the subject, proceeding type, date and text summary of each record that matches the search term. The search term is searched for in record text and metadata such as subject, Bill, Portfolio Name (e.g. Minister of Health) and Client. Regular expressions can also be used in the search term for more precise searching of record text such as searching for multiple terms anywhere in the document. The records in this dashboard can be filtered on Client as well as Date and Proceeding Type.

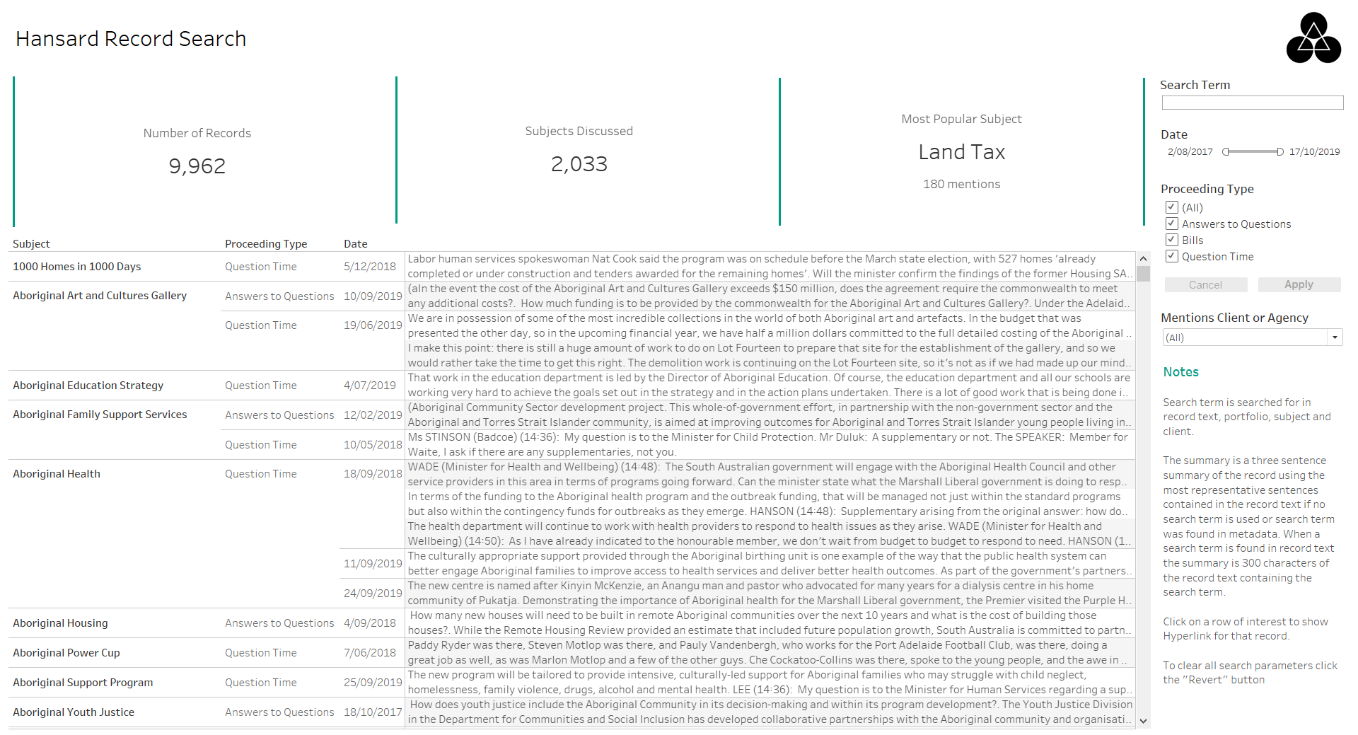


Figure 9: Record Search Dashboard

Hovering over the text summary of a record provides additional information, such as key words and sentiment, to help the user identify whether the record is relevant for their audit (Figure 10 and Figure 11). The key words displayed are the 10 most ranked words of the record text (with words lemmatised and accentuation removed) using the Python library gensim (Řehůřek, 2019). The word count is also displayed to give an indication of the size of the record. This could be useful to auditors as short records may not be relevant to their audits even if it matches the search term. Once a relevant record is identified the user can click on the HTML link to get access to the full record on the Hansard webpage. This meets the requirements of providing a HTML link to the Hansard webpage and being able to identify relevant records without having to read the entire record text.

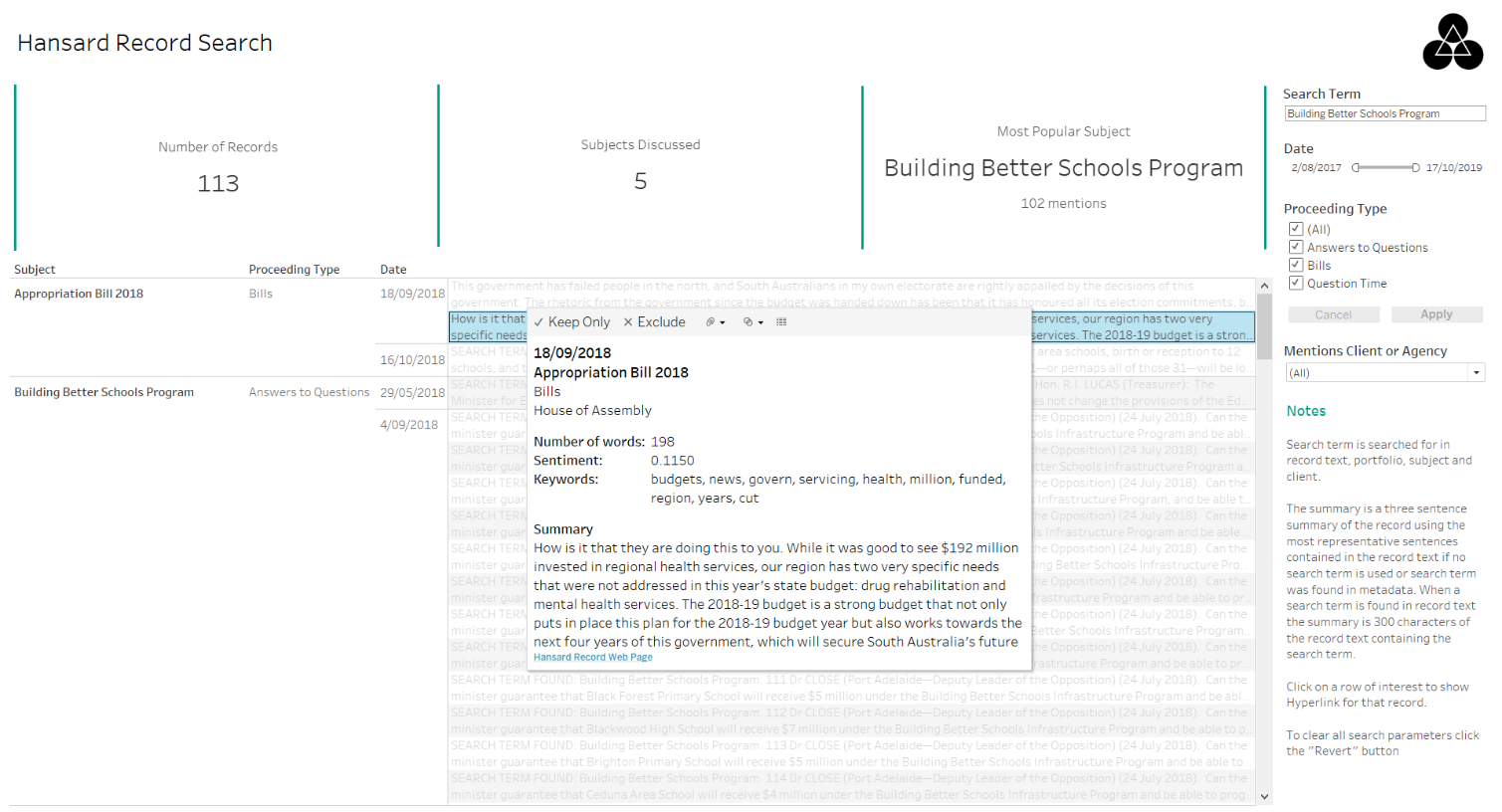


Figure 10: Record Search Dashboard (with search term and tooltip)

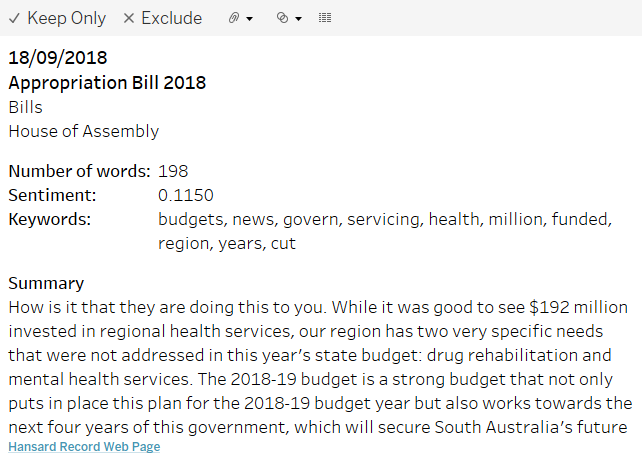


Figure : Record Search Dashboard Tooltip

#### Audit Team Dashboard

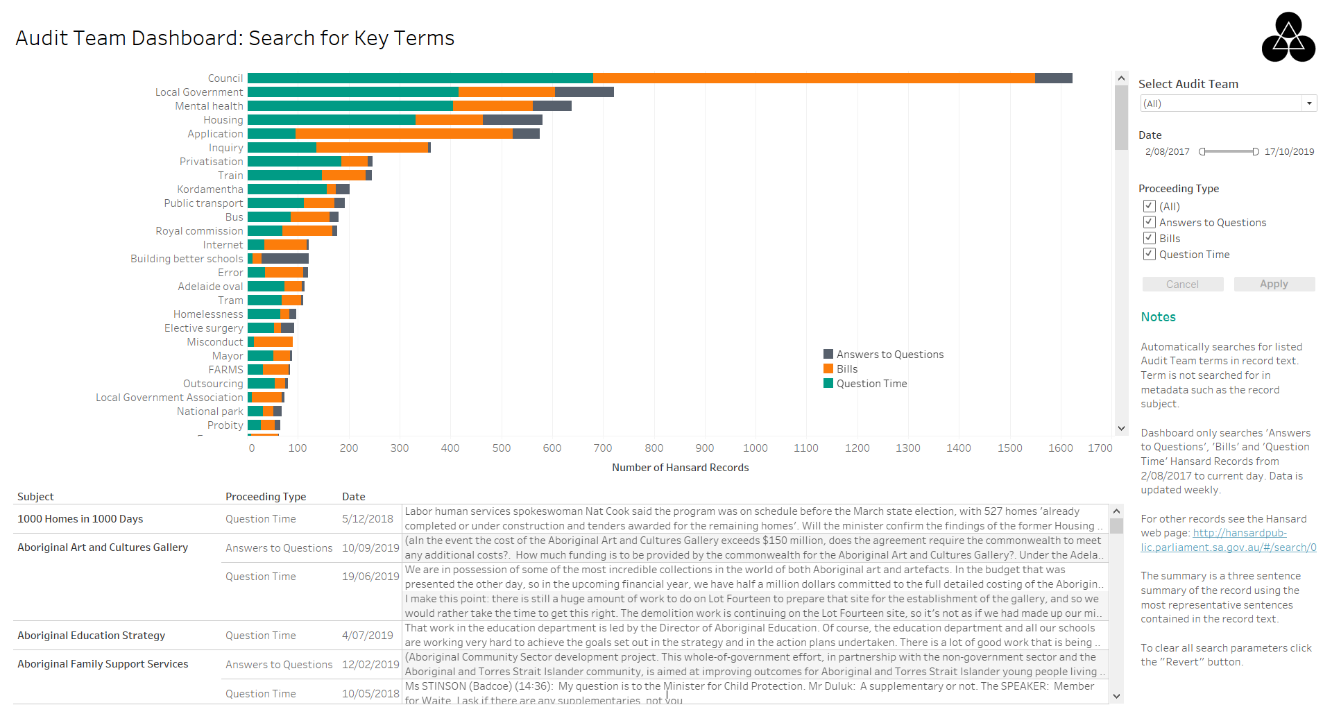
The Audit Team Dashboard (Figure 12) displays the number of records key terms have been mentioned for a given audit team. Audit teams will commonly search for specific terms which were provided in a spreadsheet. This spreadsheet also contained alternate spelling and plural forms for some of the terms. To save time these commonly searched for key terms were processed and stored in the database. Clicking on a term in the bar chart filters the table below to show only records that contained that term (as a whole word) in the record text. These terms are not searched for in metadata such as subject.

Figure 12: Audit Team Dashboard

#### Subject Summary Dashboard

The Subject Summary Dashboard (Figure 13) gives an overview of the subjects discussed in Hansard records. The treemap in the centre of the dashboard displays the number of records per Proceeding Type for a selected subject or all subjects. Clicking on a Proceeding Type drills down to show the Portfolio Names that participated in that subject.



Figure 13: Subject Summary Dashboard drilled down to Portfolio Names

#### Client Summary Dashboard

The Client Summary Dashboard (Figure 14) gives an overview of AGD Clients. Initially the number of records per Client Type is shown in the treemap and clicking on a type will display the individual clients. Clicking on a specific client will filter the table shown below the treemap to display records that mention that selected client.

