Hansard Data Mining

Final report

Aaron Butler

Bipin Karki

Katherine Noack

Mahmoud Yousefi

November 1, 2019

Contents

[Acknowledgements 3](#_Toc23488943)

[Executive Summary 3](#_Toc23488944)

[Introduction 4](#_Toc23488945)

[Background 4](#_Toc23488946)

[Project Objectives 4](#_Toc23488947)

[Scope 4](#_Toc23488948)

[Deliverables 6](#_Toc23488949)

[Project Methodology 6](#_Toc23488950)

[Project Plan and Alterations 7](#_Toc23488951)

[Project Plan Changes 7](#_Toc23488952)

[Task Changes 8](#_Toc23488953)

[Deliverables 8](#_Toc23488954)

[Project Architecture 8](#_Toc23488955)

[Extract, Transform and Load 10](#_Toc23488956)

[Dashboards 13](#_Toc23488957)

[Text Analytics 18](#_Toc23488958)

[Project Performance Assessment 25](#_Toc23488959)

[Performance against critical success factors 25](#_Toc23488960)

[Performance against schedule 25](#_Toc23488961)

[Performance against outputs 26](#_Toc23488962)

[Lessons Learned 26](#_Toc23488963)

[Recommendations and Future Work 26](#_Toc23488964)

[Conclusion 27](#_Toc23488965)

[References 28](#_Toc23488966)

[Appendix A: Database Schema 30](#_Toc23488967)

[Appendix B: Hansard XML Record Structure 31](#_Toc23488968)

[Appendix C: Dashboard Wireframe Mockups 32](#_Toc23488969)

[Appendix D: Dashboard Charts 33](#_Toc23488970)

[Appendix E: Document Summary 37](#_Toc23488971)

[Appendix F: List of Topics 38](#_Toc23488972)

[Appendix G: Gantt Chart 39](#_Toc23488973)

# 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. Finally, we would like to thank the University of South Australia for providing this opportunity to work in real-world scenario.

# Executive Summary

The Hansard Data Mining project is focused on providing the Auditor-General's Department (AGD) with an automated system to better understand, interrogate and summarise text data from South Australia’s Parliament discussions.

The project was built and tested upon two years of data extracted from the Hansard website for different proceeding types: “Bills”, “Question Time” and "Answers to Questions”. To better understand these records of proceeding types, multiple visualisation and text analytics techniques were implemented such as text summarisation, sentiment analysis, and topic modelling. Five dashboards were developed using Tableau 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 (SA) 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, performance, 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 SA Parliament. Hansard is a division of SA 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 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 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
* 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 AGD

#### Exclusions

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

* Advanced text analytics techniques
* Data sources outside of those listed in scope
* Predictions or forecasting from scraped data
* Web application development
* Evaluation of third-party software
* Evaluation of Hansard data accuracy

#### Approved Scope Changes

There were several approved changes to the scope 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 further described in the Sentiment Analysis section.

Another approved scope change was to provide a 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 text search, were provided in a user guide.

At the request of the Project Reference Group (PRG) the reasons why certain decisions were made was documented in a table 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 it was determined that a Project Review and Closure report was not required. It was determined that this project had sufficient documentation already.

## 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 | Instructions on how to setup and run the provided code, database, and dashboards on AGD systems. |
| User Guide | 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. |
| 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. |
| 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. Refer database schema in Appendix A. |
| 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 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 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 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.

Dashboard versions were packaged and emailed to James Baker (Project Owner) periodically so that they could be published at AGD for testing by the PRG. The PRG at AGD consisted of members from three different audit teams who provided user feedback and requirements (Local Government, IT and Performance audit teams). 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 was the week starting 14th October due to an upgrade of Tableau Server being required

# Project Plan and Alterations

The entire project was developed based on the initial project plan. However, there were several changes to the project plan that were necessary to meet client’s requirements.

## Project Plan Changes

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 PRG from AGD. In the initial development phase, there were six sprints with each sprint completing 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 at the end of each sprint. This resulted in tasks such as scraping, text analytics and ETL being completed across multiple sprints with iterative development increasing functionality over time. The purpose of this was to deliver example dashboards as early as possible to get feedback from the PRG. Figure 1 shows the sprint dates of the development phase.

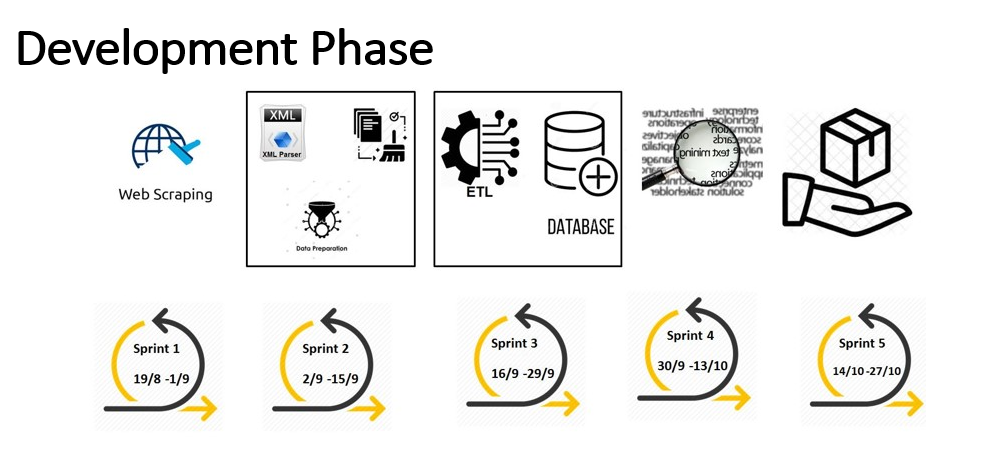


Figure 1: Development phase of Hansard data mining project

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

## Task Changes

Communication with different stakeholders in the PRG resulted in a variety of requirements. Initially, the scope was to scrape and analyse two proceeding types (“Bills” and “Answers to Questions”), but later “Question Time” was also added as this was considered more important information by AGD. Another change in task was to scrape two years of data instead of one year. Other approved scope changes were discussed previously in Background.

# Deliverables

This section describes the main deliverables of the project.

## Project Architecture

After several meetings with the client, the high-level architecture 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 and had experience working with SSIS
3. The client also had Microsoft SQL Server and therefore was selected to store Hansard data
4. Text analytics performed using R and Python
5. Hansard dashboards and visualisations developed in Tableau as per client's requirements

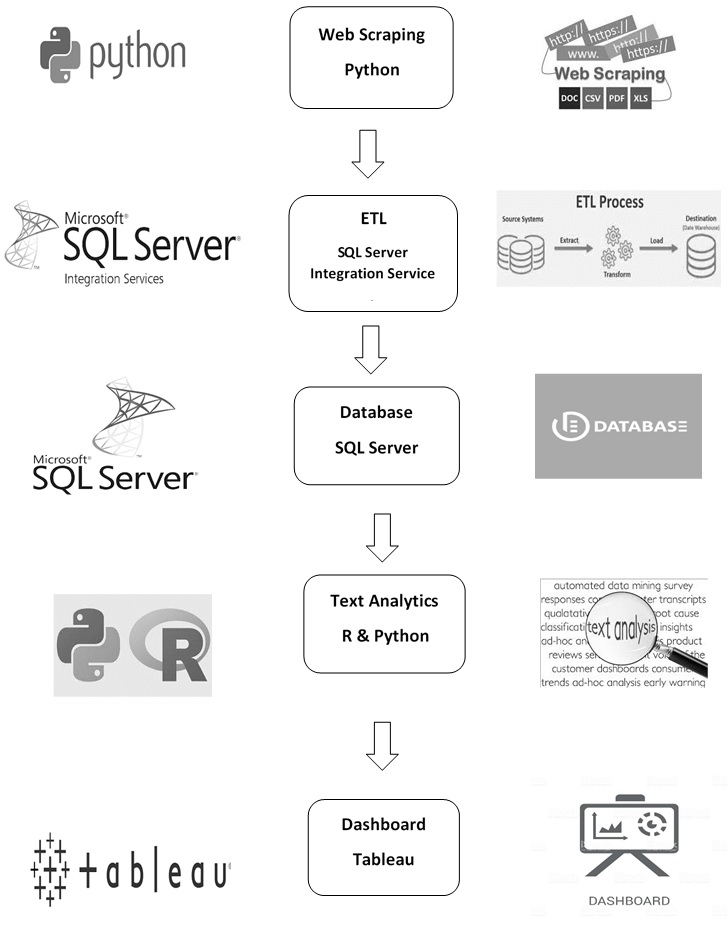


Figure 2: High-Level Architecture

**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 Hansard is more than sixty thousand. To access individual files, 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 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 (2/08/2017 to 1/08/2019) from the Hansard website in XML format. Within these two years Hansard had stored 8,855 records for three relevant proceeding types. In the second stage, the web scraper was set to automatically download records on weekly basis. As of 30th October 2019, the database contains 9,962 records from 2/08/2017 to 17/10/2019.

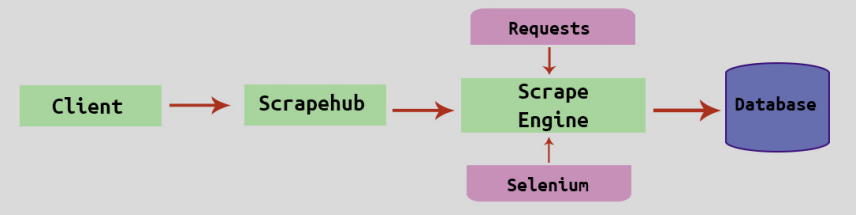


Figure 3: Web Scraping (Chaulagain et al., 2017)

Figure 3 depicts the flow of the Hansard web scraper. 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. All code was written in Python and the process automated on AGD systems. The scraper efficiently handles errors, such as the absence of XML files to be downloaded, and checks for duplicates.

## Extract, Transform and Load

A SSIS package was developed to automate the ETL process, from scraping data to storing the record data and text analysis results into a SQL Server database. 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

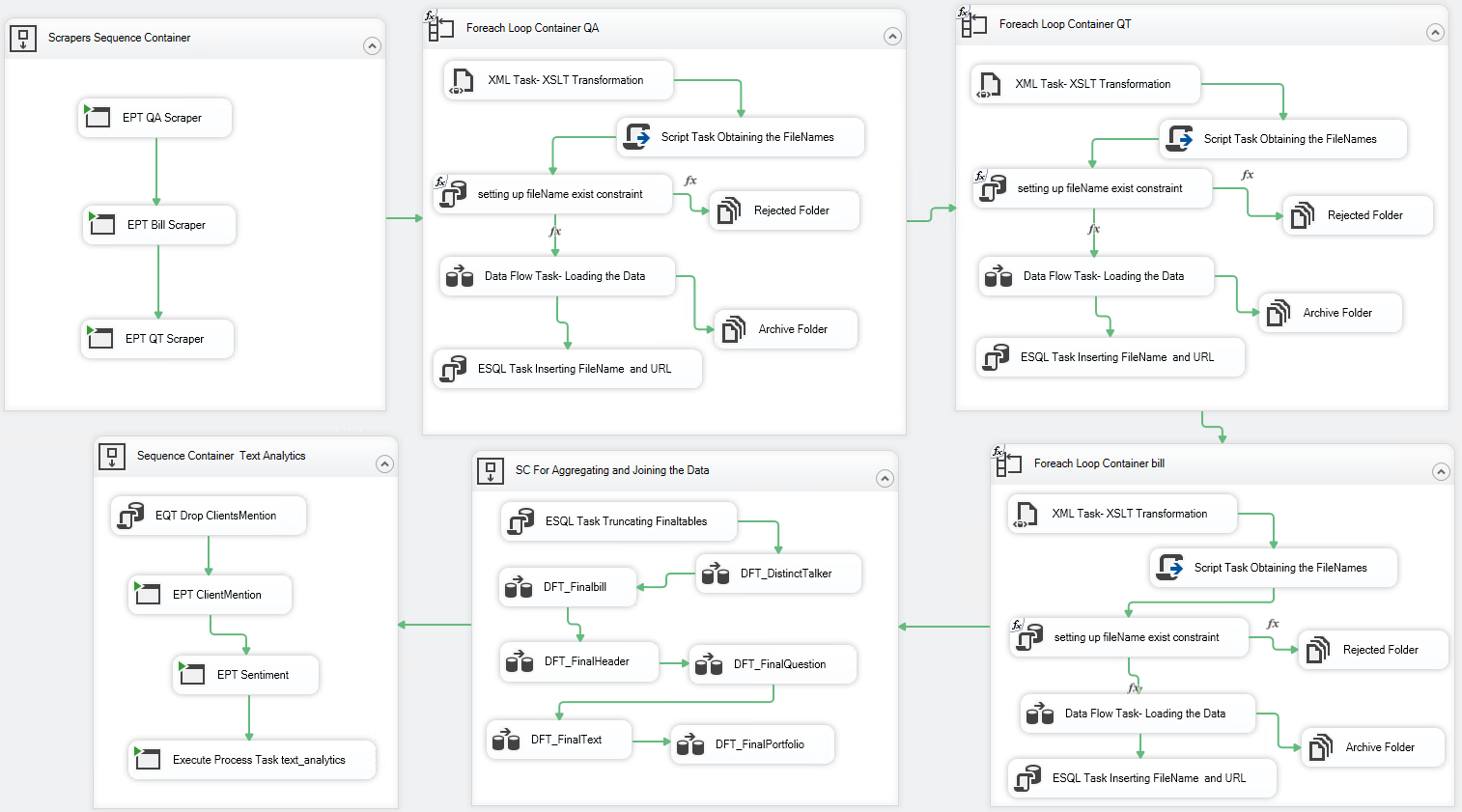


Figure 4: Hansard SSIS package overview

#### 1. Scraping module

This module runs Python scrapers for the three proceeding types (Figure 5). The destination folder to download the XML files is the input folder for the following modules.

