SRAuditor: An Automated Assessment Tool for Statement of Advice Documents

Yong-Bin Kang[a], Abdur Rahim Mohammad Forkan[b], Prem Prakash Jayaraman[b], Hung Du[b], Rohit Kaul[b], Dan Hunter[c,d]
[a]Department of Media and Communication, Swinburne University of Technology, Australia
[b]School of Science, Computing and Engineering Technologies, Swinburne University of Technology, Australia
[c]Faculty of Business & Law, Queensland University of Technology, Australia, [d]Fourth Line Pty Limited, Australia
{ykang, fforkan, pjayaraman, hungdu, rkaul}@swin.edu.au, dan.hunter@qut.edu.au

Abstract

Financial advice is given by a registered financial adviser (RFA) in the form of a statement of advice (SoA) document. To limit liability, financial advisor groups periodically assess SoA documents for compliance with legal regulations, internal policies, and best practices. However, this is a manual process that is often subjective, time-consuming and tedious. In this paper, we propose, implement and evaluate SoA Risk Auditor (SRAuditor), a natural language processing (NLP) framework to automatically assess and audit SoA documents. SRAuditor consists of two major components. The first one is a SoA transformer (SoA-T), a tool that automatically transforms and maps SoA document (generally a PDF). The other one is a question-answering engine (QA-R) that recommends legally compliant answers based on rule-based approaches for given SoA audit questions to assess and audit SoA documents. We validate the accuracy of SRAuditor's ability by evaluating it against assessments conducted by domain experts (i.e., financial advisors, lawyers). Experimental results using real-world SoA documents provided by our industry partner, Fourth Line Pty Limited indicates that SRAuditor has a high potential to be used for automatically assessing and auditing SoA documents.

1. Introduction

Financial advice plays a vital role in defining and accomplishing the financial goals of a client based on his/her financial situation and needs. It provides the client with personalised insight and ongoing support for taking control of their wealth for the future. In Australia, as in many other countries, financial advice is given by a *Registered Financial Adviser* (RFA) who is required to complete specific training and be registered with a *regulatory body* in order to provide advice. Recently, demands for seeking financial advice have attained significant attraction in many countries [1]. The 2020 Financial Advice Report says that 2.6 million non-advised Australians intend to seek advice from RFAs in the next two years. This number is significantly higher than the numbers who sought advice in 2019 (2.1 million), and double the levels observed in 2015 (1.3 million) [1].

Central to the advice given by a RFA is a *Statement of Advice* (SoA) document that sets out the advice given to a client. It includes key guidelines on which the advice is given on investing, insurance, superannuation (retirement) and estate planning, and also provides information and plan on benefits the client will receive. A SoA document ensures that a RFA provides the client with high-quality advice and contains practical information helping the client to make informed decisions. It is also a highly regulated document and often provides the evidential basis for actions against financial advisors by clients in the event that they claim that they have been given improper or poor advice. A 2018 report by the Australian Securities and Investments Commission (ASIC) estimated that 75% of advice reviewed were non-compliant with the law [2].

A *financial advice group*, which may employ many RFAs, periodically undertakes audits to assess their SoAs for compliance with legal regulations, internal policies, and best practices to limit their liability. Undertaking this audit is currently a manual process, is time-consuming, difficult, and expensive. In general, SoA documents are very complex including multiple interconnected recommendations for clients. Further, most financial advice groups have their own semi-standardized SoA templates, thus making hard to generate consistent recommendations across financial advice groups. As a result, most compliance approaches involve a manual audit of a small sample (typically less than 5%) of SoA documents generated within a financial advice group.

To address these problems, we propose and develop *SoA Risk Auditor* (*SRAuditor*) for automatically assessing the quality of SoA documents to support auditing and reporting. The SoA quality is determined through a number of pre-defined regulation and compliance questions drawn from legal and auditing requirements.

We call these questions *SoA audit questions* (SAQs). SAQs vary in difficulty: easy questions are mainly supported by an 'yes/no' response, while assessing complex questions requires contextual analysis of a SoA document. Little work has been conducted on applying NLP techniques for automatically assessing and auditing SoA documents. One recent study [3] used machine learning with focus addressing one SAQ i.e., whether the client's goals have been addressed by appropriate recommendations. Differing from it, our approach is more comprehensive and takes into account diverse SAQs which have different levels of complexity. SRAuditor is designed to contribute to market integrity and client confidence and also reducing the time needed to assess SoA documents.