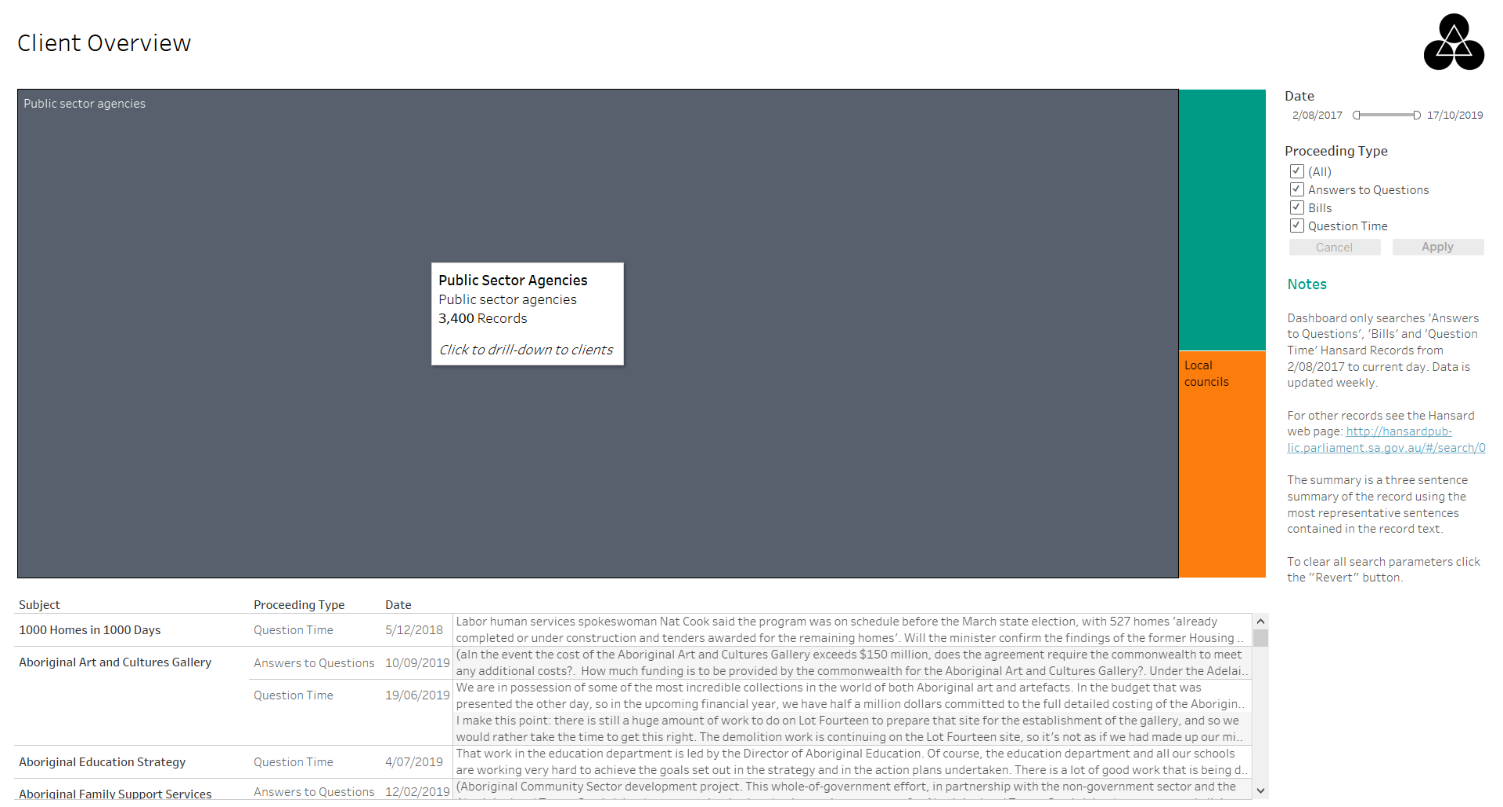


Figure 14: Client Summary Dashboard

#### Dashboard Design

All dashboards contain notes, located in the bottom right corner of the dashboard, to provide information on how to use the dashboard and to describe important features. To improve understanding of the data the number of colours used in the charts were minimised and mainly bar charts used. Unnecessary labelling and other ornamentations were also removed.

A colour-blind palette was used for the charts to improve accessibility. However, one of the original colours was replaced with a green colour (#009B85) that is used on the AGD website. While this improved the appearance of the dashboards, and coordinated with AGD colour themes, it may not be as accessible for colour-blind people compared to the original colour-blind palette. The dashboards were evaluated with small monitor sizes (1280 x 1024), as well as standard sizes, because some users at AGD still have small monitors. The layout of the dashboards assumed the English left-to-right reading pattern and therefore placed the most important information towards the top left. This included information such as metrics and the main charts. Less important or supporting information was placed on the right such as the AGD Logo (top right corner) and the explaining notes (bottom right corner).

## Text Analytics

In this section the investigation and implementation of several text analytic approaches are described. Approaches investigated included: word trees, document summarisation; sentiment analysis, and topic modelling. These text analytic approaches were applied to 9,962 Hansard records which had a median 45 words per sentence. The text of a record included short text such as headings and long text such as table contents which resulted in a wide range of sentence lengths (minimum 1 word and maximum 5,479 words in a sentence).

#### Word Tree

The following visualisation was delivered to the client as an alternative that would not be developed within Tableau dashboards. The benefits to the user are a seamless exploration experience of a word tree. The word tree is a collection of Hansards text that contain a user selected root word. The flexible click through experience allows the user to navigate relevant Hansard records that contain the same keyword in a simultaneous fashion.

This visualisation enables the user to search (1) for any word which returns a word tree showing the structure of language around the word contained in all documents that contain the root word (1). The interactive visualisation allows the user to click on other words within the visualisation and see the language used around the clicked word (2).

The visualisation modifies the display of text using two methods to enhance the visibility of frequency: size and boldness. The higher the frequency of usage for a term the larger/bolder it appears.

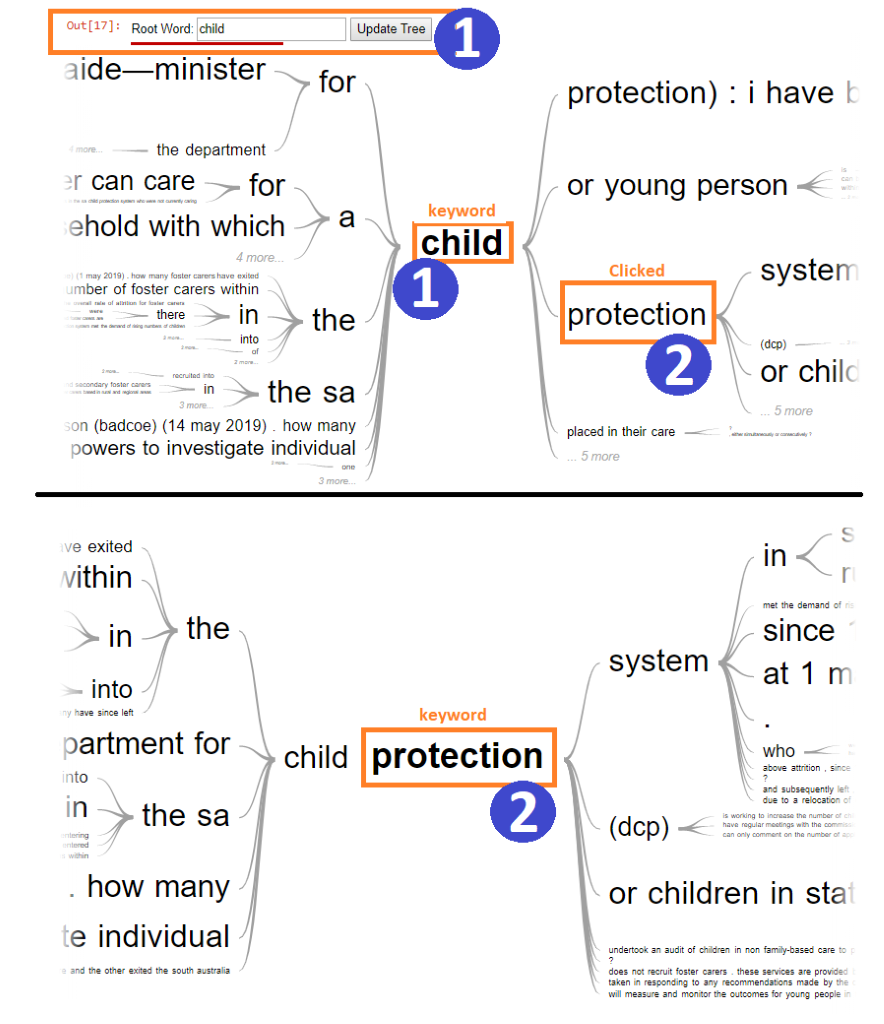


Figure 15: Word Tree

Word tokenization was an important transformation process within the project in order to establish efficiency within the computation of algorithms. A bag-of-words was generated in order to establish a structured set of words that would correspond to an integer datatype. Additionally, on generation of the bag-of-words a lookup reference table was created that would map individual words to Hansard text documents.

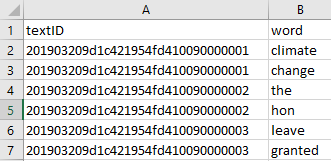


Figure 16: Words mapped to Hansard text

A Python process was written (updateLookUp.ipynb) that takes a list of Hansard Text IDs from the database and compares them to the wordHansardLookup.csv which contains the processed Text IDs. The process then compares the database list of Text IDs against the wordHansardLookup.csv Text IDs to return only the Text IDs that have not been processed and then proceeds to tokenize the delta of Text IDs.

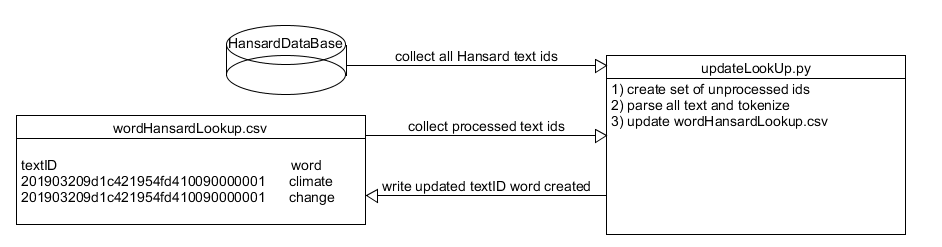


Figure 17: Word Tree Process

#### Hansard Synsets and Hypernyms, Lemmas and Synonyms

“A Synset is one or more sets of synonyms where as a hypernym is a word or phrase that is included within another words semantic field”, where as a Lemma is the word that stands as the definition, as to a lexeme is a group of words that have the same meaning (Cruz et al., 2014) .

#### Hansard Antonyms Application

Antonyms are an important concept when using algorithms such as sentiment analysis. In the case of measuring if the sentence is positive or negatively positioned Antonyms play the role of identifying and replacing inverse logical phrasing. Such as “Not bad” would be replaced with “good”. This is an important component for data transformations that would be considered “pre-processing” that is required in order to adequately measure text sentiment (Cruz et al., 2014).

#### Text Summarisation

A requirement described by the AGD Project Reference Group was to identify relevant Hansard records without having to read the entire record. To meet this requirement several text summarisation techniques were investigated and implemented to summarise a Hansard record. Text summarisation is the process of creating a short and coherent version of a longer document. There are two main approaches for text summarisation: extractive methods which use phrases and sentences from the source document to make up a new summary; and abstractive methods which generate new phrases and sentences that capture the meaning of the document (Brownlee, 2017). Abstractive methods may produce summaries similar to how humans summarise, but this approach is much more challenging to implement. Therefore, due to the short duration of this project abstractive methods were not implemented.

Three approaches were implemented and presented to the AGD Project Reference Group for evaluation. The three text summary approaches evaluated were:

1. First 200 characters of the record text
2. First 300 characters of the record text that start from a search term specified by the user. If no search term is specified, or regular expressions are used in the search term, Option 3 (three-sentence summary) is displayed.
3. Three sentences from the record text that was analysed as being most representative of the record

The first two approaches were implemented within Tableau using in-built string operations. The third approach was implemented using the PageRank algorithm in the Python programming language (Dubey, 2018). The PageRank algorithm is an algorithm originally designed by Google to rank web pages based on the structure of the incoming links (NetworkX, 2014). It assumes that important pages are linked by other important pages. Instead of web pages, this algorithm was used to rank the most representative sentences in a Hansard record. An advantage of this approach is that it does not rely on any previous training data and can work with any piece of text. The numpy (NetworkX, 2014) implementation of the PageRank algorithm was used because other implementations used methods which were not guaranteed to converge to a solution and this implementation was faster.

Several pre-processing steps were undertaken before the text was analysed to determine the most representative sentences. The text for a Hansard record was pre-processed using the following steps:

* Sentences evaluated must be between 5 and 100 words (median sentence length for all records was 45 words)
* Removed tags (e.g. 1.&#x9;)
* Converted characters to lowercase
* Removed punctuation
* Lemmatised words using Word Net Lemmatiser
* Stemmed words using Snowball Stemmer (English)
* Removed common English words (e.g. the, is, at)

Limiting the sentences analysed resulted in only 54.38% of the Hansard record text being analysed to form a summary. This in combination with the other pre-processing steps greatly reduced the running time for the algorithm. Producing summaries for 9,962 records took approximately two hours compared to more than five hours when all text was considered.