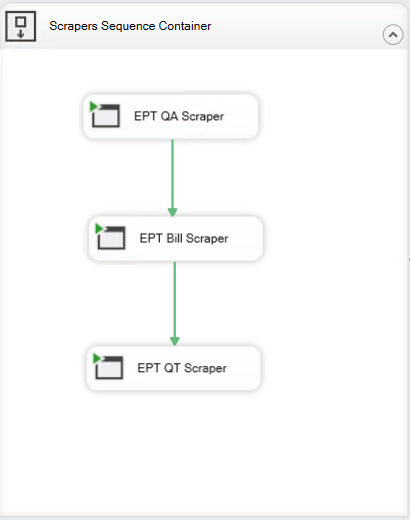


Figure 5: Scraping Module

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

This module iterates through the XML files scraped from the Hansard website and uses XSLT (eXtensible Stylesheet Language Transformations) to transform the XML into a more readable structure. The original XML record had numerous XML elements which were not relevant to this project (e.g. page numbers) which was not necessary to extract and store in database. The XSLT was designed for all 15 proceeding types currently available in Hansard instead of only three proceeding types used in this project. This enables the XSLT to be applied to additional proceeding types that may be added to the project in the future. Next the transformed were loaded into relevant staging tables (Header, Question, Bill, Portfolio, Talker and Text) and the unique XML file name added as a way to identify individual records.

This module avoids loading duplicate files using a mechanism to archive and reject duplicate files. First, the mechanism gets the XML file names and checks whether the file already exists in the database. If it exists, the file is moved to the rejected archive folder and renamed, otherwise the data is loaded 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 are moved to the Reject folder and renamed.

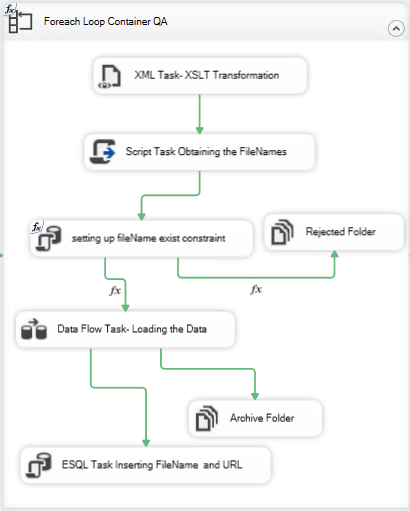


Figure 6: Staging table module

#### 3. Load data into target tables

This module eliminates redundant data and updates the final target tables to include the Hansard ID variable, rather than File Name, to match the database schema (Appendix A). This ensures that no duplicate records are loaded into the tables. For example, the staging table for talker information (TalkerStaging) contains duplicate talkers which are removed to produce the final DistinctTalker table. The final record tables are: DistinctTalker, FinalBill, FinalHeader, FinalQuestion, FinalText and FinalPortfolio.

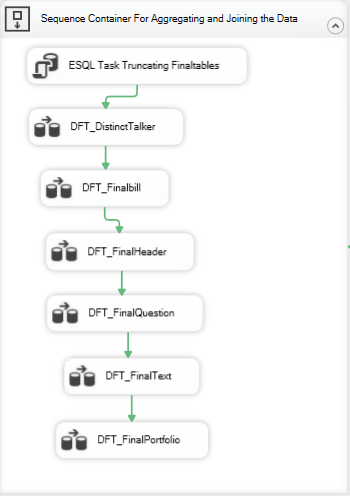


Figure 7: Target table module

#### 4. Store text analytics results into database

This module automatically runs R and Python text analytics code and stores the results in the database. The text analytics executed by this project are:

* Client Mentions: Which text mentions AGD clients listed in a spreadsheet
* Summary for record
* Word count of text
* Key words of record
* Audit Team Key Terms: Which text mentions key terms listed in a spreadsheet
* Sentiment of record

This code is executed by SSIS which adds the results to different tables in the HANSARD database (e.g. client mentions results added to ClientMentions table).

#### Deployment and automation of the package

After designing and testing the package, it was deployed to SQL Server Management Studio. As per the client’s request, the package has been scheduled to run automatically (weekly) at 12AM Saturday.

## Dashboards

Five proof-of-concept dashboards were developed using the visualisation software Tableau (Tableau, 2019). Tableau can query different data sources such as relational databases and spreadsheets and can generate variety of visualisations. Tableau 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 (PRG) 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 PRG. The dashboard requirements and their current progress status were documented in a spreadsheet which was available to all members of the project team and PRG.

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 D 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.

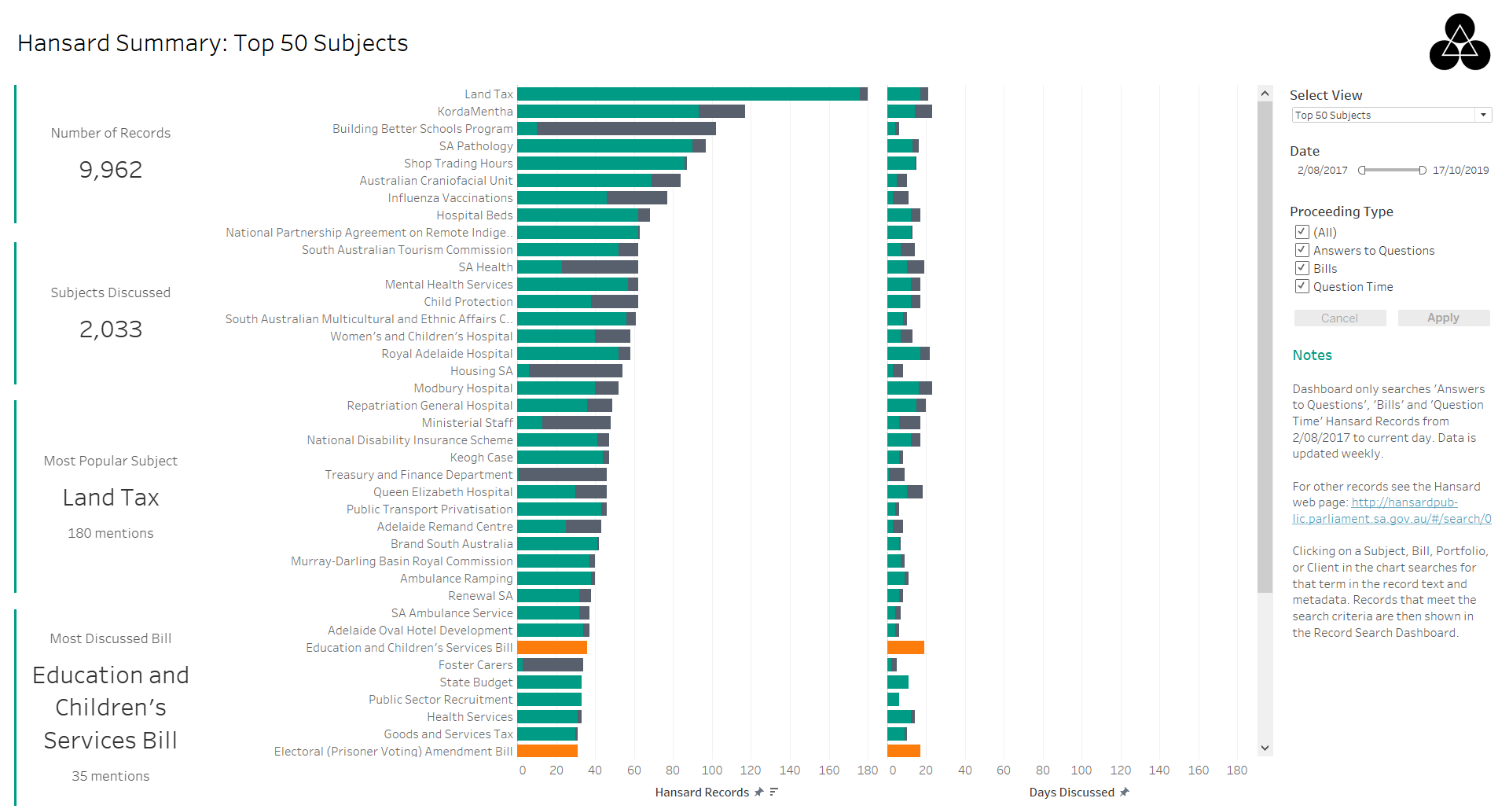


Figure 8: Summary Dashboard

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 (Figure 9) 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 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.

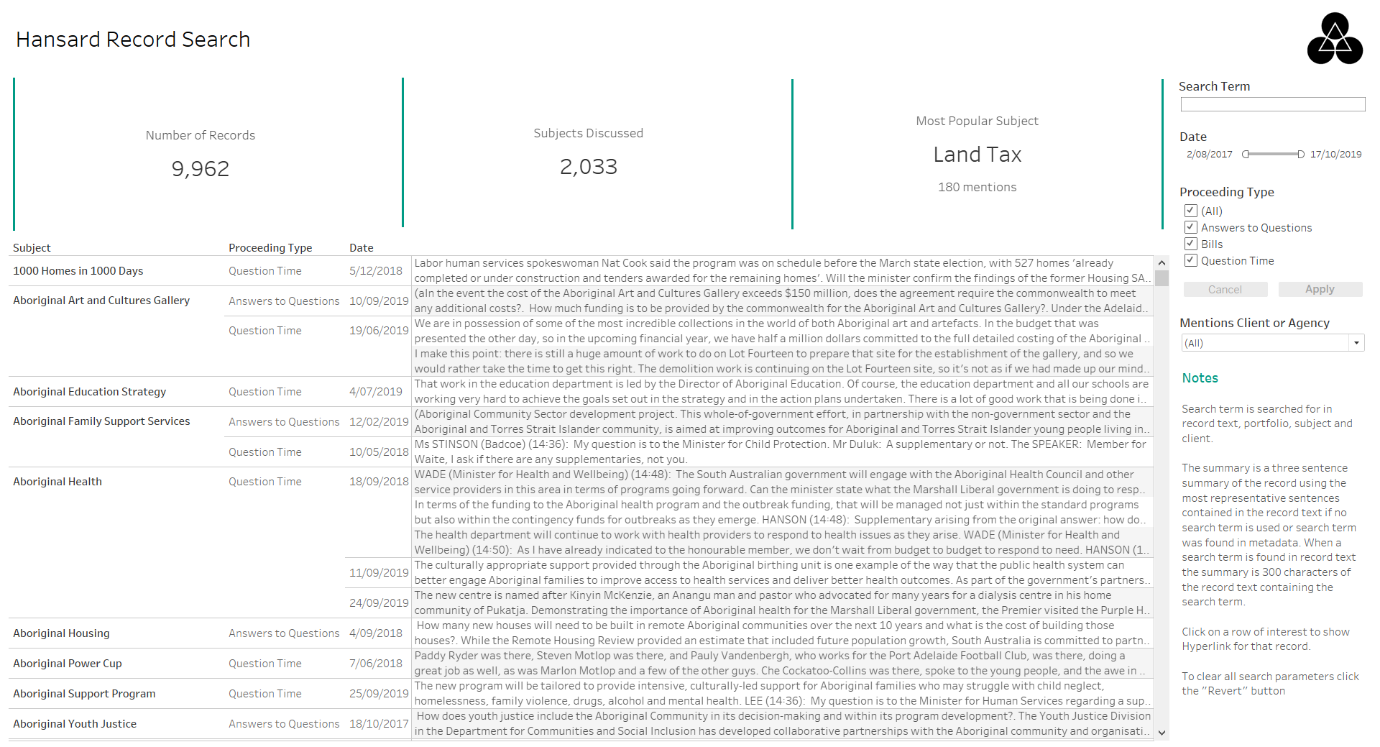


Figure 9: Record Search Dashboard

Hovering over text summary of a record provides additional information, such as key words and sentiment, which helps user to identify the importance of the record for auditing (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.