This paper makes the following contributions:

- SRAuditor, a NLP-based framework to automatically assess and audit SoA document that incorporates:
  - SoA-T, a tool to transforms SoA documents into a machine-interpretable format.
  - QA-R, a tool that recommends legally compliant answers based on rule-based approaches for a given SAQs to assess and audit the SoA document
- Experimental evaluations using real-world datasets to assess the accuracy of SRAuditor.

The rest of the paper is organised as follows. Section 2 presents related works. Section 3 discusses the architecture of SRAuditor and how it works in detail. Section 4 presents the details of our evaluation, and Section 5 concludes this paper.

2. Background and Related Work

SoA documents are typically unstructured and text-heavy [4] containing headings, subheading, paragraphs and tables. These documents are used by RFAs. Usually, RFAs may have their own preferred vendors [5, 6]. Accordingly, they are likely to use a template of the chosen vendor to create SoA documents. Spruit et al. stated that The format diversity in different SoA templates makes SoA documents very difficult to automatically analyse [7]. For example, in a template, client goals can be found under the 'Goals and Objectives' heading [6]. In another template, however, it is identified under the 'Your Objectives' subheading under the 'Executive Summary' heading [5]. That is, similar information is highly likely to be located under different headings and subheadings in different templates. As another example, in some templates, required information (e.g., client's income) needs to be extracted from a paragraph in the body text, while, in some other templates, the same information is found inside a table. Therefore, although a regulatory body can provide guidelines about some general structure of SoA, according to Breymann et al. there is no standard governing the finer level structure of SoA [8]. The practice of using domain-specific vocabularies by RFAs further complicates the use of a fully automated process for information extraction from SoA documents. For example, one type of retirement strategy can be written variously as 'self-managed super fund', 'SMSF', or 'SM superannuation fund'.

In the digital era, these sorts of regulatory documents are growing rapidly, as is the demand for automated processing mechanisms. NLP and AI algorithms have been extensively used in for document analysis in finance, audit and accounting as per review of Nassirtoussi et al. [9], and in application areas such as markets' predictability [10], risk assessment [11] and compliance checking [12]. However, automatic assessment of regulatory documents is a relatively new field in financial document analysis and little work has been reported [3, 7]. However, the survey [13] summarised the importance of NLP and AI for accounting, auditing and finance sector for future research. The work [13] also highlighted automated assessments of financial documents will be a part of future research based on the fact that many companies have failed to conduct effective risk assessments.

The research [14] proposed a semi-automated approach for extracting key elements from legal documents. In the research, a rule-based technique was developed to recognise part of sentences and entity linking based on the knowledge input of legal experts. The study [3] described a framework to validate the 'goal-advice appropriateness' requirements in SoA documents. The authors used 194 SoA documents with different SoA templates and generated a gold-standard dataset manually labelled by regulatory compliance officers with 1139 sentences as 'goal' and 903 sentences 'advice'. The sentences are mapped to 17 domains (e.g., insurance, superannuation). This work focused on automatic assessment of a single SAQ (i.e., whether clients goals are properly advised) whereas in SRAuditor we considered assessing eight SAQs with three different complexity levels. For generating the gold-standard data, we followed similar mechanisms (i.e., manual assessment by RFAs) and evaluated the quality of the answers for the eight SAQs using accuracy as suggested in [3]. Unlike [3], we do not need any training data, as we formulate our problem as an unsupervised approach.