Some Hansard records can be very short. For example, the record HANSARD-10-21289 consisted of the text: “Bills. Children and Young People (Safety) Bill. Assent. His Excellency the Governor assented to the bill” (Parliament of South Australia, 2019). In these cases, it is not efficient or useful to construct a summary from the most representative sentences. Therefore, if the full text of a record is less than 200 characters long, the entire text is returned instead of a three-sentence summary.

The three document summary approaches were implemented in three Record Search dashboards that were identical except for the summary method presented. These dashboards were published on AGD systems so that the Project Reference Group could evaluate the three different options. The summary approach preferred by the Project Reference Group was the second option which contained the user specified search term. This option was preferred because it gave context to the search term match in a record, allowing the user to better evaluate whether a record was relevant. The dashboards delivered to AGD only contained the dashboard with the preferred search term summary.

#### Topic Modelling

Generally topic modelling is used for automatically arranging, comprehending, scrutinising, and summarising big electronic archives. In other words, topic modelling provides an efficient and effective way to analyse high volumes of text. The main mechanism behind topic modelling is to identify patterns of words present in different documents and connecting documents sharing alike patterns. All documents are mixtures of numerous topics and a topic is a probability distribution over words (Tong and Zhang, 2016). There are different approaches to perform topic modelling in text documents such as Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), and Correlated Topic Model (CTM). But one commonly used technique is Latent Dirichlet Allocation (LDA). This project used LDA to perform topic modelling of the “Question” in Hansard records. Topic modelling would help users to better understand and highlight main topics that lie in the thousands of Hansard discussions. This assists the client with their audit planning process.

Latent Dirichlet Allocation (LDA) is a text mining algorithm based on Bayesian inference. The LDA algorithm connects all document with a probability distribution over topics and topics are probability distributions over words (Alghamdi and Alfalqi, 2015). Before performing LDA the text document was pre-processed. All common English words (such as is, the, that, you etc.) were removed and the text converted to lowercase letters. All names of parliament members and electorates were removed from the documents. Also removed were phrases used in parliament proceedings such as, “Members interjecting:”, “The SPEAKER: Order!” etc. After pre-processing, the text was put in a document term matrix (DTM) for further processing. This project used the “textmineR” library (Jones and Doane, 2019) to perform topic modelling because it is a commonly used library. This DTM is then used to perform topic modelling on the data.

One major problem that was encountered was selecting the appropriate number of topics. To address this problem the probabilistic coherence score was used to select the optimum number of topics. The highest coherence score results in the optimum number of topics that explains the relationships between words in the documents. The coherence score increases when the most probable words frequently co-occur in a topic. Therefore, different number of topics where passed to the LDA algorithm and the coherence score of each topic calculated and plotted (Figure 18).

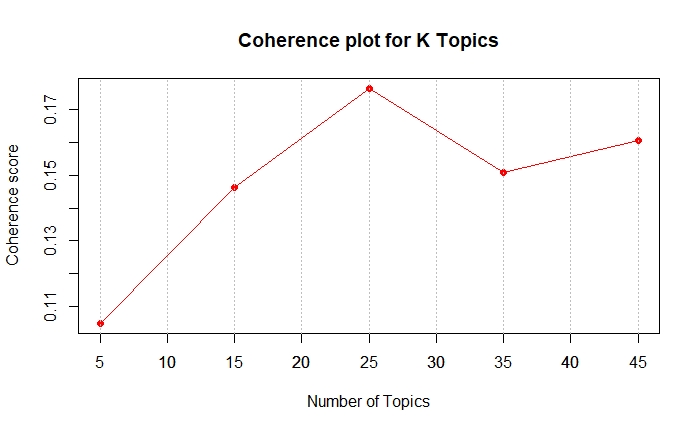


Figure : Coherence score of number of topics

For the two years of Hansard “Question” data, the number of topics was set to 25 as it had the highest coherence score (Figure 18). After determining the total number of topics to generate, the LDA model was fitted. We also calculated the prevalence of individual topics which shows how much a topic explains the document (DiMaggio et al., 2013). A higher prevalence score indicates that that topic explains more about the document. Appendix M: List of Topics Generated by LDA shows the list of 25 topics that were generated with the Top 5 words of each topic and its prevalence score.

The “LDAvis” library in the R programming language was used to visualise the generated topics (Sievert and Shirley, 2016). This library helped to create an interactive web-based visualisation of the topic model. Figure 19 shows a screenshot of the interactive visualisation.

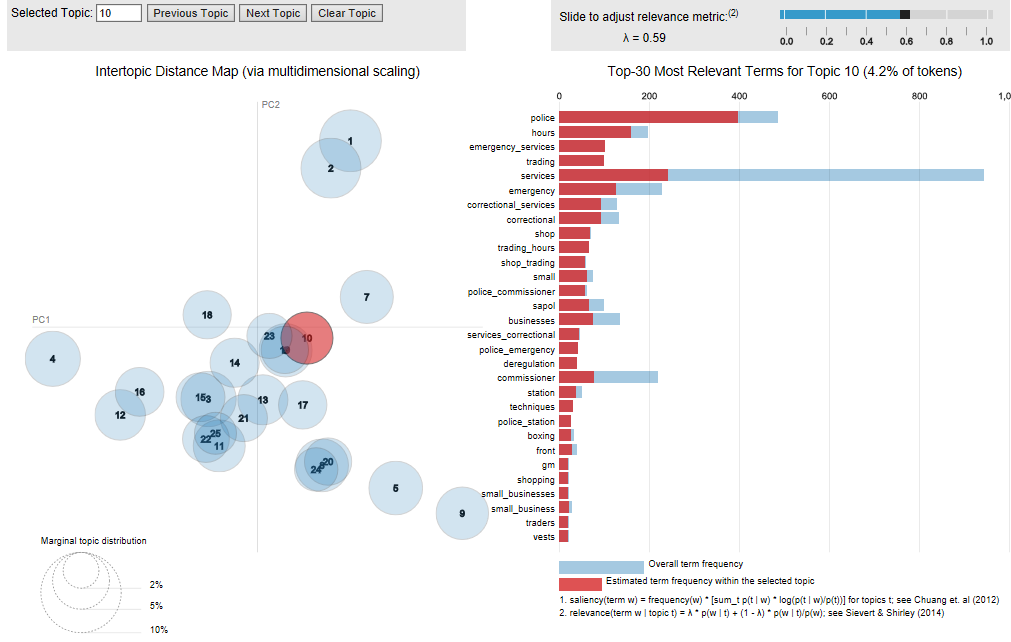


Figure : Topic model visualisation

In the above visualisation, the left side shows 25 different topics plotted in a two-dimensional graph. The distance between the topics shows the similarity between the topics. If the distance between the topics is short, then the topics share similar terms (words) and distant topics shares vague terms. The bar chart on the right side shows the frequency of words in the entire document (represented by “sky blue” colour) and frequency of words in individual topics (represented by red colour). Here, the value of relevance matric shown in top right corner, and represented by lambda (λ), can be adjusted from 0 to 1 which gives the relevance of different words in an individual topic. In general, the value of λ is set to 0.6 to obtain the best results (Sievert and Shirley, 2014).

#### Sentiment Analysis

Sentiment analysis is one approach used to examine the articulation of text or speech and identify if it is positive, negative or neutral, and to what extent (Abdullah and Khan, 2019). In this project sentiment analysis is performed on each Hansard record and was requested by the Project Reference Group.

Sentiment analysis was coded in the R programming language using package “sentimentr” (Rinker, 2016). In general, the most common approach is to use a simple sentiment analysis technique that calculates sentiment or polarity of the document by matching words back to a dictionary of words identified as “positive,” “negative,” or “neutral” (Kaushik et al., 2015). However, “sentimentr” uses a different approach to analyse the sentiment of the sentence by making use of a lexicon of words called valence shifters (Rinker, 2016). These valance shifters act as support to the sentiment of the words in either direction by considering negators (not, can’t), amplifiers (absolutely, certainly), de-amplifiers (almost, barely) and adversative conjugation (although, that being said) (Rinker, 2016). The sentimentr package was used to calculate the sentiment of a Hansard record which was visualised in Tableau as heat maps. Figure 20 and Figure 21 shows the sentiment heat map for Bills and subjects of Hansard records respectively. Dark blue indicates positive sentiment, red is negative sentiment and white indicates neutral sentiment.