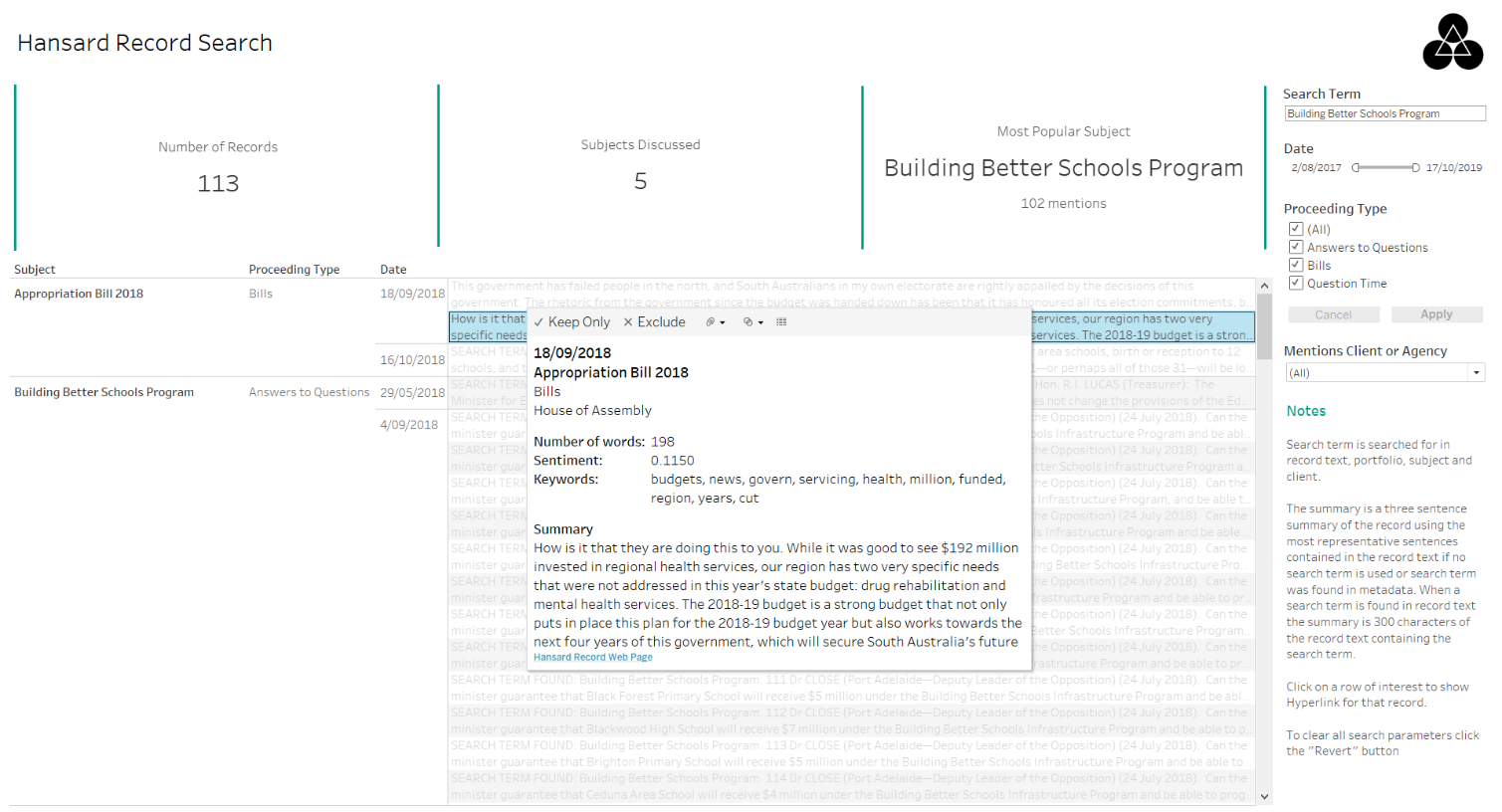


Figure 10: Record Search Dashboard (with search term and tooltip)

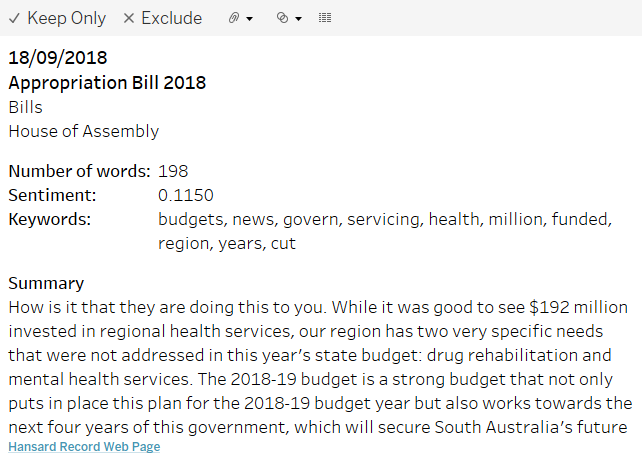


Figure 11: Record Search Dashboard Tooltip

#### Audit Team Dashboard

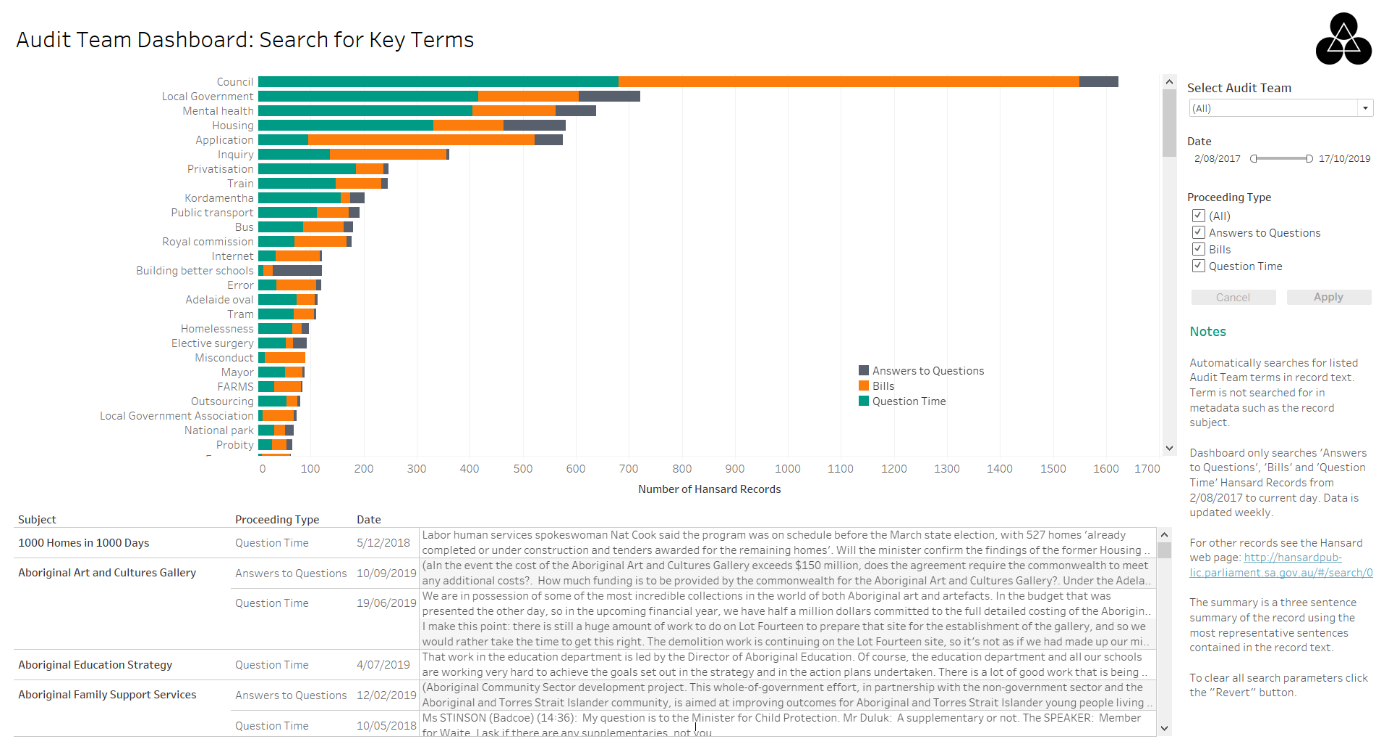
The Audit Team Dashboard (Figure 12) displays the number of records key terms have been mentioned for a given audit team. Audit teams will commonly search for specific terms which were provided in a spreadsheet. This spreadsheet also contained alternate spelling and plural forms for several terms. To 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.