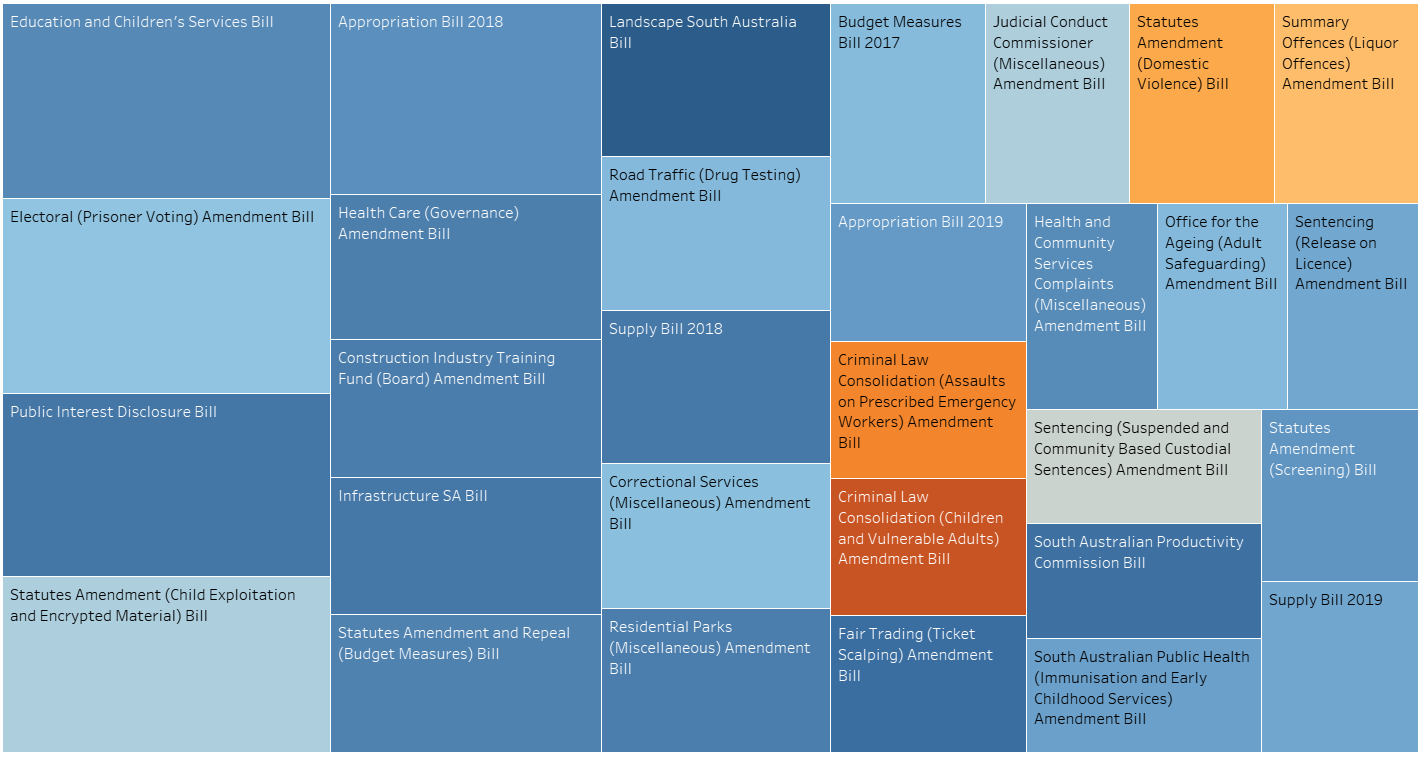


Figure 20: Average sentiment for each Bill heatmap

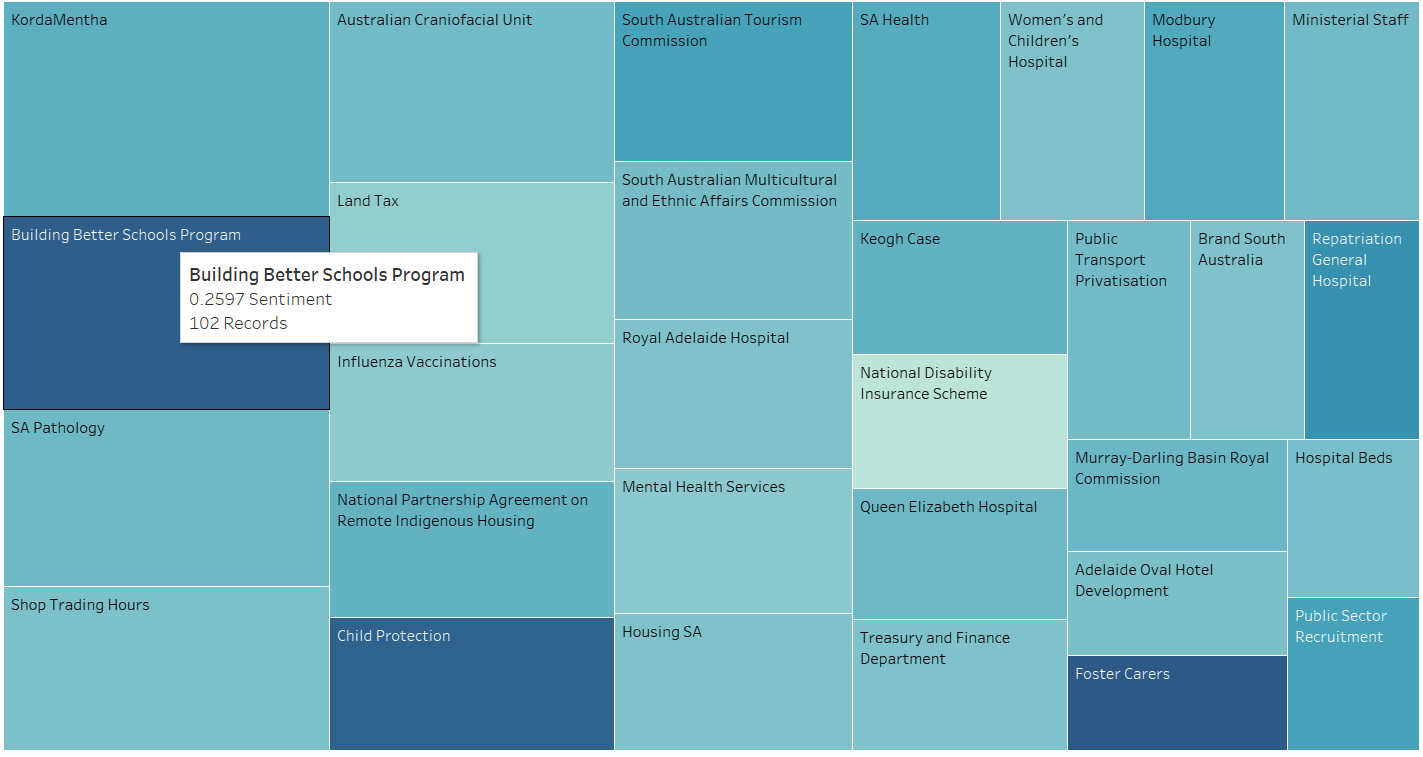


Figure 21: Average sentiment for each subject heatmap (with tooltip)

The distribution of sentiment of Hansard records is shown in Figure 22. This shows that most records have a slightly positive sentiment.

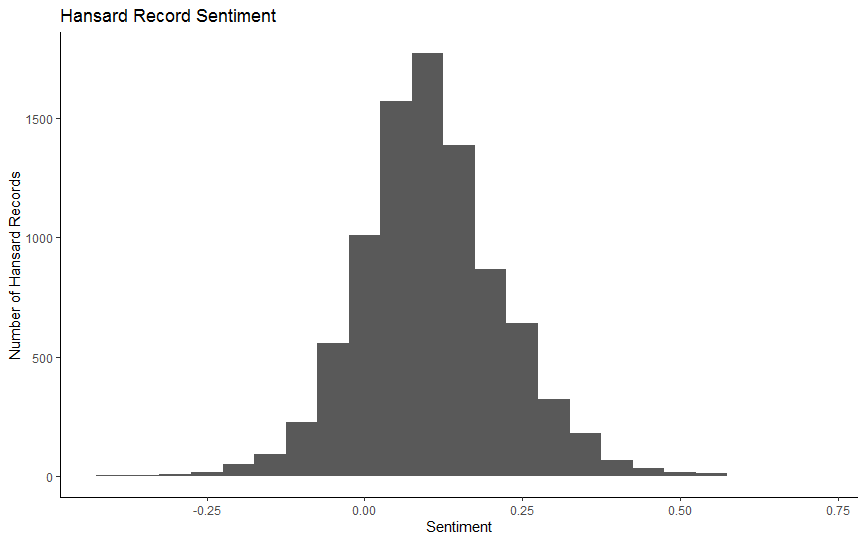


Figure 22: Sentiment of Hansard Records

The minimum score of –0.419 represents the extreme negative sentiment of the records whereas the most positive sentiment score was only 0.675 (Table 2). The overall average sentiment score is 0.105 which indicates Hansard records have a very neutral sentiment using this sentiment analysis technique.

|  |  |  |  |
| --- | --- | --- | --- |
| Min | Median | Mean | Max |
| -0.419 | 0.098 | 0.105 | 0.675 |

Table : Sentiment Score Statistics

# Project Performance Assessment

This section assesses how the project team performed against the critical success factors, schedule and outputs.

## Performance against critical success factors

This project met all critical success factors that were described in the project plan. Performance against the four critical success factors are discussed below.

1. *Dashboards will be interactive and allow the searching of text*

All dashboards created for AGD were interactive as described in the Dashboards section. The Record Search Dashboard allowed the searching of record text to find relevant records. Regular expressions can also be used for more advanced searching.

1. *Ability to be able to identify in documents when the Auditor-General's Department of South Australia and its clients are mentioned*

The dashboards allow AGD to identify which clients are most mentioned in records and can narrow records to those that mention one or more clients.

1. *Ability to be able to identify hot topics in parliament discussions*

The Summary Dashboard allows AGD auditors to identify which subjects are most mentioned in records and can filter records that discuss a specific subject. The popularity of a subject over time is also visualised (Figure 29) allowing auditors to identify whether a subject was discussed over a long period of time or was discussed in many records during a short period of time (potential hot topics).

1. *Software and automated processes developed will be developed and documented in a way that allows them to be deployed by AGD on their own systems*

As discussed in previous sections all code, dashboards and processes were developed so that they could be deployed on AGD systems. The dashboards are currently published for use by the department and the automated scraping, processing and database has been deployed on AGD systems. An Installation and Deployment Guide was also provided with instructions on how to setup the project.

In addition to meeting critical success factors this project resulted in clear organisation benefits. The dashboards provided will allow AGD auditors to more quickly identify relevant records and topics for their audits. AGD plan to publish the dashboards so that everyone at the department can use them during their audit process and provide feedback for future project work.

## Performance against schedule

The project team met all deadlines on time as per the project plan schedule. Most of the project team attended scheduled client meetings with some members unable to attend occasionally due to work commitments. The two-week development sprints were planned the day before the sprint were due to start and the sprint plan communicated to the client using Microsoft Teams. There were some minor variations to the planned tasks of a sprint such as the Installation and Deployment Guide draft being delayed due to other work taking priority. When these delays occurred, the delayed tasks were able to be completed in the following sprint.

## Performance against outputs

All the planned outputs in the project plan were delivered to AGD to the client’s satisfaction. Some additional outputs were also completed such as writing a user guide. The dashboards and text analytics presented to the Project Reference Group received positive feedback with members saying that the products will be useful for their audits. The clients were also satisfied that the project could run on their own systems with only minor modifications required if it were moved to another server.

# Lessons Learned

Lessons learned during the Hansard Data Mining Project were captured by the project team with the most important ones summarised below:

* The importance of negotiating and finding common ground with clients during all phases of the project
* People communicate differently which can cause conflict if not managed properly. Using a variety of communication methods helped to address this problem.
* Extracting the correct project requirements is very important as not doing this may result in wrong deliverables. Regular communication with clients and confirming decisions made were important in making sure everyone had the same understanding of the requirements and deliverables.
* Communication is key to the success of any project. We learned that iterative feedback and development can lead to greater team efficiency.
* The importance of practising Agile project management principals in IT projects. Asking early feedback on the project deliverables are important to ensure that products meet the client’s requirements. For instance, showing dashboards early before real data was available was crucial in developing dashboards that met their needs.
* Agreements with the client on infrastructure is critical. We were thinking of using a wide range of tools and technologies. However, after meeting with the client we found out that AGD does not want to use cloud solutions (e.g. Microsoft Azure), and they preferred to use tools they have already been using. This helped us to avoid wasted effort as well as to decrease the project budget.
* Importance of team collaboration tools (e.g. Microsoft Teams, Microsoft Planner, WhatsApp etc.) which allowed us to raise problems and questions early rather than waiting until the Monday project team meeting

# Recommendations and Future Work

The project team identified several recommendations and future work that could be completed for this project. It is recommended that improvements be made to the project to improve process setup and robustness. This includes:

* Automatically install required R and Python packages
* Extend configuration file for use in SSIS (currently only used by text analytics and web scraper)
* Automate Tableau data extract and publishing of dashboards to AGD
* Specify which columns in HANSARD database tables cannot be null and which are primary keys. This has not been done for most tables.
* Automatically determine when the spreadsheet listing audit team key terms have changed and trigger reconstruction of the table. Currently, this requires manual execution of a batch file.
* Optimisation of scraping, processing, and SSIS to decrease time required to upload new data into the database

Improvements could also be made to the text analytics applied in the dashboards. The text analytics implemented should be evaluated against other implementations to ensure that it is the most appropriate for AGD. For example, the sentiment analysis should be evaluated as the algorithm used scores the word ‘government’ as negative sentiment. This may not be appropriate when applied to government records. Additional techniques could also be investigated and applied such as document similarity which could be used to find similar records to ones that were already identified as relevant by an auditor. Named entity recognition to detect named entities in the text such as organisations, people, locations and more could also be useful. Additional work on topic modelling and word tree could be completed to integrate this work into the dashboards.

Additionally, the improvements listed below have been requested by the Project Reference Group and were not able to be completed in this project:

* Email alerts about new records which mention clients or key terms
* Inclusion of Committee information records in database and dashboards
* Apply this dashboard approach to Gazettes data

# Conclusion

The Hansard Data Mining Project has allowed the project team to exercise practical skills on an industry problem. Overall the project has been successful with timely delivery of a complex ETL project that has enabled text analytics and dashboarding for AGD.

Within this project, organisational benefits have been derived for AGD. A complex process that originally involved the manual collection and reading of multiple web-based files by AGD auditors has been partially automated for ease of analysis and consumption. AGD has expressed that the Tableau dashboards produced will be published on AGD systems for the entire department to use.

References

Abdullah, A and Khan, MZ 2019, A Study on Sentiment Analysis Techniques of Twitter Data, International Journal of Advanced Computer Science and Applications (IJACSA), 10(2), pp. 361-374.

Alghamdi, R, & Alfalqi, K. 2015, A Survey of Topic Modeling in Text Mining, International Journal of Advanced Computer Science and Applications (IJACSA), 6(1), pp. 147-153.

Auditor-General's Department 2019, Auditor-General's Department, Viewed 26th October 2019, <https://www.audit.sa.gov.au/about-us/the-auditor-general>.

Brownlee, J 2017, A Gentle Introduction to Text Summarization, Viewed 25th October 2019 <https://machinelearningmastery.com/gentle-introduction-text-summarization/>.

Chaulagain, R, Pandey, S, Basnet, SR & Shakya, S 2017, Cloud Based Web Scraping for Big Data Applications, 2017 IEEE International Conference on Smart Cloud.

Cruz, FL, Troyano, JA, Pontes, B & Ortega, FJ 2014, Building layered, multilingual sentiment lexicons at synset and lemma levels, Expert Systems with Applications, Oct 1st 2014, Vol. 41(13), p.5984(11).

DiMaggio, P, Nag, M & Blei, D 2013, Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of U.S. government arts funding, Poetics, Volume 41, p. 570–606.

Dubey, P 2018, Understand Text Summarization and create your own summarizer in python: An Introduction to Text Summarization, Viewed 28th October <https://towardsdatascience.com/understand-text-summarization-and-create-your-own-summarizer-in-python-b26a9f09fc70>.

Hansard 2008, Parliament of South Australia, Viewed 27th October 2019, <http://www.parliament.sa.gov.au/Hansard/Pages/GeneralHansardInformation.aspx>.

Jones, J, & Doane, W 2019, The Comprehensive R Archive Network, Viewed 1st June 2019,   
<https://cran.r-project.org/web/packages/textmineR/textmineR.pdf>.

Kaushik, A, Kaushik2, A, & Naithani, S 2015, A Study on Sentiment Analysis: Methods and Tools, International Journal of Science and Research (IJSR), 4(12), pp. 287-292.

NetworkX 2014, pagerank\_numpy, Viewed 25th October 2019 <https://networkx.github.io/documentation/networkx-1.9/reference/generated/networkx.algorithms.link\_analysis.pagerank\_alg.pagerank\_numpy.html>.

Parliament of South Australia 2017, Statutes Amendment (Heavy Vehicles Registration Fees) Bill, Viewed 17th October 2019, <http://hansardpublic.parliament.sa.gov.au/Pages/HansardResult.aspx#/docid/HANSARD-10-21310>.

Parliament of South Australia 2019, Children and Young People (Safety) Bill, Viewed 25th October 2019, <http://hansardpublic.parliament.sa.gov.au/Pages/HansardResult.aspx#/docid/HANSARD-10-21289>.

Řehůřek, R (2019), Keywords for TextRank summarization algorithm, Viewed 30th October 2019, <https://radimrehurek.com/gensim/summarization/keywords.html>.

Rinker, T, 2016. Sentimentr, Viewed 12th August 2019,   
<https://cran.r-project.org/web/packages/sentimentr/readme/README.html>.

Sievert, C & Shirley, K 2014, Proceedings of the Workshop on Interactive Language Learning, Visualization and Interfaces, Baltimore, Maryland, USA, Association for Computational Linguistics.

Sievert, C & Shirley, K 2016, The Comprehensive R Archive Network, Viewed 10th June 2019 <https://cran.r-project.org/web/packages/LDAvis/LDAvis.pdf>.

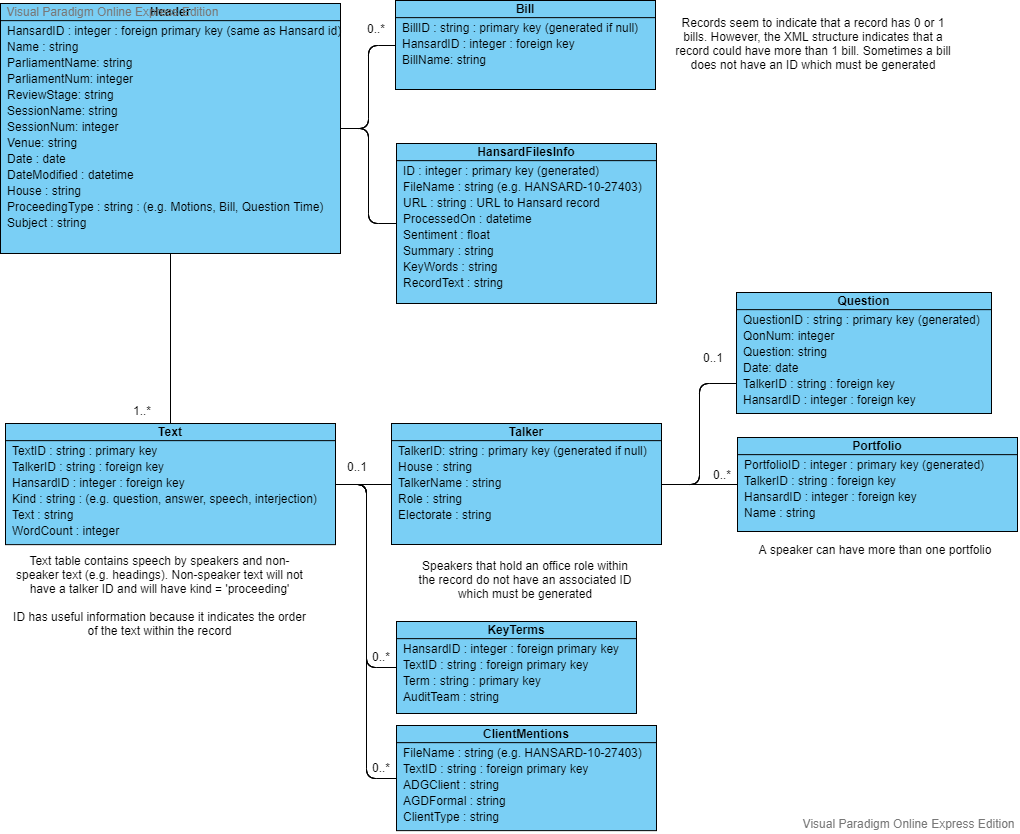
Tableau 2019, Tableau Desktop, Viewed 25th October 2019, <https://www.tableau.com/en-au/products/desktop>.

Tong, Z, & Zhang, H 2016, A Text Mining Research Based on LDA Topic Modelling, Sixth International Conference on Computer Science, Engineering and Information Technology.

# Appendix A: Decisions Made

|  |  |  |
| --- | --- | --- |
| Date | Decision | Reason |
| 28/08/2019 | Not scraping requested Committee Information Hansard records | Records are in a different format (PDF) which means that additional custom code will need to be written to scrape and store the record in the appropriate format for analysis. This would take additional time that is not available in this project with its short timeframe. |
| 9/09/2019 | Scraping two years of data rather than one | Was ahead of schedule for scraping data and therefore had time to scrape additional records (Bills and Answers to Questions). Initially only one year of Hansard records were scraped. |
| 11/09/2019 | Document decision made and the reason to be handed up with deliverables | Useful information for AGD to identify why the project includes and doesn’t include specific functionality |
| 12/09/2019 | Scraping last two years of Question Time records | Received feedback that Question Time records are more useful to audit teams than Answers to Questions records |
| 2/09/2019 | Not using Azure | AGD do not want to use Azure due to the complications with current processes and policies. |
| 7/10/2019 | Using regular expressions in Tableau to allow multiple term and wildcard search | This appeared was the easiest way to include more complicated string search in Tableau as the software is not designed for text search. The project team will document some common regular expressions in a user guide. |
| 7/10/2019 | Not all text analytics being investigated and implemented will be included in the final dashboards given to AGD. Instead they will be submitted and documented as potential future work for the project. | Time constraints mean that not all text analytics could be properly investigated, implemented and put in a format that can be integrated into the Tableau dashboard. Therefore, work not integrated into the dashboards will be documented appropriately in the final report as potential future work for the project. |
| 7/10/2019 | User guide included in deliverables | A short user guide will be produced for the dashboards to better explain some of the more complex functionality such as the use of regular expressions in text search. |
| 11/10/2019 | Will not be including Gazette information in the project | This was identified as being another useful data source to be analysed. However, given time constraints it is not possible to include this additional data source. This potential data source will be documented as future work for the project. |
| 12/10/2019 | Will not use names of only one-word length when searching for clients in Hansard records | Search terms consisting of only one word, in the Client names spreadsheet, are too broad and would return too many irrelevant records. |
| 16/10/2019 | Subject TreeMap in Subject Summary Dashboard will only drill-down two levels of information. | Allowing the TreeMap to drill-down more than two levels results in an unreadable chart because the text is too small. |

# Appendix B: Database Schema



# Appendix C: Data Dictionary

This appendix describes the data dictionary for the seven SQL database tables as shown in Appendix A.

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| HansardID | Integer | Unique identifier for Hansard record. Same as Hansard table id |
| Name | String | Name of Hansard record |
| Parliament Name | String | Parliament name (e.g. Parliament of South Australia) |
| Parliament Num | Integer | Parliament number |
| Review Stage | String | Review stage of record (e.g. Published) |
| Session Name | String | Session name |
| Session Num | Integer | Session number |
| Venue | String | Venue proceeding was held |
| Date | Date | Date proceeding was held |
| Date Modified | Date/Time | Date and time proceeding record was last modified |
| House | String | House (e.g. Legislative Council) |
| Proceeding Type | String | Type of proceeding (e.g. Bills, Motions, Question Time) |
| Subject | String | Subject of proceeding |

Table 3: Header Table

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| ID | Integer | Unique identifier for Hansard record |
| File Name | String | Name of XML file. Includes file extension (.xml) |
| URL | String | URL to original record on Hansard webpage |
| ProcessedOn | Date/Time | Date and time that Hansard record was processed and added to database |
| Sentiment | Float | Sentiment of Hansard record text |
| Summary | String | Three-sentence summary of Hansard record using the Page Rank algorithm |
| Key Words | String | Maximum of 10 most ranked words identified from Hansard record text |
| Record Text | String | Entire record text |

Table 4: HANSARDFilesInfo Table

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| Bill ID | String | Unique identifier for Bill |
| Hansard ID | Integer | Unique identifier for Hansard record |
| Bill Name | String | Name of Bill |

Table 5: Bill Table

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| Text ID | String | Unique identifier for text |
| Talker ID | String | Unique identifier for talker |
| Hansard ID | Integer | Unique identifier for Hansard record |
| Kind | String | Kind of text (e.g. speech, question, interjection, proceeding) |
| Text | String | Transcribed speech or proceeding text |
| Word Count | Integer | Number of words in the text |

Table 6: Text Table

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| Talker ID | String | Unique identifier for talker |
| House | String | House talker belongs to (e.g. Legislative Council) |
| Talker Name | String | Talkers name or position in proceeding |
| Role | String | Role of talker (e.g. office, member) |
| Electorate | String | Electorate of talker |

Table 7: Talker Table

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| Hansard ID | Integer | Unique identifier for Hansard record |
| Question ID | Integer | Unique identifier for question |
| Talker ID | String | Unique identifier for talker that asked the question |
| Qon Num | Integer |  |
| Question | String | Subject of question |
| Date | Date | Date question was asked |

Table 8: Question Table

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| Portfolio ID | Integer | Unique identifier for portfolio |
| Hansard ID | Integer | Unique identifier for Hansard record |
| Talker ID | String | Unique identifier for talker |
| Name | String | Name of portfolio (e.g. Minister for Health and Wellbeing) |

Table 9: Portfolio Table

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| Hansard ID | Integer | Unique identifier for Hansard record |
| Text ID | String | Unique identifier for text |
| Term | String | Key term of interest to audit team |
| Audit Team | String | Name of audit team at Auditor-General's Department |

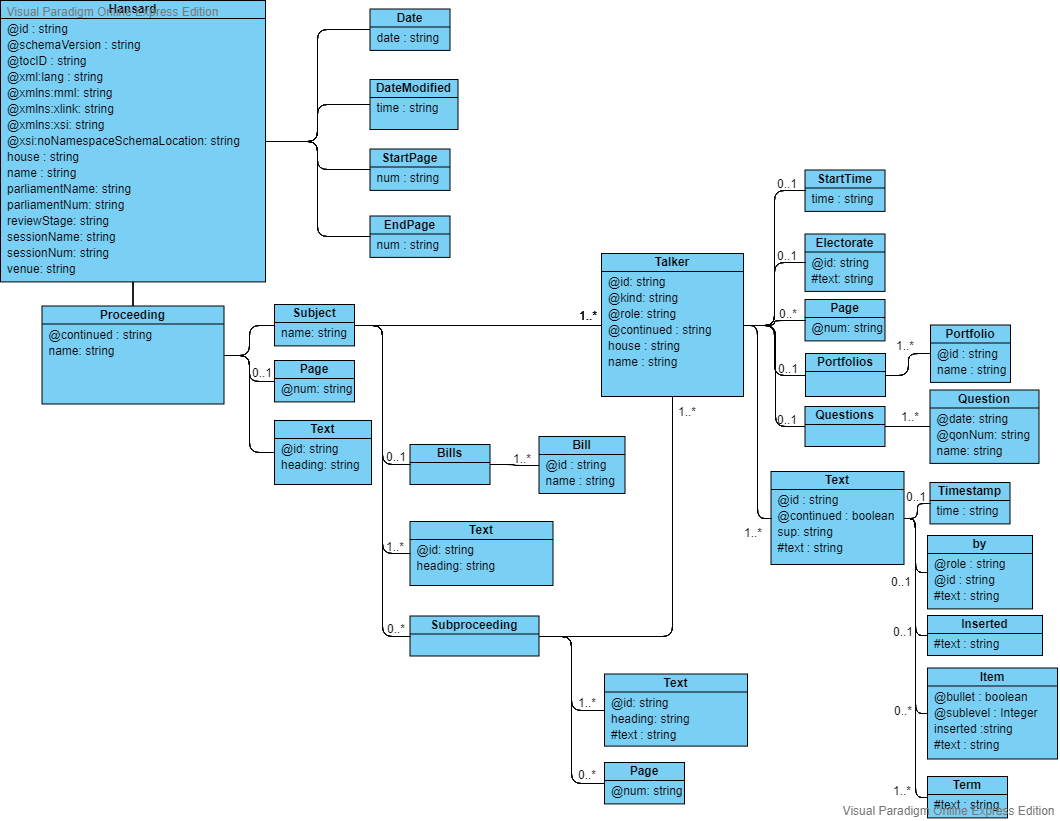
Table 10: Key Terms Table

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| File Name | String | Name of XML file. Includes file extension (.xml) |
| Text ID | String | Unique identifier for text |
| AGD Client | String | Common name of client |
| AGD Formal | String | Formal name of client |
| Client Type | String | Type of client (e.g. Local councils) |

Table 11: Client Mentions Table

# Appendix D: Hansard XML Record Structure

The UML diagram below shows the structure of the XML records that were downloaded from the Hansard webpage. This diagram was created to assist with understanding the Hansard records.



# Appendix E: Dashboard Wireframe Mockups

The figures below show the wireframe mockups of the proposed dashboards. There were presented to the Project Reference Group on 11th September 2019 as part of requirement elicitation.