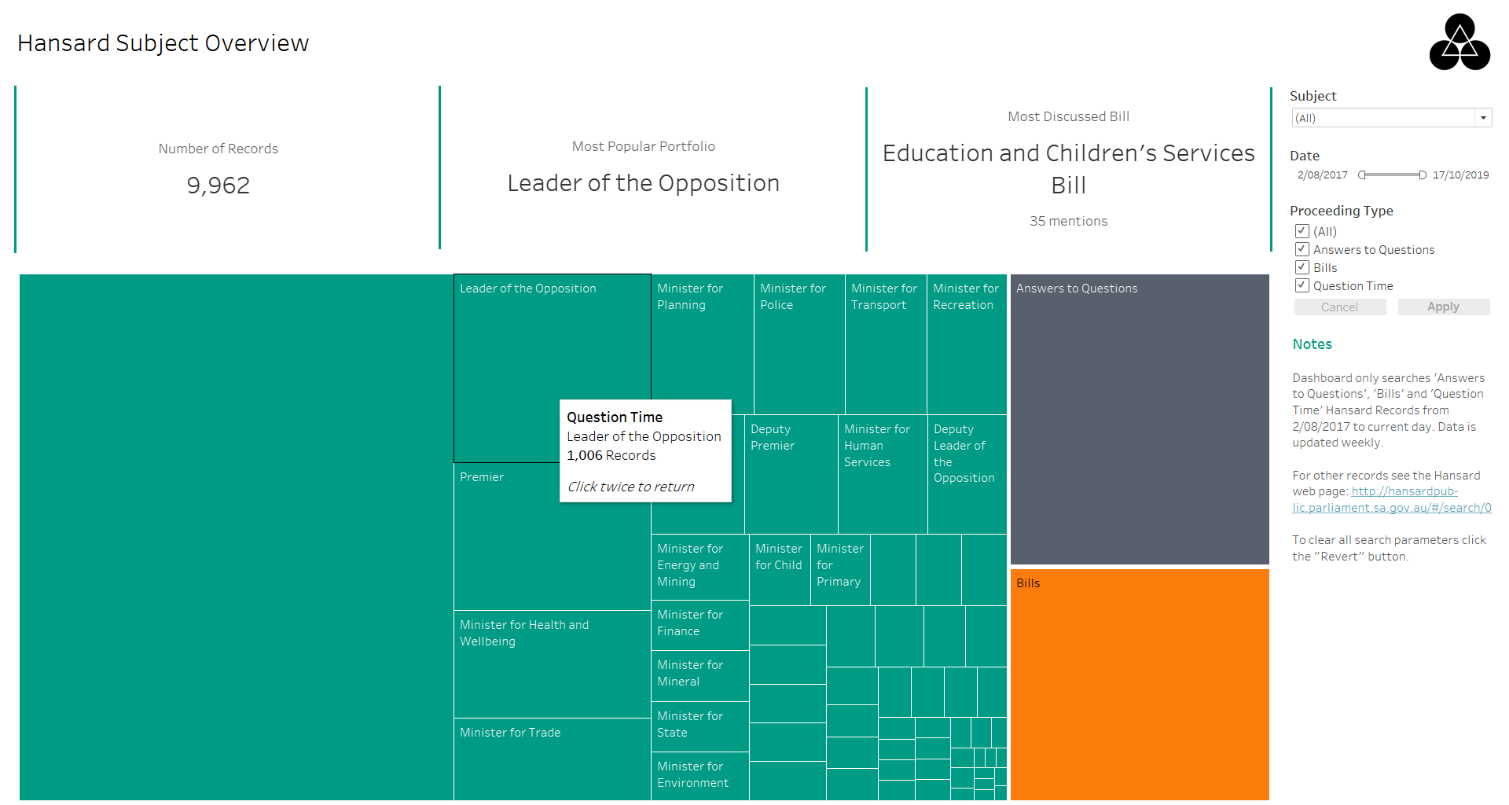


Figure 13: Subject Summary Dashboard drilled down to Portfolio Names

#### 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.

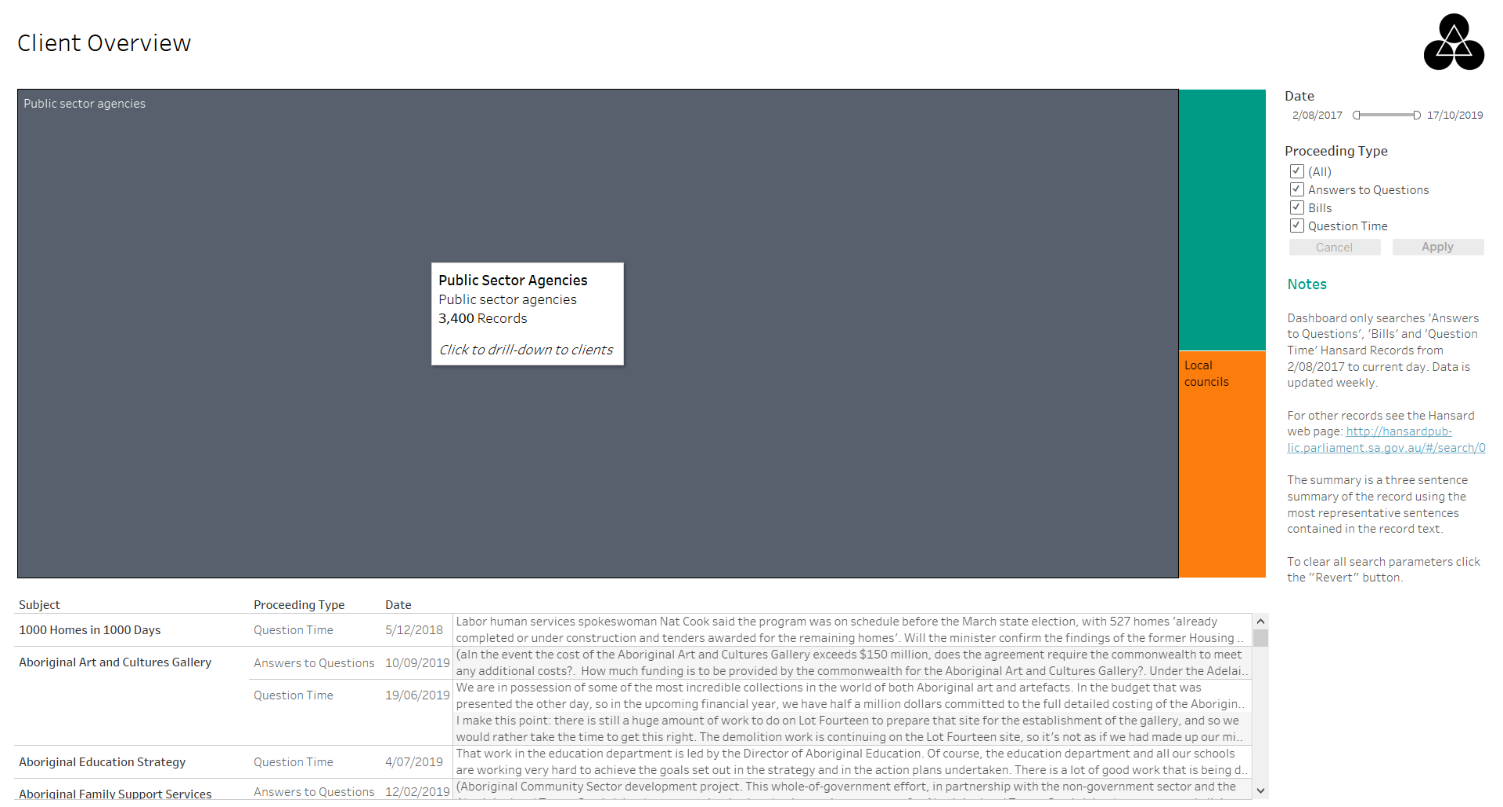


Figure 14: Client Summary Dashboard

#### 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 were 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) similar to 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 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 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 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.

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). In this visualisation the higher the frequency of usage for a term the larger/bolder it appears.

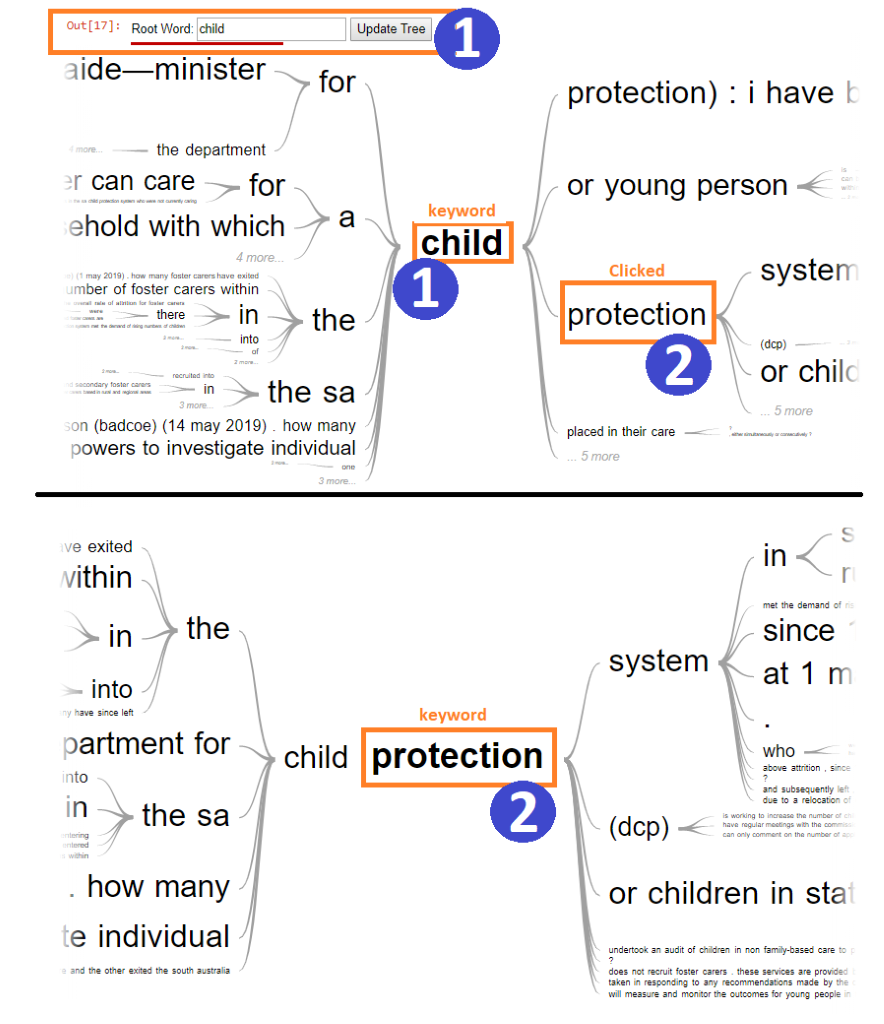


Figure 15: Word Tree

Word tokenization was an important transformation process within the project to establish efficient computation of algorithms. A bag-of-words was generated 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.

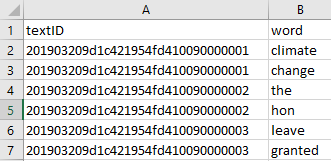


Figure 16: Words mapped to Hansard text

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

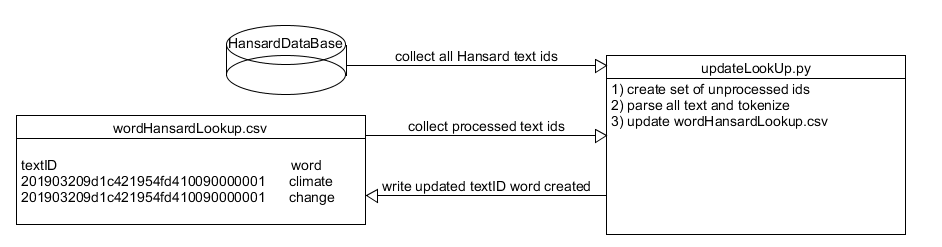


Figure : Word Tree Process

#### Text Summarisation

A requirement described by the PRG was to identify relevant Hansard records without having to read the entire record. To meet this requirement several text summarisation techniques were 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 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 for evaluation by PRG. 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 to rank web pages based on the structure of incoming links (NetworkX, 2014). It assumes that important pages are linked by other important pages. Instead of webpages, 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. An example output of this approach is shown in Appendix E.

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

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

Limiting the sentences analysed resulted in only 54.38% of the text being analysed to form a summary. This in combination with other pre-processing steps greatly reduced the algorithm running time. 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 PRG could evaluate the three different options. The summary approach preferred by the PRG 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

Topic modelling is used for automatically arranging, comprehending, and summarising big electronic archives. In other words, topic modelling provides an efficient way to analyse high volumes of text. Topic modelling works by identifying 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, Probabilistic Latent Semantic Analysis, and Correlated Topic Model. 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 thousands of Hansard discussions.

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) 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:” and “The SPEAKER: Order!”. After pre-processing, the text was put in a document term matrix (DTM) for further processing. This project used the “textmineR” library to perform topic modelling because it is a commonly used library (Jones and Doane, 2019).

One major problem that was encountered was selecting the appropriate number of topics. To address this problem the probabilistic coherence score was used. 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 were passed to the LDA algorithm and the coherence score of each topic calculated (Figure 18).

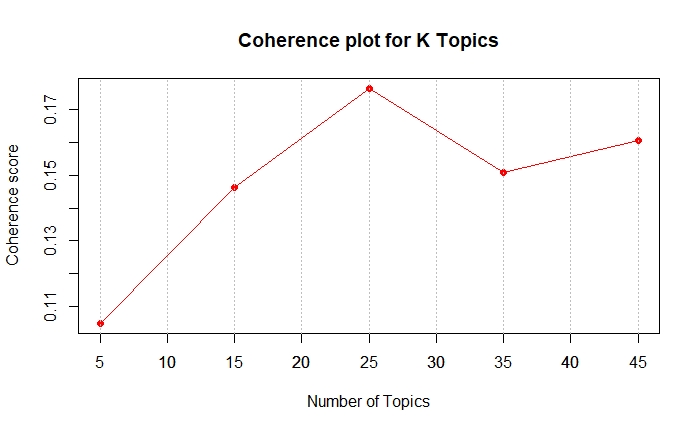


Figure 18: Coherence score for number of topics

For two years of “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 the topic explains more about the document. Appendix F shows the list of 25 topics that were generated and their prevalence score.

The “LDAvis” R library 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.

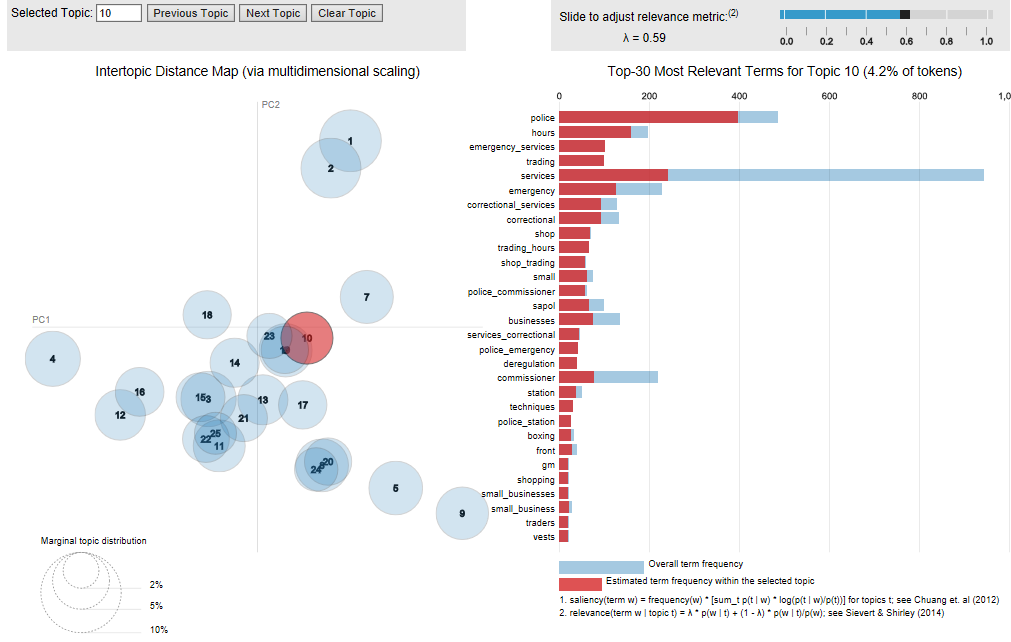


Figure 19: Topic model visualisation

Figure 19 shows 25 different topics plotted in a two-dimensional graph on the left side. The distance between the topics shows the similarity between topics. Short distances between the topics suggest that topics share similar terms and distant topics shares vague terms. The bar chart on the right side shows the frequency of words in the entire document (“sky blue” colour) and the frequency of words in individual topics (red colour).The relevance metric lambda (λ) is shown in the top-right corner and can be adjusted from 0 to 1. This 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 identify whether text is considered 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 PRG.