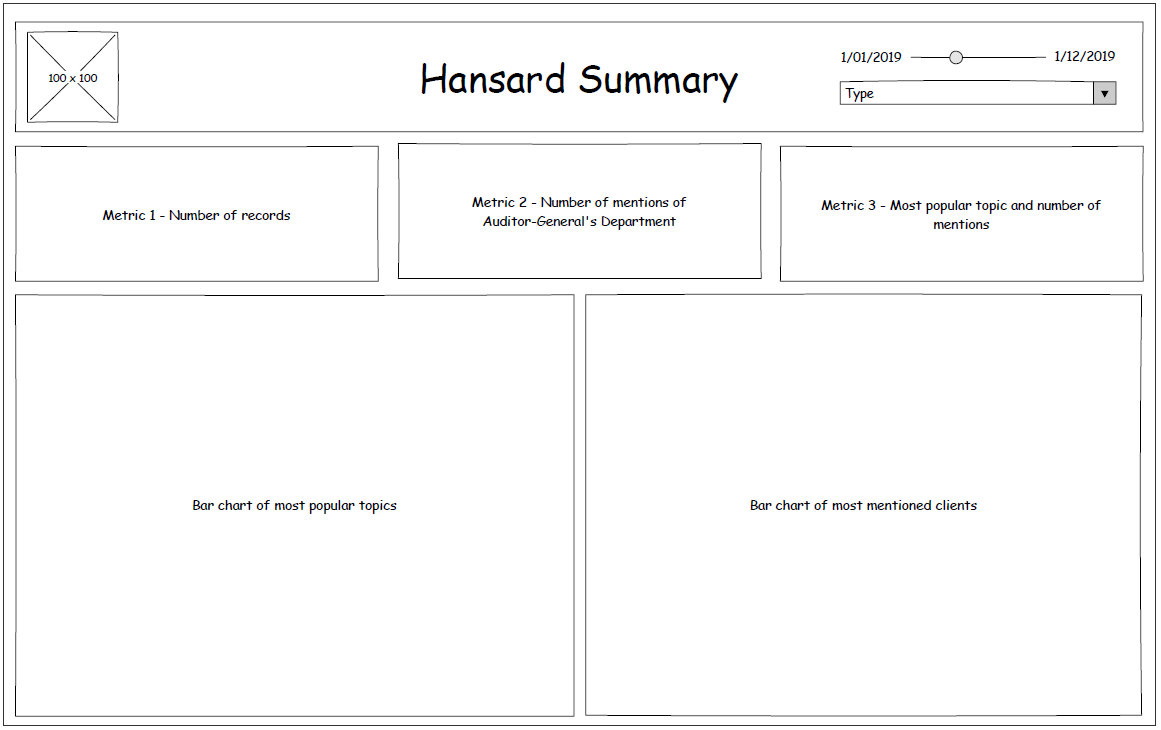


Figure 23: Hansard Summary Dashboard Wireframe

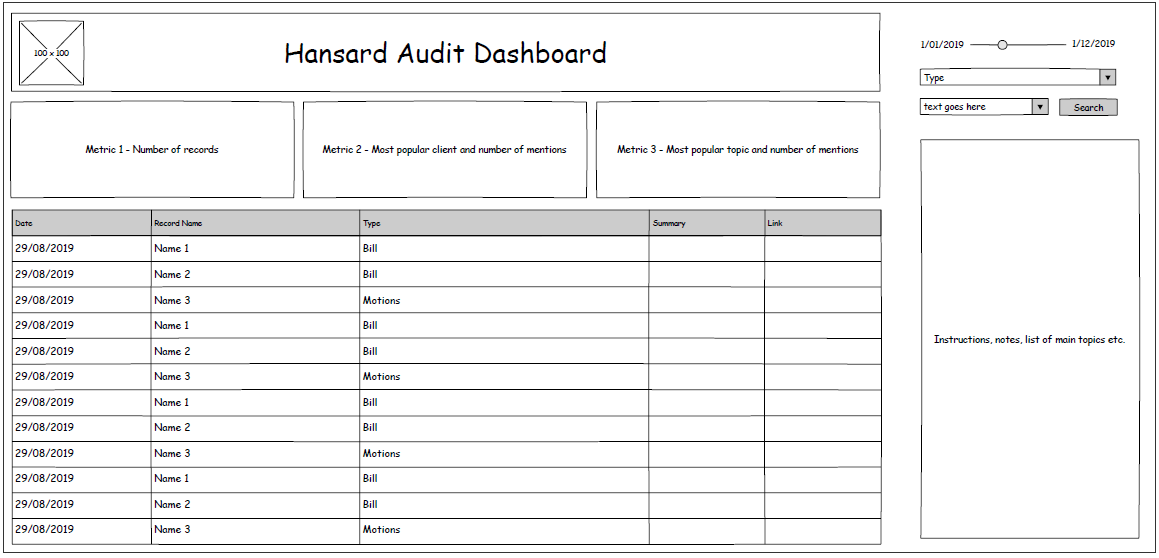


Figure 24: Hansard Audit Dashboard Wireframe

# Appendix F: Dashboard Charts

This appendix contains screenshots of charts from the final dashboards.

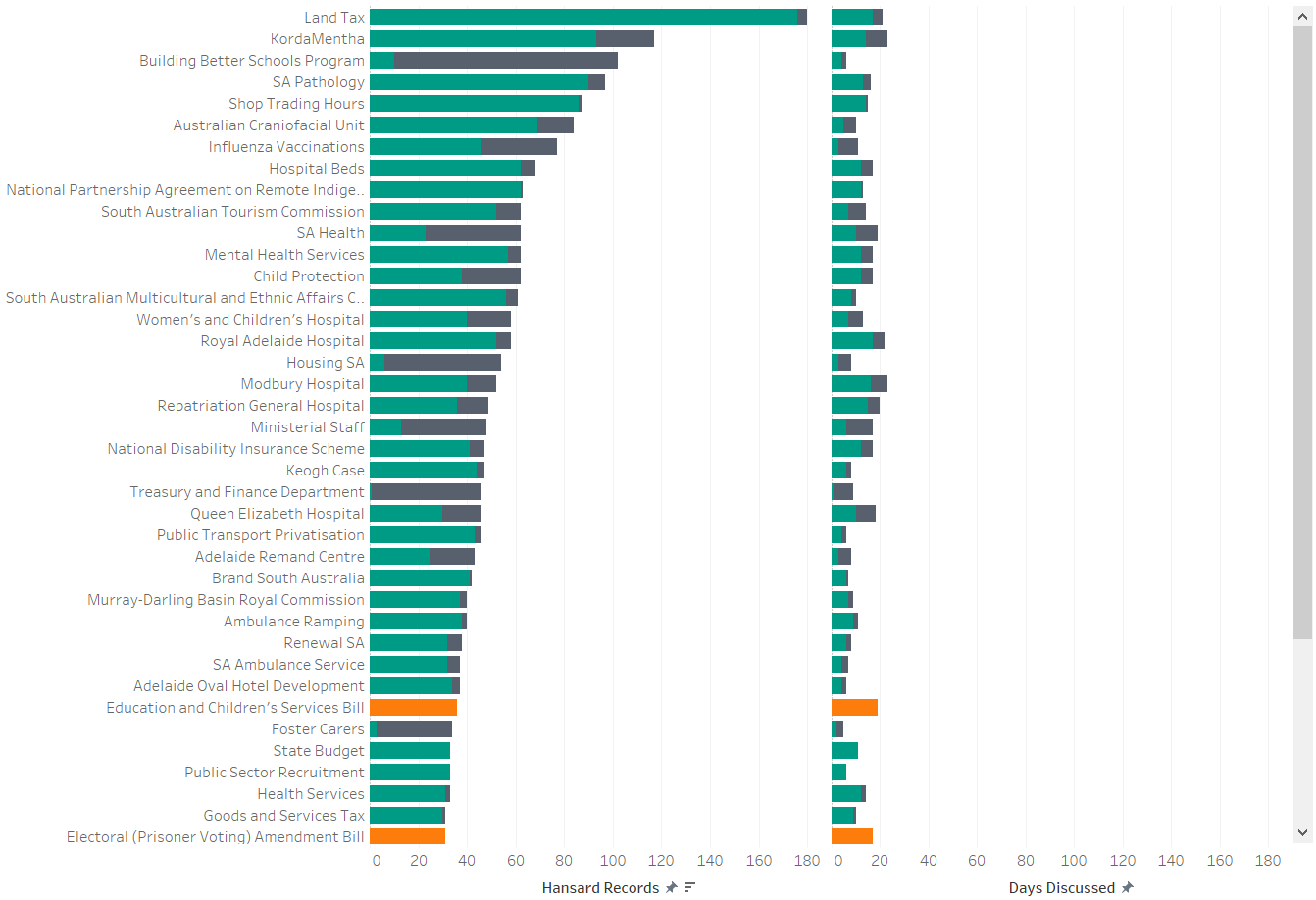


Figure 25: Top 50 Clients bar chart in Summary Dashboard

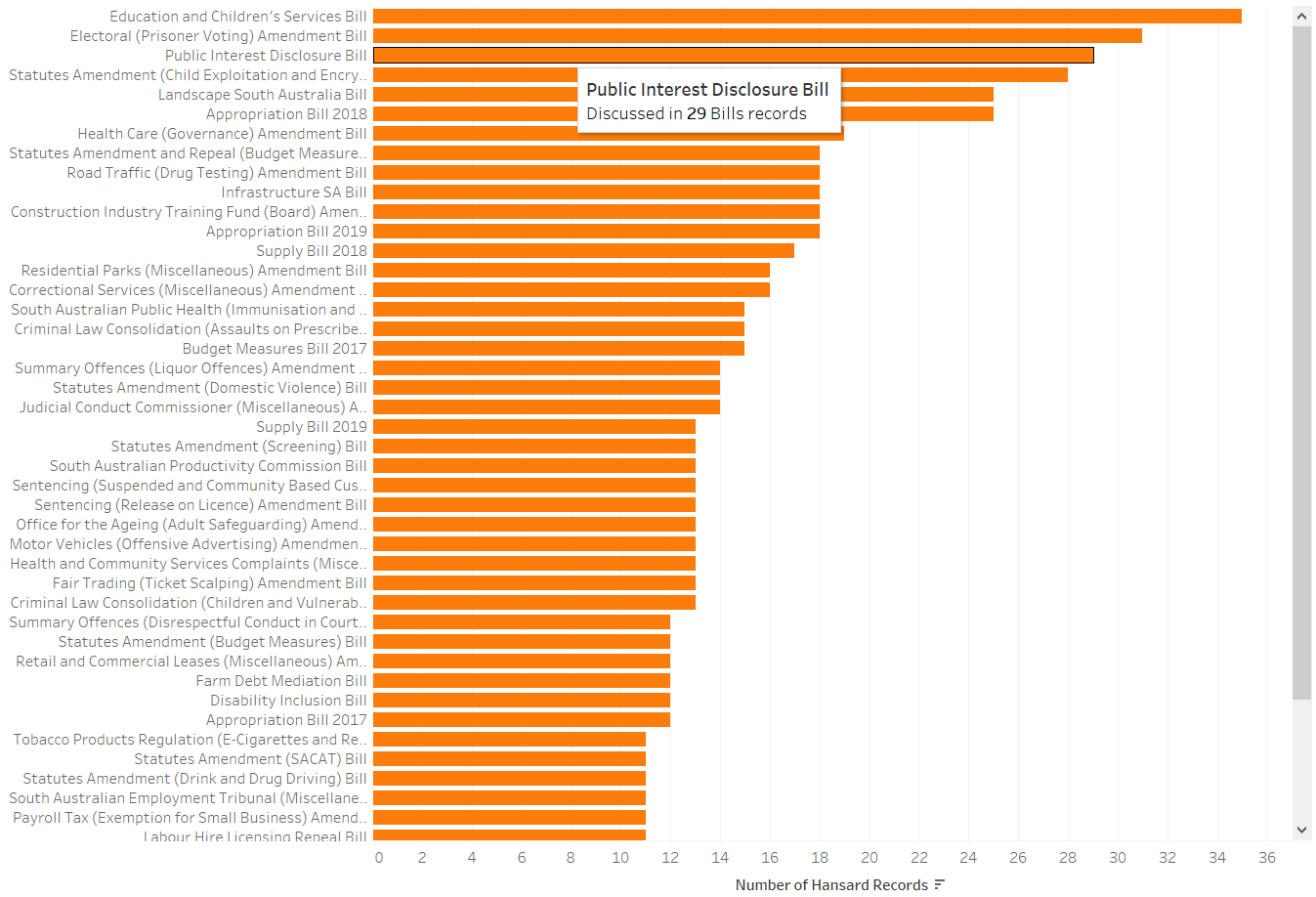


Figure 26: Top 50 Bills bar chart in Summary Dashboard

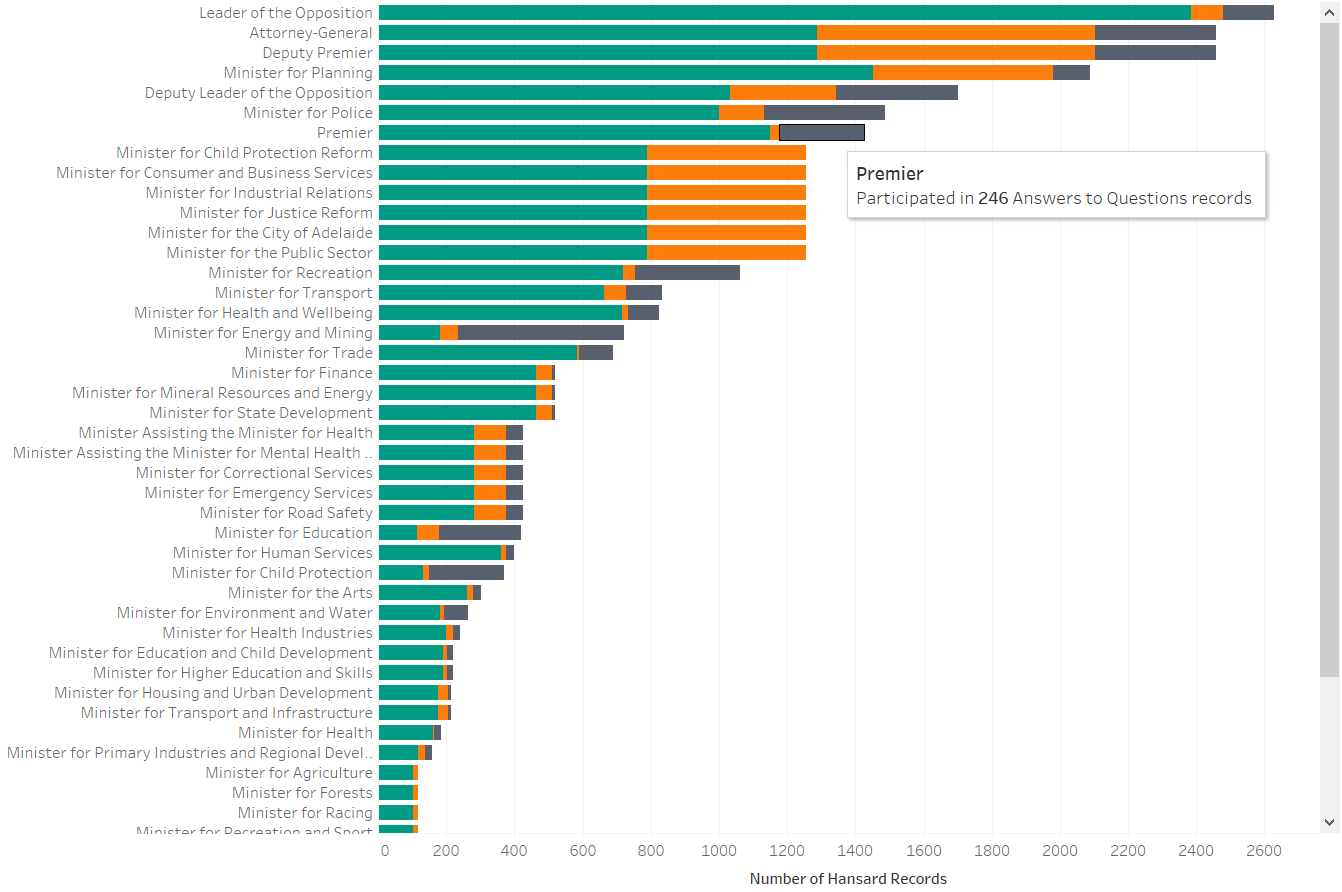


Figure 27: Top 50 Portfolios bar chart in Summary Dashboard

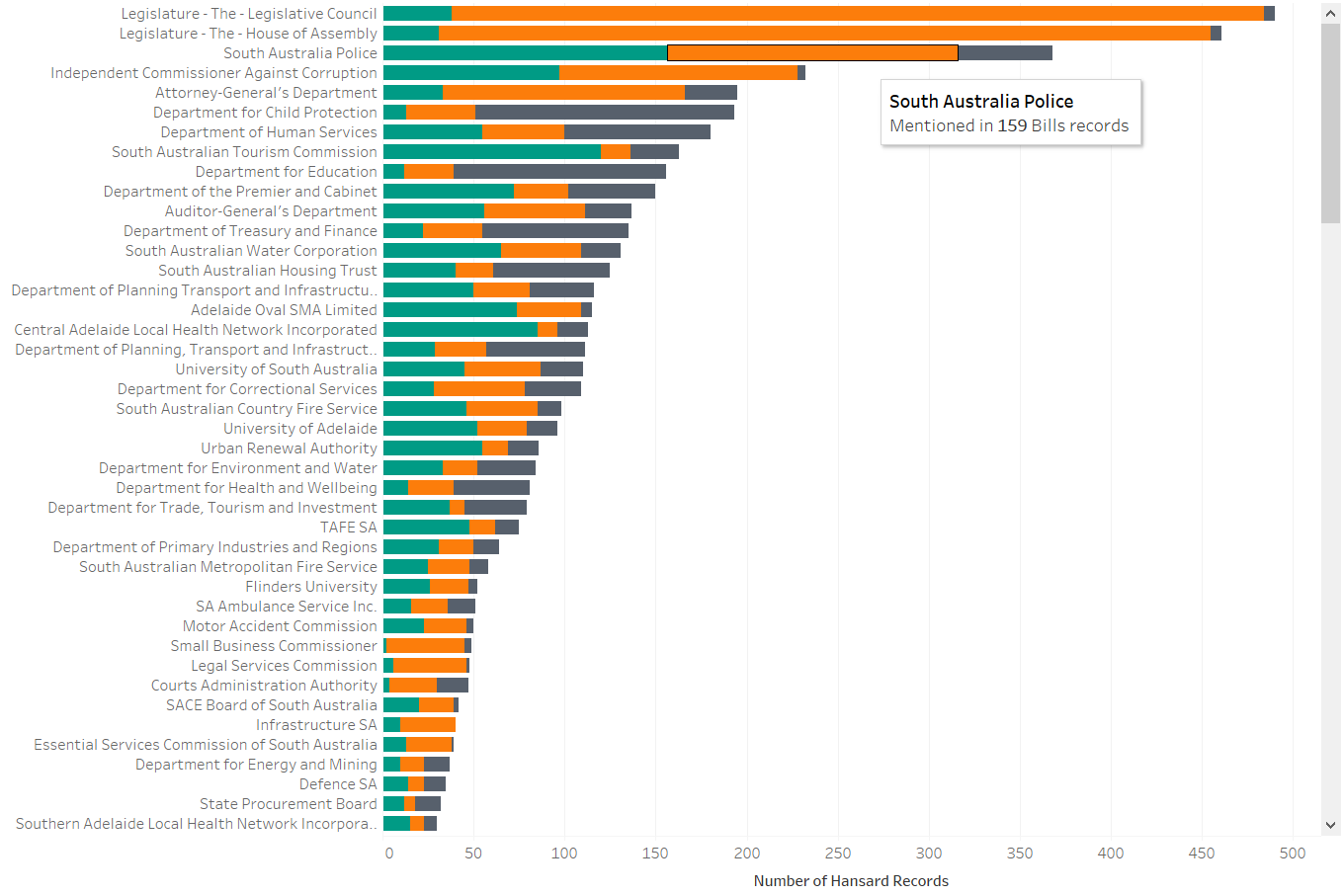


Figure 28: Top Clients bar chart in Summary Dashboard

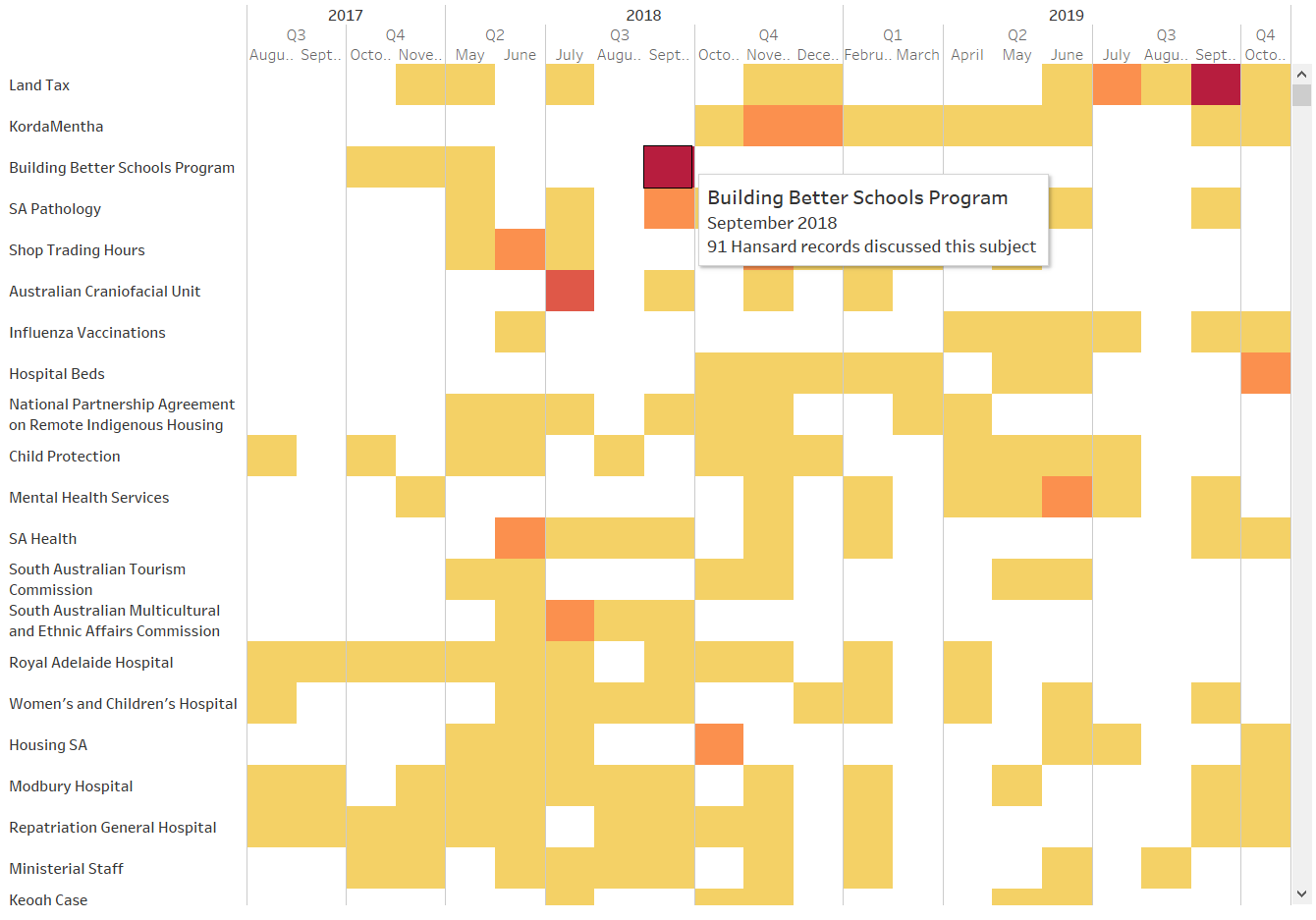


Figure 29: Subjects Discussed Over Time heatmap in Summary Dashboard

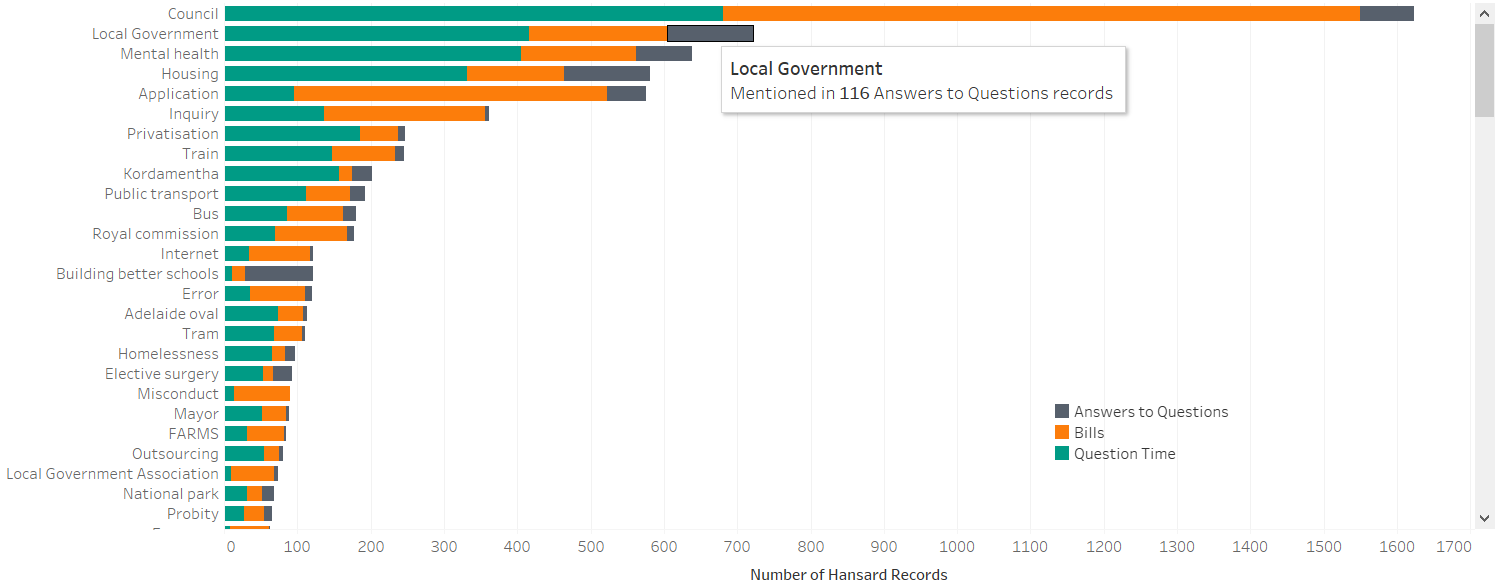


Figure 30: Key Term Search bar chart in Audit Team Dashboard

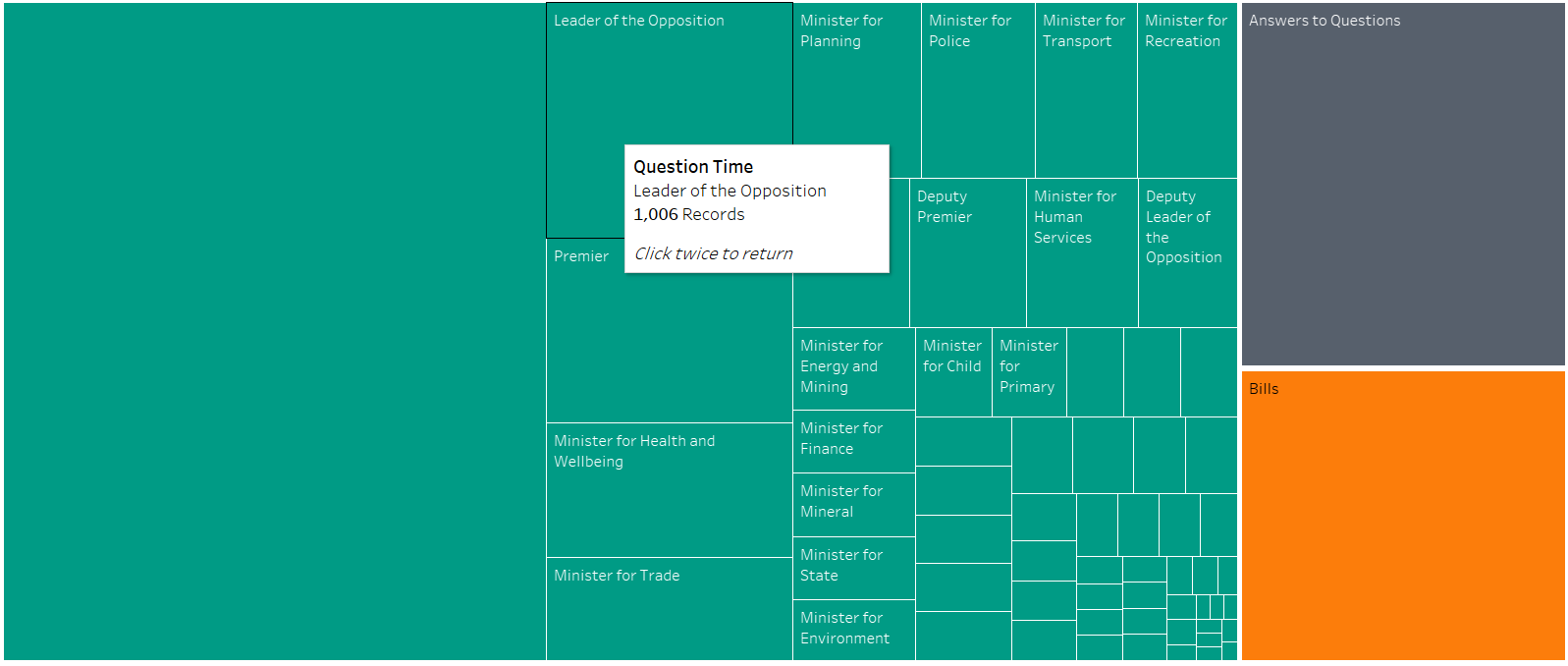


Figure 31: Subject treemap in Subject Overview Dashboard

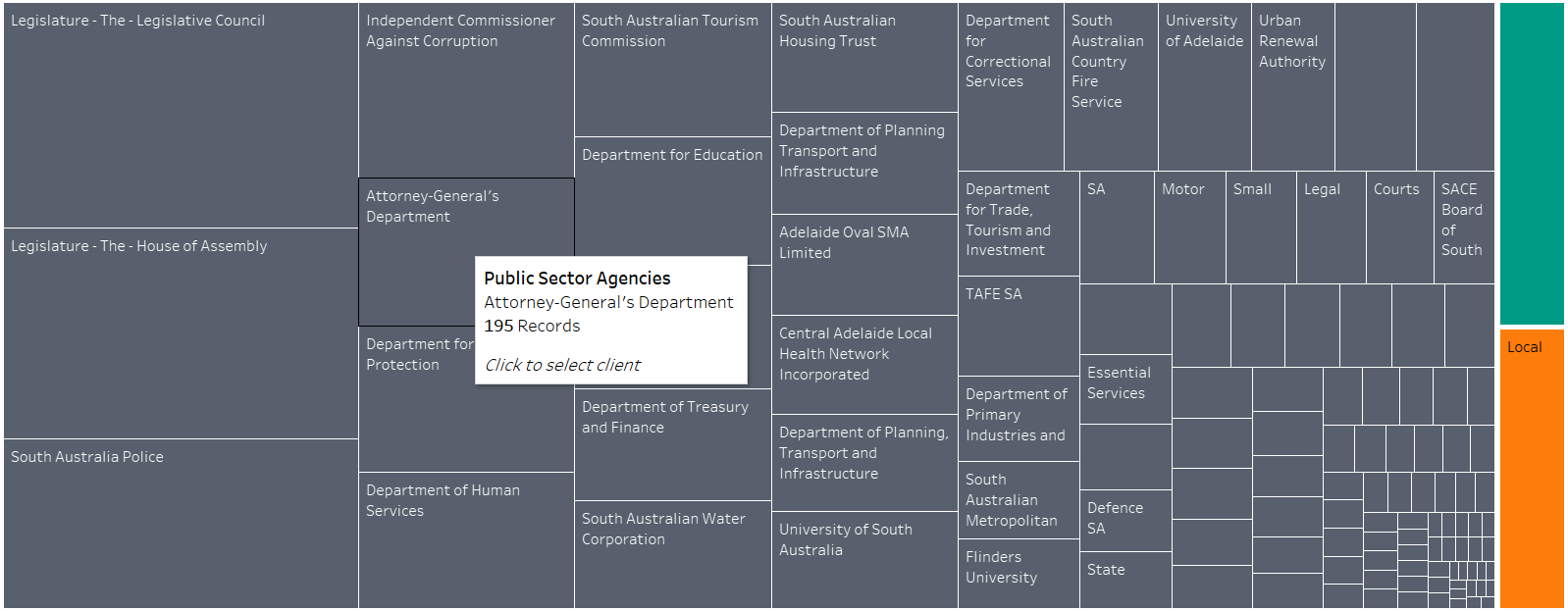


Figure 32: Client treemap in Client Overview Dashboard

# Appendix G: Document Summary Example

**Original text from record HANSARD-10-21310 (Parliament of South Australia, 2017):**

*Bills*

**STATUTES AMENDMENT (HEAVY VEHICLES REGISTRATION FEES) BILL**

*Second Reading*

Adjourned debate on second reading.

(Continued from 1 June 2017.)

**The Hon. J.M.A. LENSINK (16:35):** I rise to indicate opposition support for this bill, which is part of the harmonisation of national laws which relate to heavy vehicles. In particular, this bill amends the Highways Act and the Motor Vehicles Act so that South Australia can meet its obligations under the Heavy Vehicle National Law (South Australia) Act 2013, which contains the national law as a schedule. This bill provides for the creation of a national heavy vehicle regulator. For the benefit of readers, heavy vehicles are defined as trucks with a gross vehicle mass of 4.5 tonnes or more.

The section relating to registration of the national law has not commenced yet, so heavy vehicle registrations remain under state legislation; however, those jurisdictions which are participants, which I understand includes all states and territories except for the Northern Territory and WA, are governed by model law approved by the national Transport and Infrastructure Council, made up of state and territory ministers. Vehicle registration charges are currently calculated on the basis of road user charge and regulatory charge components.

Participating jurisdictions have agreed that the revenue which is collected will be transferred to the regulator to undertake its duties. Amendments to the Motor Vehicles Act clarify that deductions from concessional registration charges for people living in remote areas and from primary producers will be taken from the roads component rather than the regulatory component of the fees provided to the regulator's fund.

This bill is, in effect, a stopgap to cover arrangements until the other arrangements are completed. I am advised that registration fees will not increase, but a portion will be provided to the NHVR instead of going to the Highways Fund. With those comments, I commend the bill to the house.

Debate adjourned on motion of Hon. J.E. Hanson.

**Three-sentence summary using PageRank algorithm:**

'This bill provides for the creation of a national heavy vehicle regulator. Statutes Amendment (Heavy Vehicles Registration Fees) Bill. In particular, this bill amends the Highways Act and the Motor Vehicles Act so that South Australia can meet its obligations under the Heavy Vehicle National Law (South Australia) Act 2013, which contains the national law as a schedule'

# Appendix H: SSIS - Visual Basic script to read in the file names

#Region "Imports"

Imports System

Imports System.Data

Imports System.Math

Imports Microsoft.SqlServer.Dts.Runtime

#End Region

<Microsoft.SqlServer.Dts.Tasks.ScriptTask.SSISScriptTaskEntryPointAttribute()>

<System.CLSCompliantAttribute(False)> \_

Partial Public Class ScriptMain

Inherits Microsoft.SqlServer.Dts.Tasks.ScriptTask.VSTARTScriptObjectModelBase

Public Sub Main()

Dts.Variables("FileName").Value = System.IO.Path.GetFileName(Dts.Variables("FileName").Value.ToString())

Dts.TaskResult = ScriptResults.Success

End Sub

#Region "ScriptResults declaration"

'This enum provides a convenient shorthand within the scope of this class for setting the

'result of the script.

'This code was generated automatically.

Enum ScriptResults

Success = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Success

Failure = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Failure

End Enum

#End Region

End Class

# Appendix I: T-SQL scrips to check if the file name exists in DB (SSIS)

DECLARE @FileName VARCHAR(255)