Sentiment analysis was coded using the R package “sentimentr” (Rinker, 2016). The most common approach is to use a simple 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 a 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 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, red is negative and white indicates neutral sentiment.

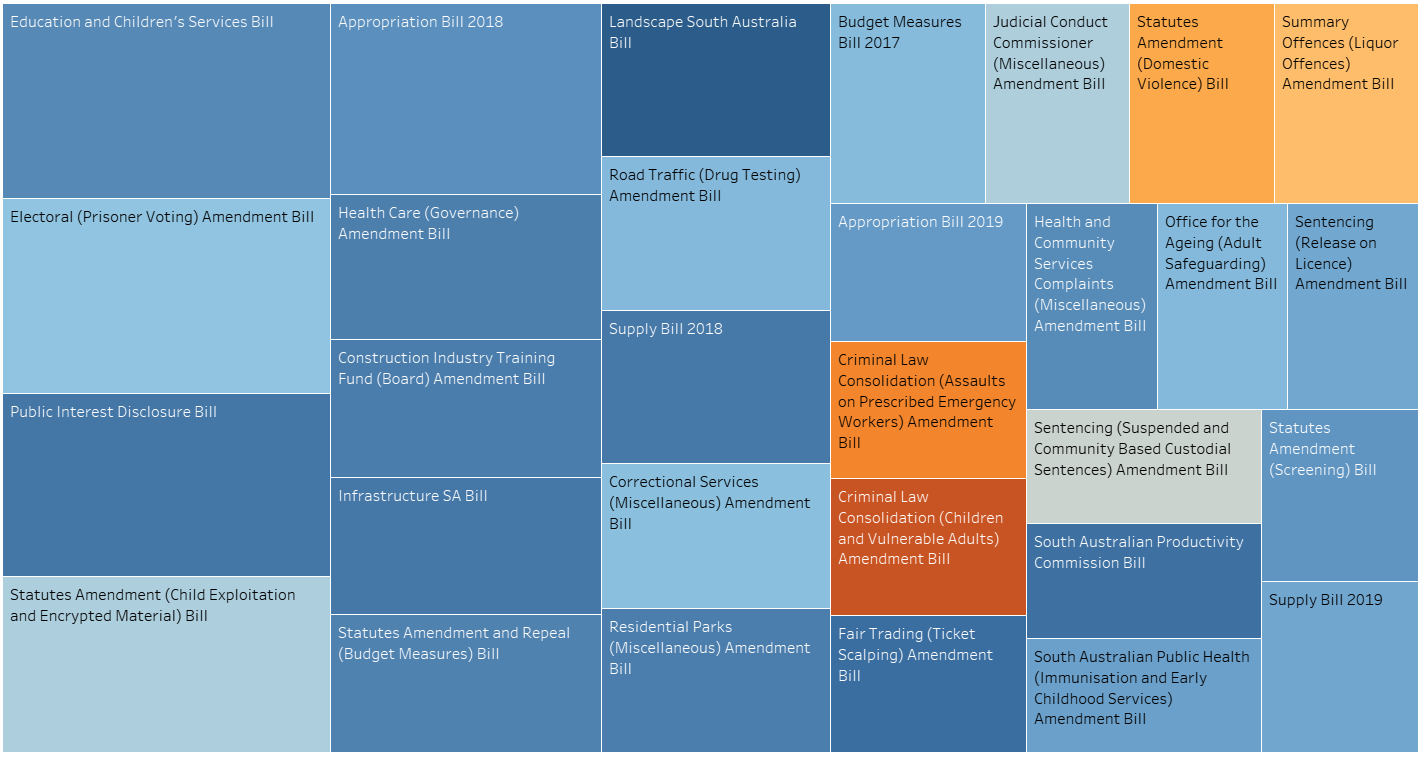


Figure 20: Average sentiment for each Bill heatmap

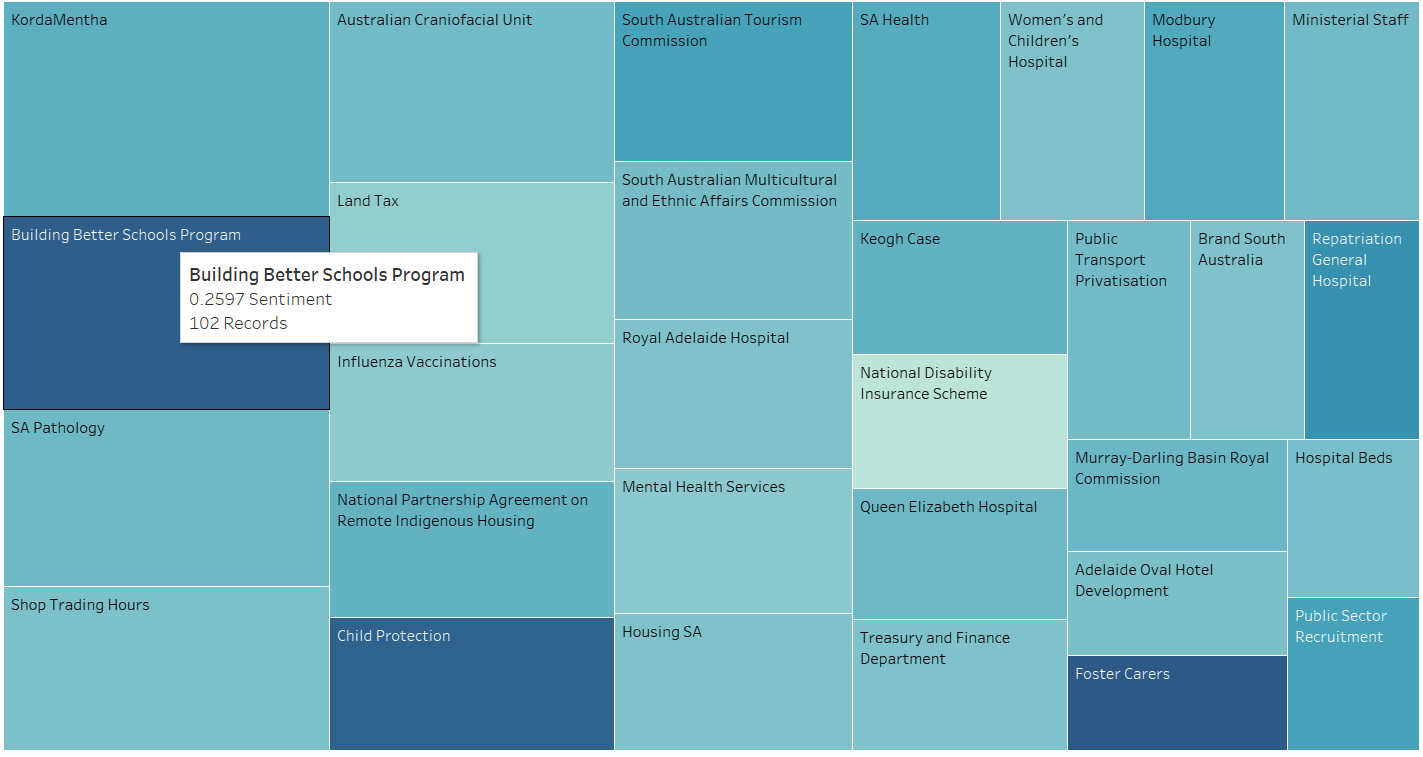


Figure 21: Average sentiment for each subject heatmap (with tooltip)

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

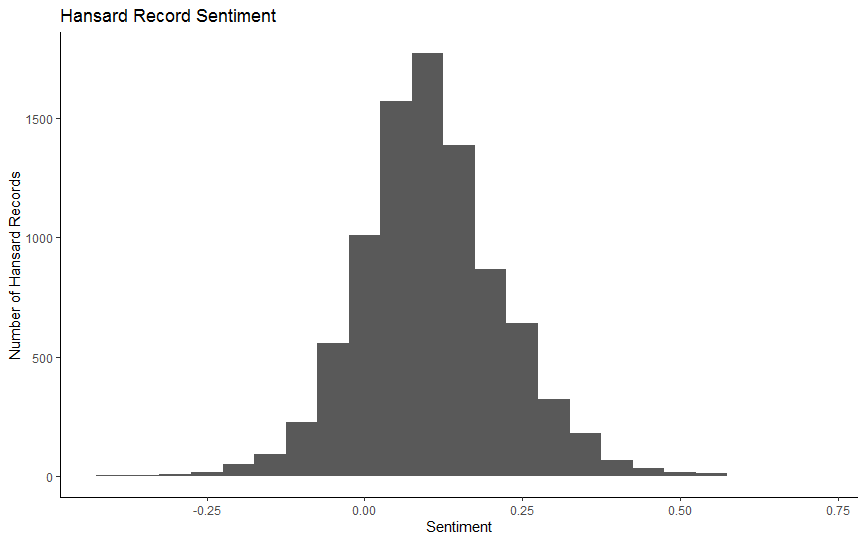


Figure 22: Sentiment of Hansard Records

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 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 2: Sentiment Score Statistics

# Project Performance Assessment

This section assesses how the project team performed against 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 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 AGD 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 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 27) allowing auditors to identify whether a subject was discussed over a long period of time or was discussed over 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 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 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 was 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 planned outputs 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 PRG 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 project captured by the project team are 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 can 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 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 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 for this project. It is recommended that improvements be made 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)
* Modify SSIS to drop staging tables at end of process
* 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 load new data into the database

Improvements could also be made to the text analytics. The techniques implemented should be evaluated against other implementations to ensure that it is the most appropriate for AGD. For example, sentiment analysis should be evaluated as the algorithm scores the word ‘government’ as negative sentiment. This may not be appropriate when applied to government records. Additional techniques could also be implemented such as document similarity. This technique 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 them into dashboards.

Additionally, the improvements listed below were requested by the PRG but 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.

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>.

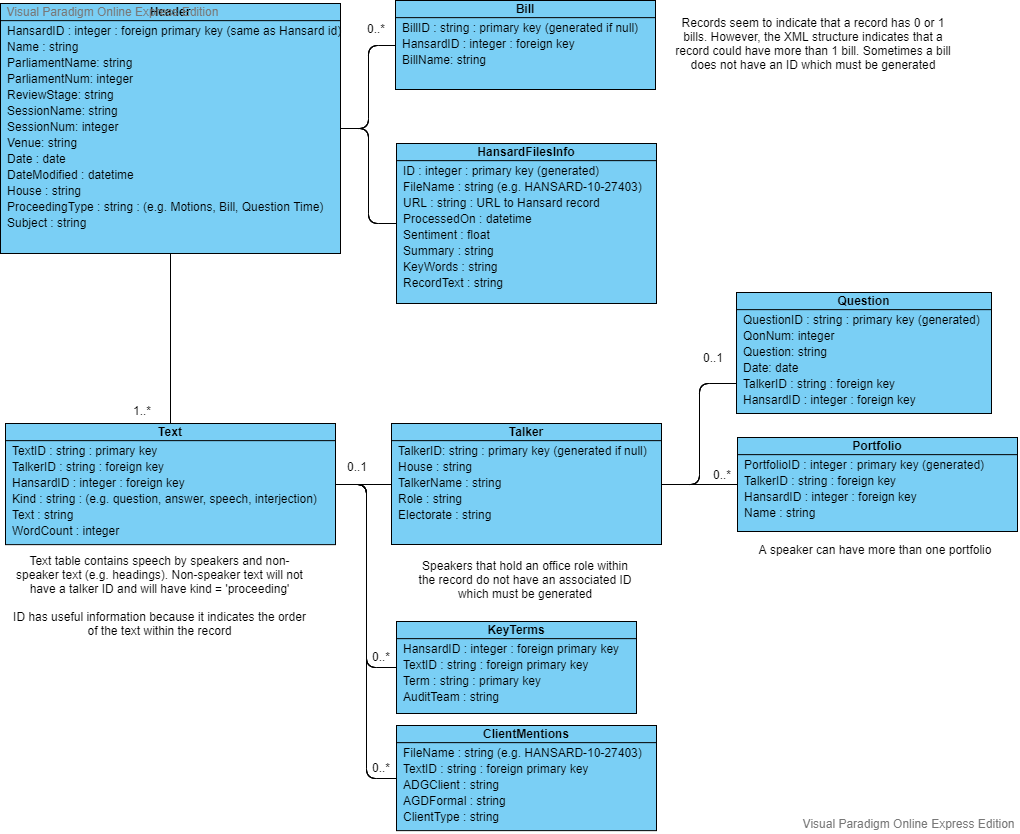
Sievert, C & Shirley, K 2014, Proceedings of the Workshop on Interactive Language Learning, Visualization and Interfaces, Baltimore, Maryland, USA, Association for Computational Linguistics.

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>.