SET @FileName=''

IF EXISTS (SELECT 1

FROM [dbo].[HANSARDFilesInfo]

WHERE FileName = @FileName)

BEGIN

SELECT 1 AS FileExistitsFlg

END

ELSE

BEGIN

SELECT 0 AS FileExistingFlg

END

# Appendix J: SQL Command to truncate the final tables in SSIs

TRUNCATE TABLE dbo.DistinctTalker

GO

TRUNCATE TABLE dbo.FinalBill

GO

TRUNCATE TABLE dbo.FinalHeader

GO

TRUNCATE TABLE dbo.FinalQuestion

GO

TRUNCATE TABLE dbo.FinalText

GO

TRUNCATE TABLE dbo.FinalPortfolio

# Appendix K: SQL Command to join tables to get the HansardID in SSIS

SELECT

HANSARDFilesInfo.ID as HansardID,

Bill.BillID,

Bill.BillName

FROM

HANSARDFilesInfo

inner JOIN Bill ON

Bill.HansardID= HANSARDFilesInfo.FileName

ORDER BY

Bill.BillID

# Appendix L: SSIS Variable Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description of the variable** | **Name** | **Scope** | **Data Type** | **The value needs to be changed if the package moves?** |
| **Archive folder path, which contains all the laded XMLs** | **AchiveFolder** | **Package** | **String** | **Yes** |
| **Rejected folder path, which contains all the rejected XMLs (duplicates)** | **RejectFolder** | **Package** | **String** | **Yes** |
| **Source folder path for Bill XMLs before they being loaded** | **DirBill** | **Package** | **String** | **Yes** |
| **Source folder path for Questions & Answers XMLs before they being loaded** | **DirQA** | **Package** | **String** | **Yes** |
| **Source folder path for Question Time XMLs before they being loaded** | **DirQT** | **Package** | **String** | **Yes** |
| **Renaming variable after they moved to the Archive folder** | **FileDestPath** | **Package** | **String** | **No** |
| **Name of the XML files** | **FileName** | **Package** | **String** | **No** |
| **Renaming variable after the rejected XMLs are moved to the Rejected folder** | **FileRejectedPath** | **Package** | **String** | **No** |
| **Full path and file name of Bill XMLs** | **FileSourcePathBill** | **Package** | **String** | **No** |
| **Full path and file name of Question & Answers XMLs** | **FileSourcePathQA** | **Package** | **String** | **No** |
| **Full path and file name of Question Time XMLs** | **FileSourcePathQT** | **Package** | **String** | **No** |
| **URL** | **URL** | **Package** | **String** | **No** |
| **XMLs after XSLT transformation** | **varXMLOutput** | **Package** | **String** | **No** |

# Appendix M: List of Topics Generated by LDA

|  |  |  |
| --- | --- | --- |
| **Labeled Topic** | **prevalence** | **Top five terms of the topic** |
| royal\_hospital | 5.649 | hospital, health, beds, patients, medical |
| mental\_health | 5.036 | health, sa, services, mental, mental\_health |
| domestic\_violence | 5.032 | chamber, original, report, public, parliament |
| regional\_development | 4.739 | regional, development, industries, primary, regional\_development |
| elective\_surgery | 4.649 | budget, million, estimates, services, additional |
| transport\_infrastructure | 4.521 | transport, infrastructure, transport\_infrastructure, local, road |
| liberal\_party | 4.257 | liberal, disability, party, election, people |
| emergency\_services | 4.182 | police, services, hours, emergency, emergency\_services |
| remote\_aboriginal | 4.175 | funding, federal, aboriginal, agreement, housing |
| abc\_radio | 4.13 | work, workers, cent, employment, drug |
| chief\_executive | 4.058 | chief, executive, chief\_executive, report, contract |
| child\_protection | 3.915 | child, protection, child\_protection, care, children |
| diesel\_generators | 3.805 | energy, cost, plan, generators, policy |
| safework\_sa | 3.726 | public, deputy, sa, information, investigation |
| land\_tax | 3.721 | land, tax, land\_tax, commission, royal |
| kangaroo\_island | 3.715 | island, kangaroo, centre, kangaroo\_island, conservation |
| environment\_water | 3.639 | water, environment, environment\_water, industry, skills |
| brand\_sa | 3.636 | funding, email, sa, cut, programs |
| energy\_mining | 3.614 | energy, david, concerns, mining, unit |
| treasury\_finance | 3.523 | public, services, working, treasury, positions |
| housing\_sa | 3.517 | housing, sa, housing\_sa, trust, keogh |
| aged\_care | 3.433 | public, care, home, safety, aged |
| recreation\_sport | 3.285 | sport, recreation, recreation\_sport, racing, sport\_racing |
| ethnic\_affairs | 3.118 | assistant, board, community, consultation, romaldi |
| education\_child | 2.925 | education, building, program, schools, plans |

# Appendix N: Gantt Chart