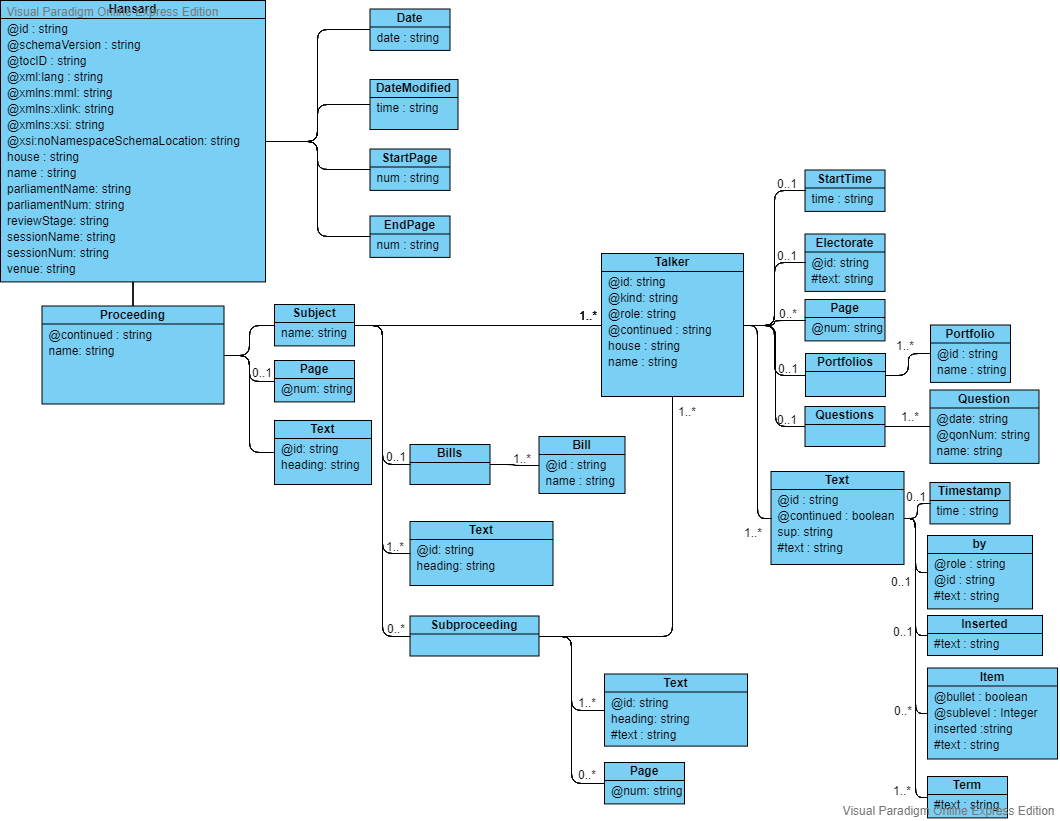
Tong, Z, & Zhang, H 2016, A Text Mining Research Based on LDA Topic Modelling, Sixth International Conference on Computer Science, Engineering and Information Technology.

# Appendix A: Database Schema



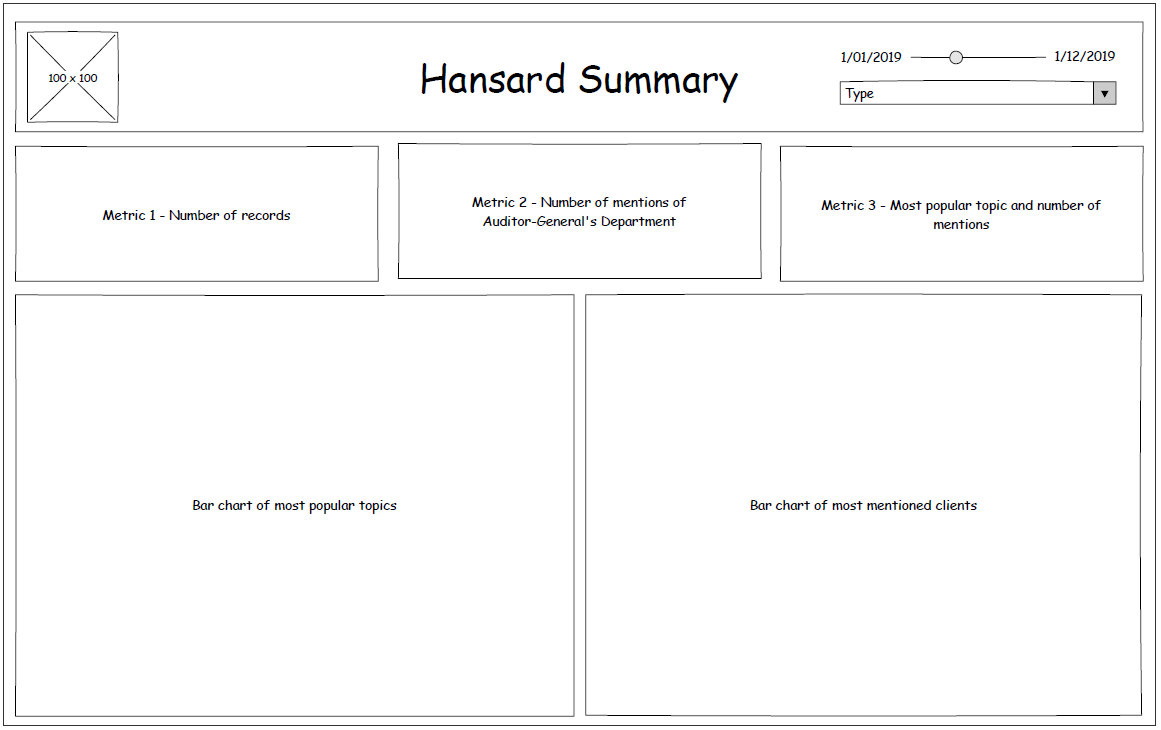
# Appendix B: Hansard XML Record Structure

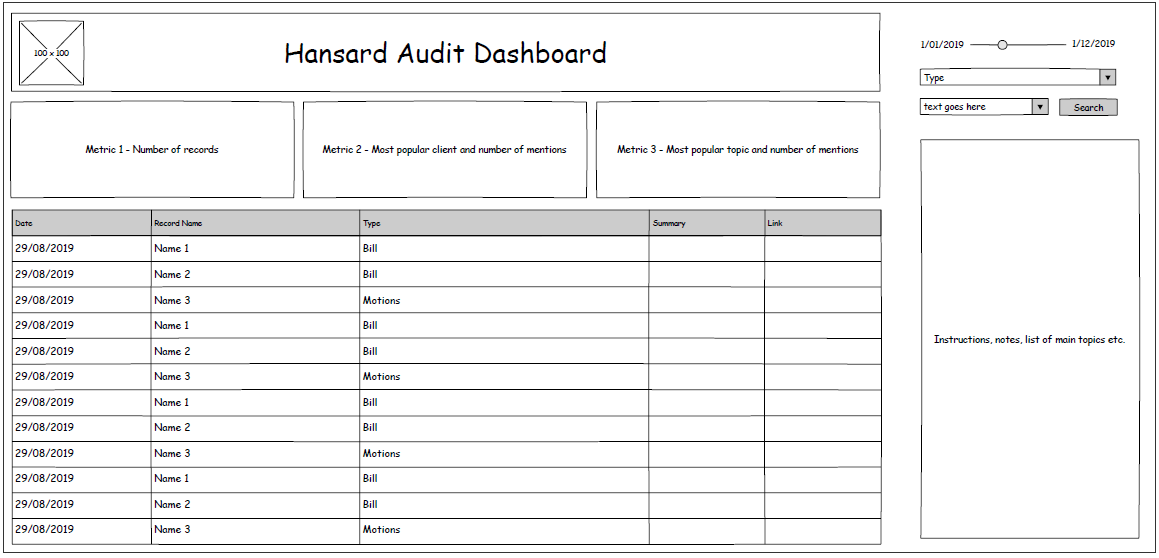
The UML diagram below shows the structure of the XML records that were downloaded from the Hansard webpage. This diagram was created to assist with understanding Hansard records.



# Appendix C: 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.





# Appendix D: Dashboard Charts

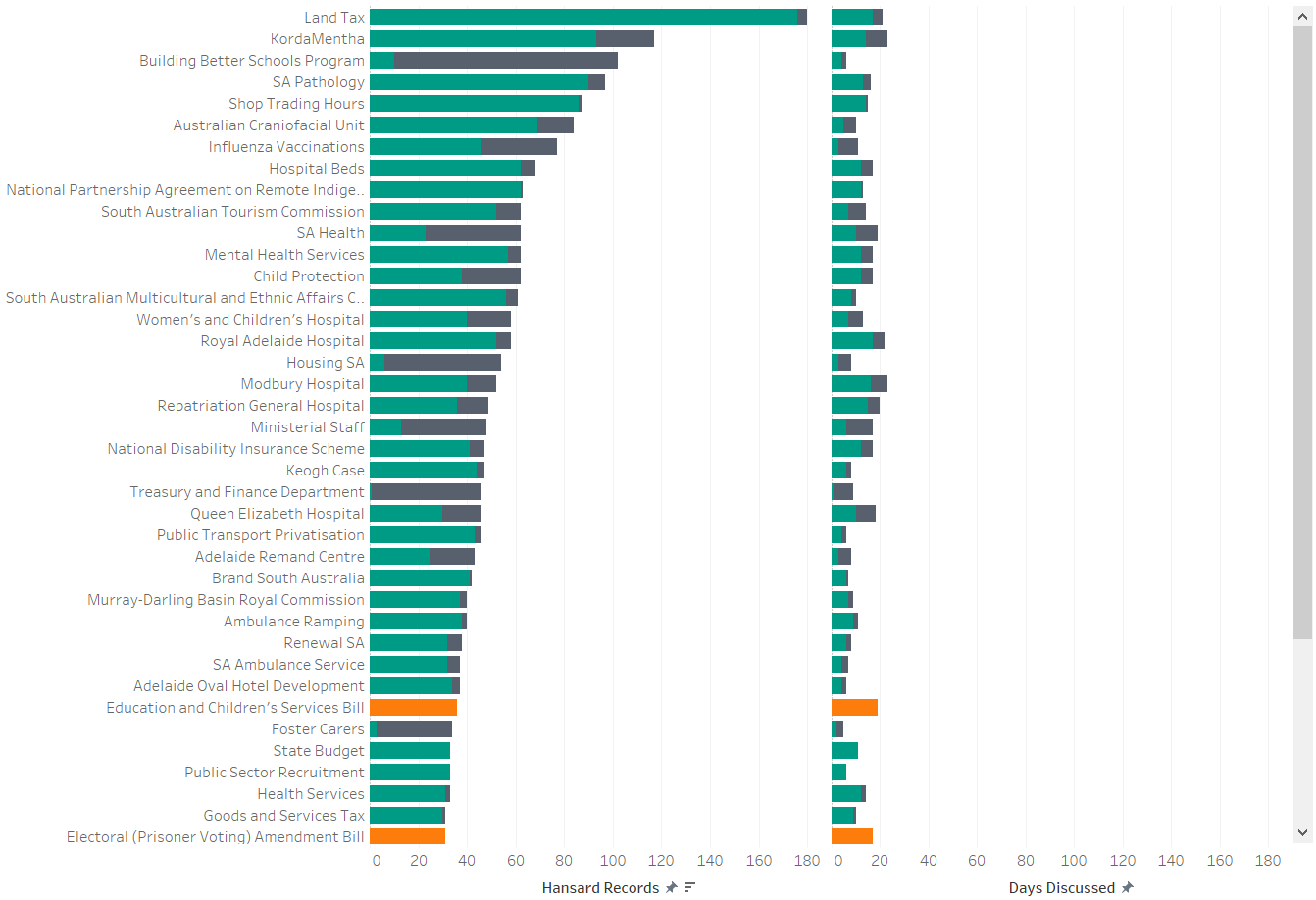


Figure 23: Top 50 Clients in Summary Dashboard

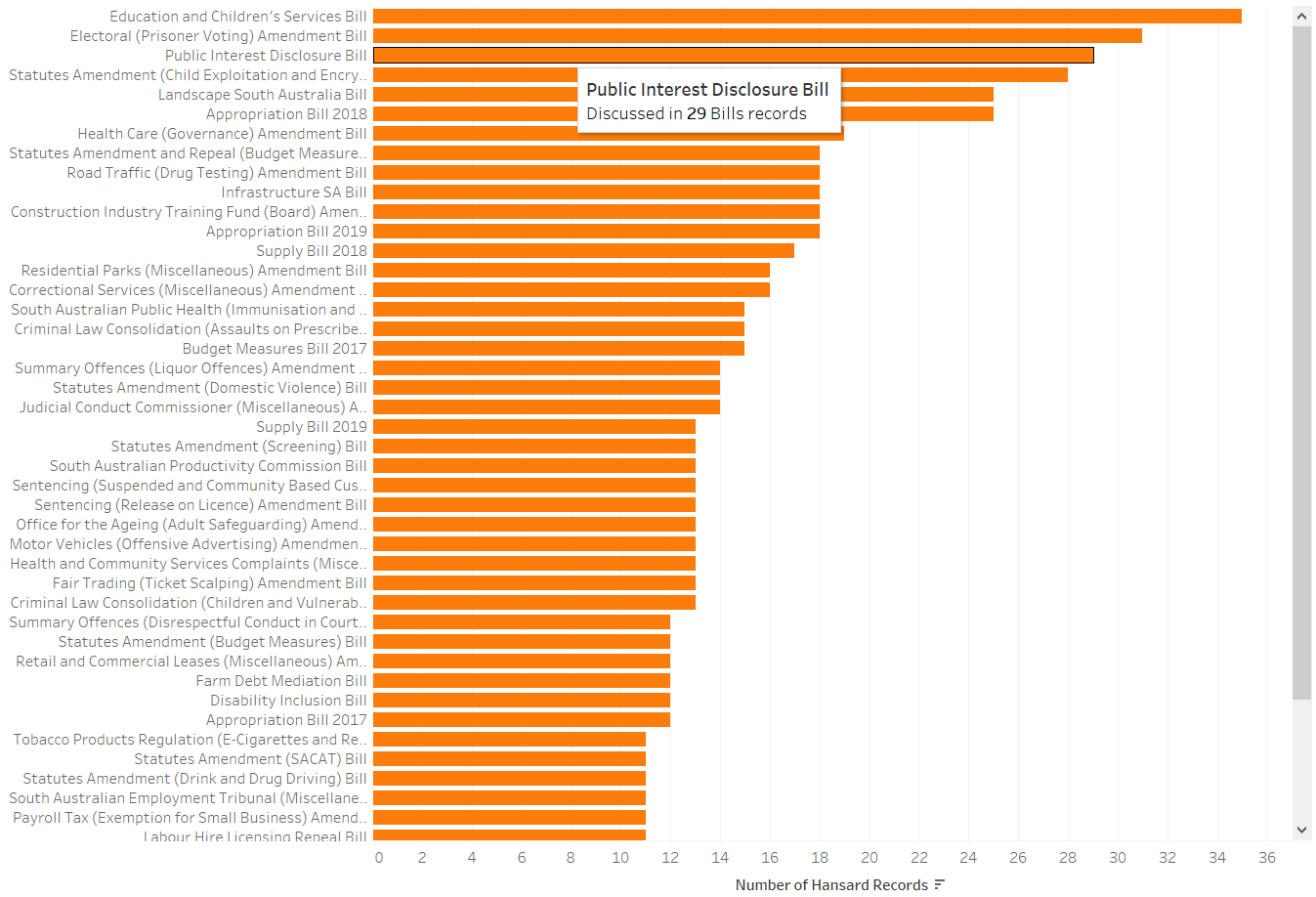


Figure 24: Top 50 Bills in Summary Dashboard

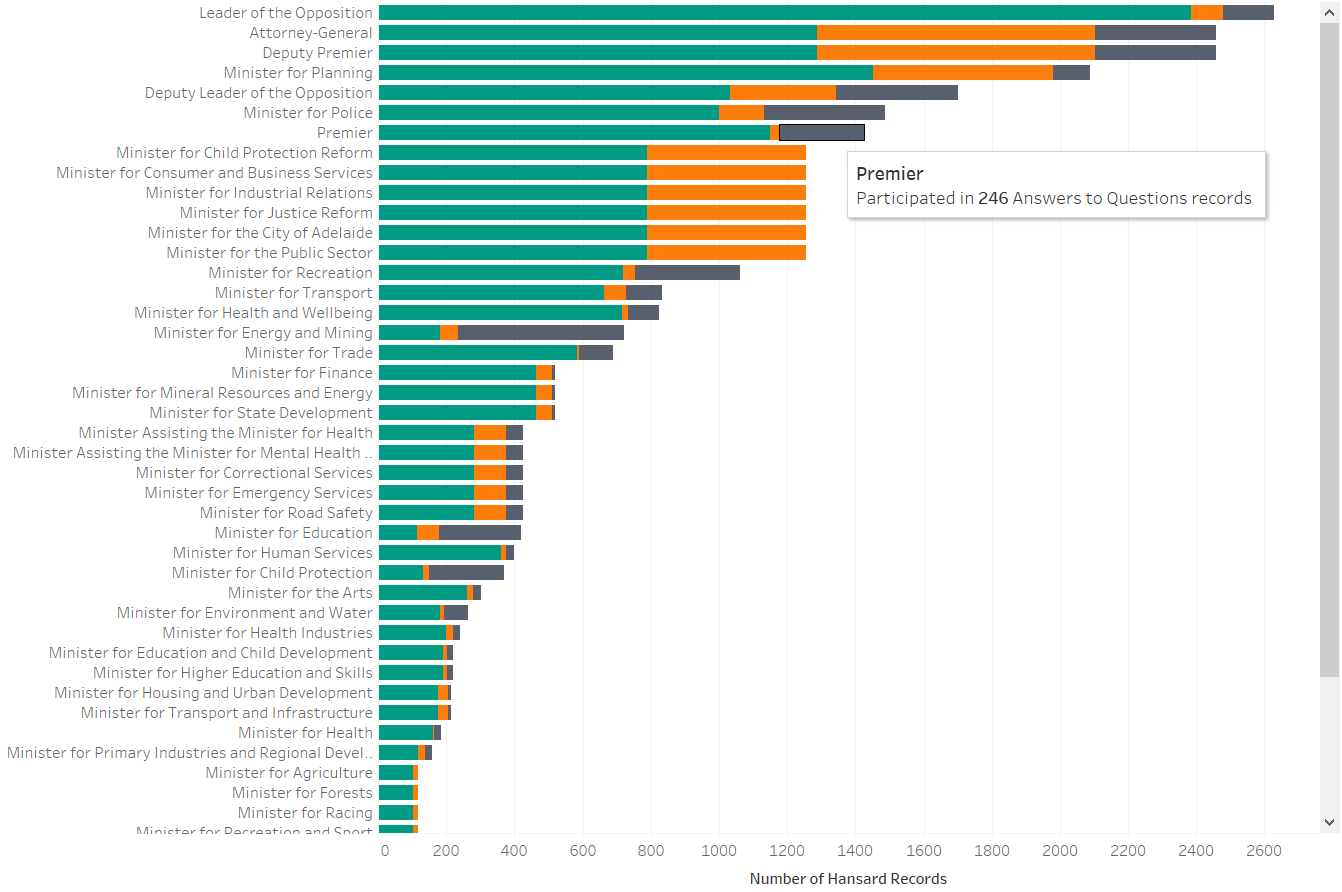


Figure 25: Top 50 Portfolios in Summary Dashboard

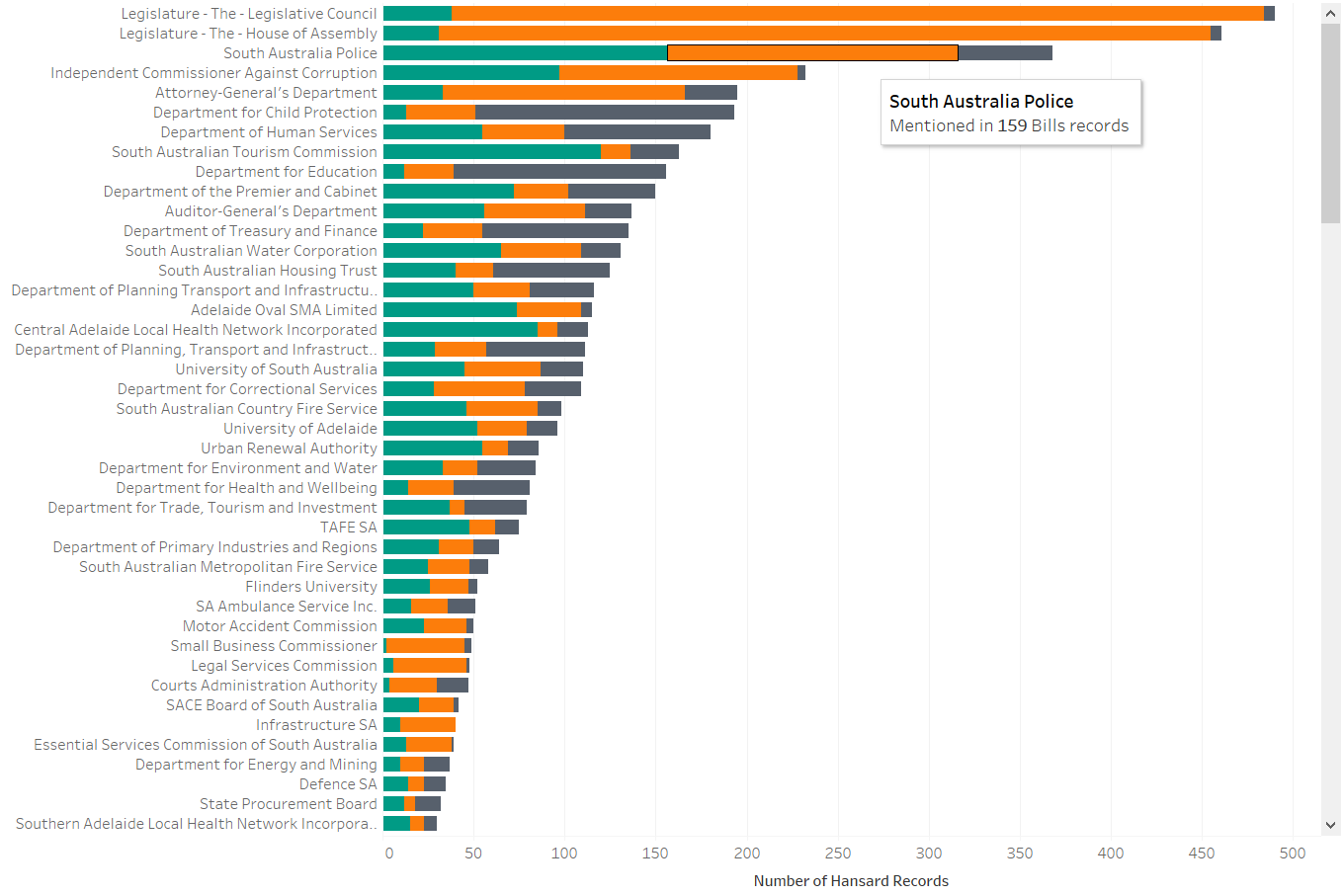


Figure 26: Top Clients in Summary Dashboard

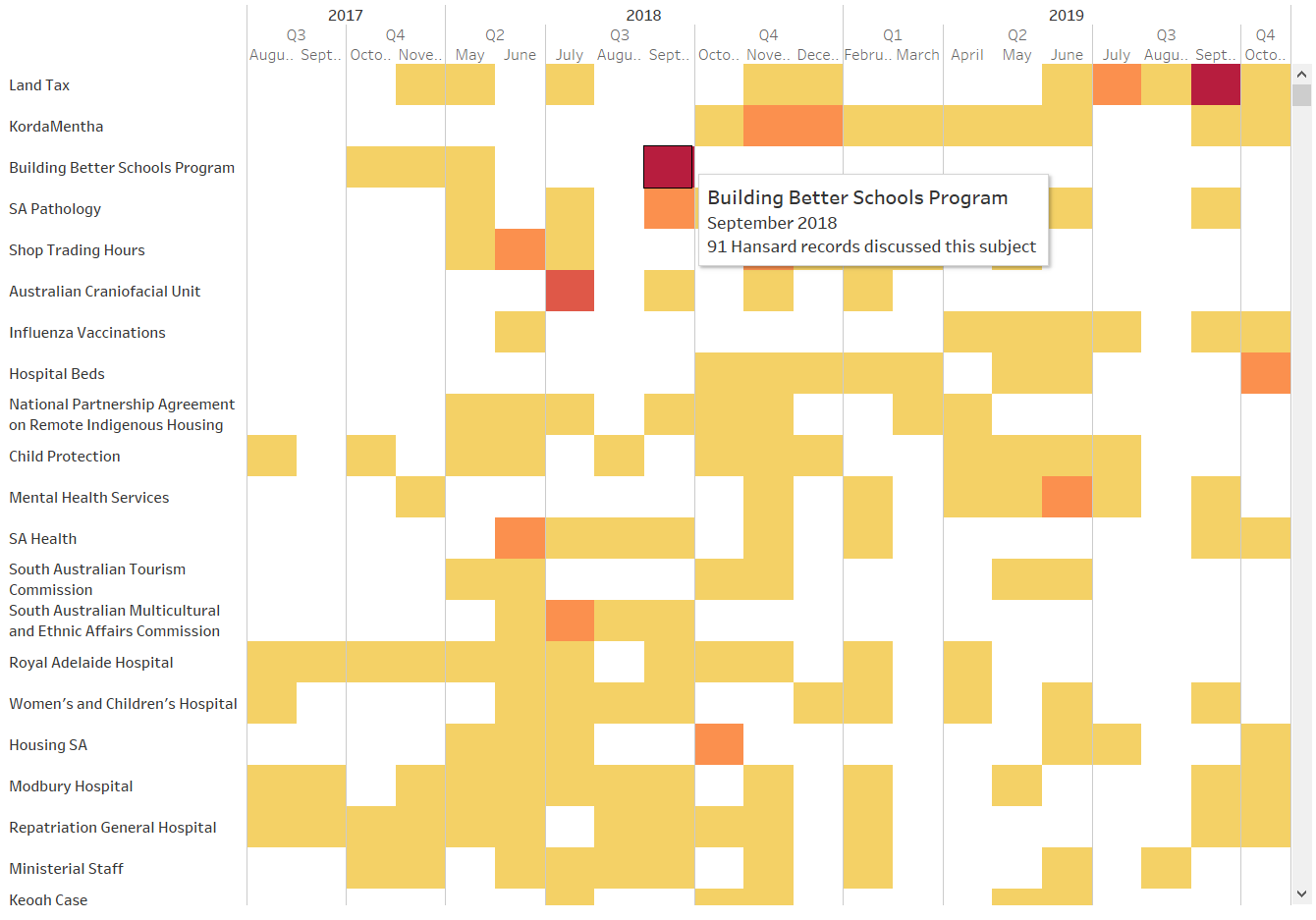


Figure 27: Subjects Discussed Over Time in Summary Dashboard

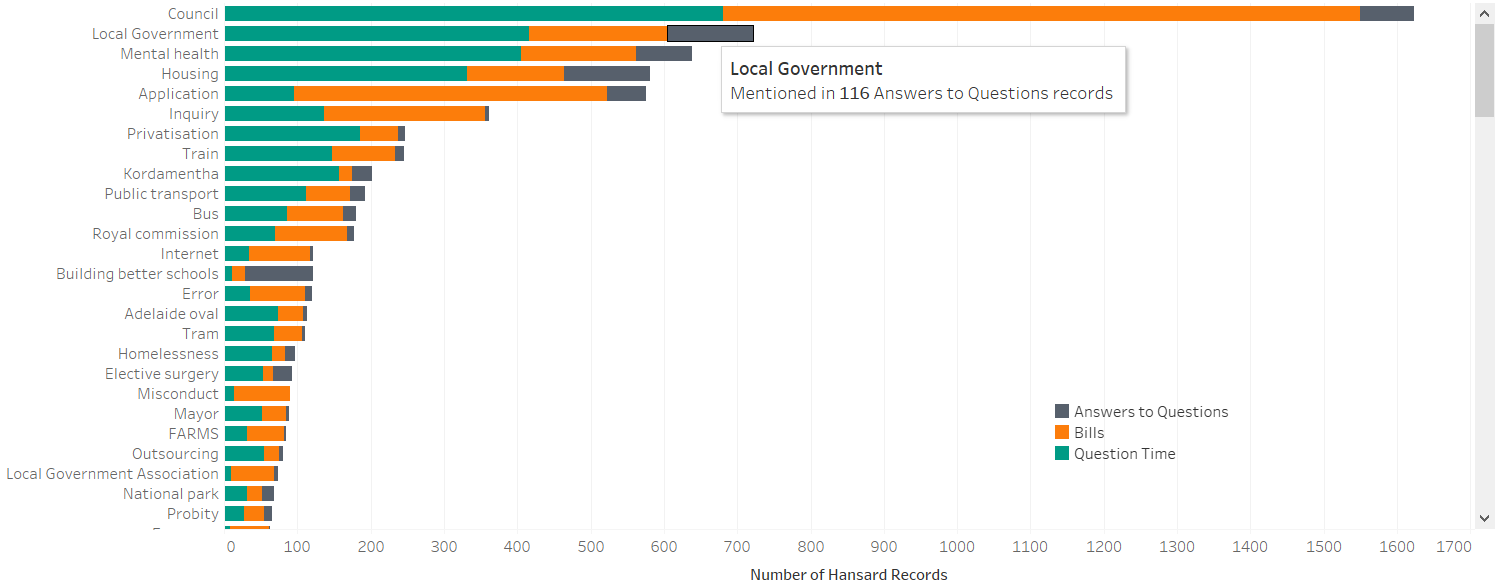


Figure 28: Key Term Search in Audit Team Dashboard

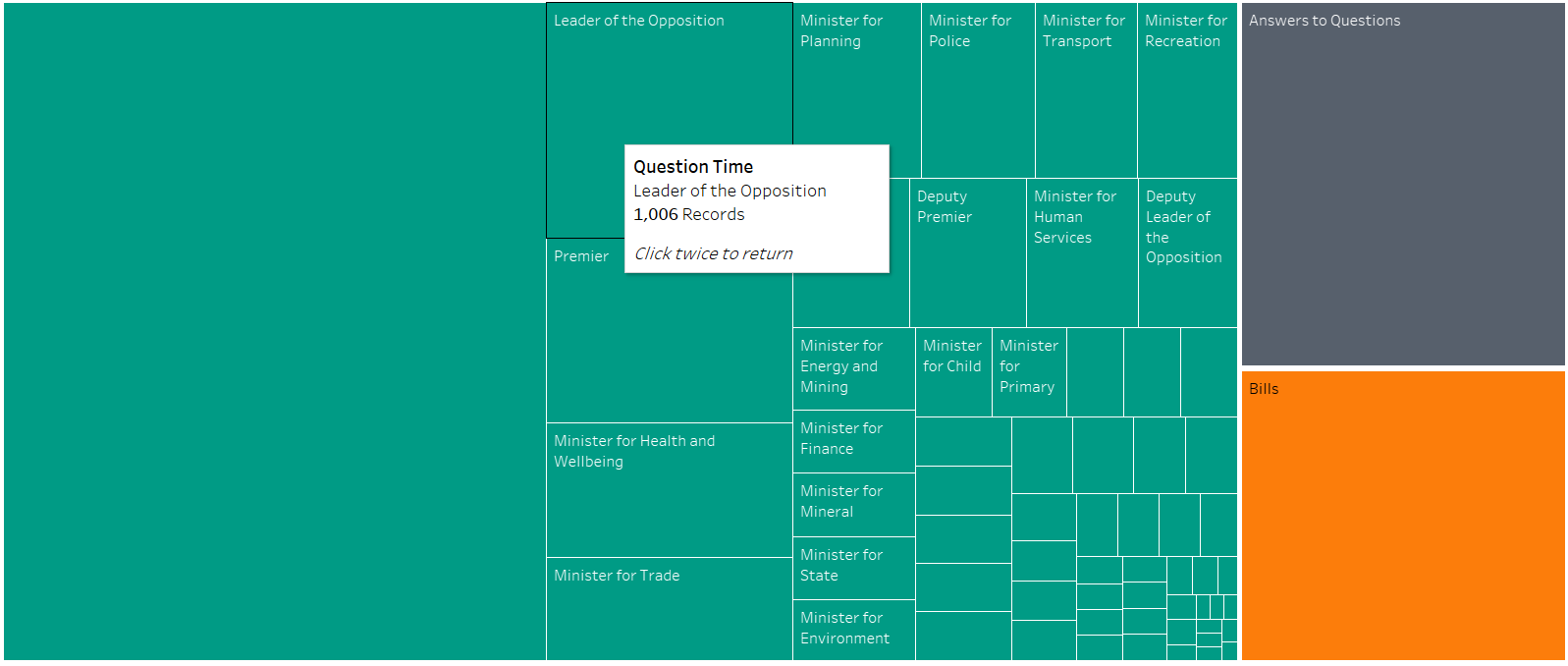


Figure 29: Subject treemap in Subject Overview Dashboard

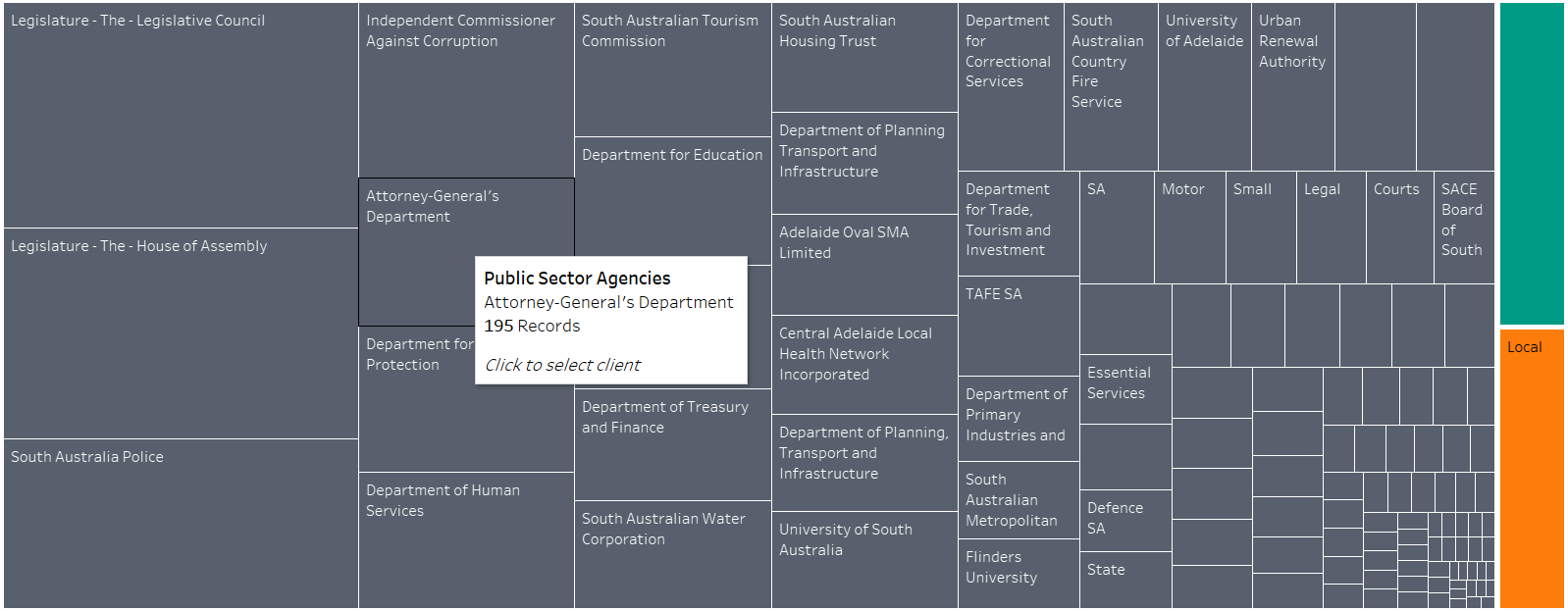
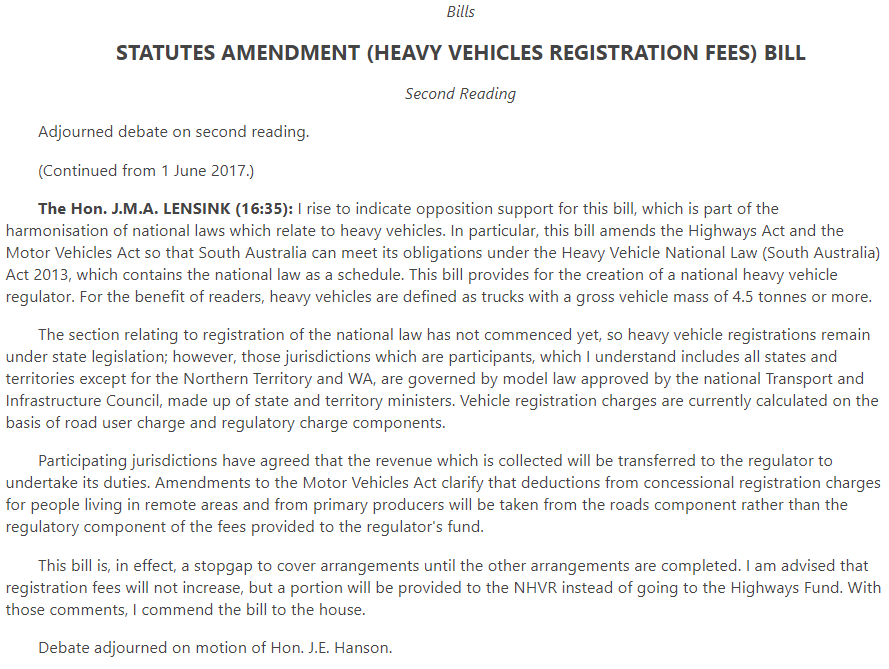


Figure 30: Client treemap in Client Overview Dashboard

# Appendix E: Document Summary

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

****

**Three-sentence summary:**

'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 F: List of Topics

|  |  |  |
| --- | --- | --- |
| Topic | Prevalence | Top five terms |
| 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 G: Gantt Chart