In summary, inspired by these literature surveys and

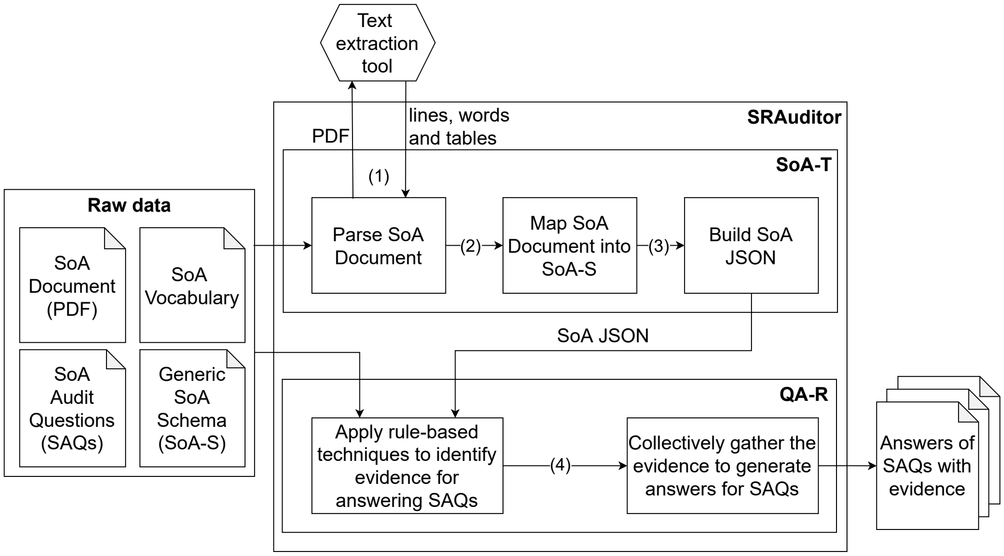

Text extraction tool
PDF
lines, words and tables
SRAuditor
SoA-T
(1)
Raw data
SoA Document (PDF)
SoA Vocabulary
SoA Audit Questions (SAQs)
Generic SoA Schema (SoA-S)
Parse SoA Document
(2)
Map SoA Document into SoA-S
(3)
Build SoA JSON
SoA JSON
QA-R
Apply rule-based techniques to identify evidence for answering SAQs
(4)
Collectively gather the evidence to generate answers for SAQs
Answers of SAQs with evidence

Figure 1: The overall architecture of SRAuditor

we make contribution by SRAuditor to support automated assessment and auditing of financial documents.

3. SoA Risk Auditor: SRAuditor

The overall architecture of SRAuditor is presented in Figure 1. There are two major components in SRAuditor: SoA-T and QA-R. SoA-T transforms a SoA document into a JSON object (called SoA JSON), which is used in QA-R for answering a given SAQ. QA-R recommends the right answers for given SAQs with relevant evidence that indicates reasons on how the answers have been induced. In the following, we present their detailed descriptions.

3.1. SoA Transformer: SoA-T

SoA-T performs three steps as presented in Figure 1. First, given a SoA document, it uses a text extraction tool (e.g., AWS Textract, IBM Datacap, Microsoft Azure Text Analytics) to extract information about words, lines and tables in the document. Note that we do not depend on a particular extraction tool.

Second, given a SoA generic schema, namely SoA-S, SoA-T further extracts the information of the headings and subheadings and their associations from the information extracted in the first step and maps them to the SoA-S structure. SoA-S proposed in this paper has been co-developed in collaboration with financial advisor group (i.e., Fourth Line Pty Limited[1]) (as shown in Table 1) who are domain experts in finance. SoA-S defines general headings, subheadings, and their associations across different SoA templates. An important benefit of defining SoA-S is that it can help us to better capture the key headings, subheadings, and their associations that can be commonly observed in various SoA document templates. Another benefit is that by mapping various SoA templates to SoA-S, automatic answering for SAQs can be more formally formulated and simplified. Also, we assume that a SoA vocabulary is predefined and given to SoA-T, where it defines key terms with their synonyms that are necessary for QA-R to recommend the right answers.

Third, SoA-T constructs the SoA JSON, mapped using SoA-S, which has the headings, subheadings, tables, pages numbers (i.e.,the start and end page numbers of each heading and subheading), paragraphs and their associations that can be observed in a given SoA document. This JSON contains a precise, compact and useful information for QA-R to make answer recommendation for SAQs.

In the following, we describe the above three steps in more detail.

Parse SoA Document: In this step, our goal is to extract the information about words, lines and tables from a SoA document. This information will be essentially used for QA-R to identify right answers for SAQs. There are credible machine tools are available that automatically understand and extract text, forms and tables from a text document. In our case, we use Amazon Web Service (AWS) Textract [15].

More specifically, given a SoA document, we focus on identifying the following types of useful information:

[1] https://fourth-line.com.au/

No	Heading	Subheading	Delivered information by Heading
1	Intro	Cover Letter, Index	Contains cover letter of the document and table of content.
2	Executive Summary	Goals, Scope, Summary of advice, Outcomes, Personal summary	Contains the list of the client(s) financial goals on which they would like to seek advice.
3	Personal Details	Personal details, Lifestyle assets, Managed, Superannuation, Liabilities, Net worth, Insurance, Cashflow, Estate planning	Contains client(s) information regarding the current financial position.
4	Strategy	Strategy, Basis, Risks, Consequences, Alternatives, Product details, Product feature and benefits, Product risks, Product replacement	Contains the advice on the required strategies for client(s), if requested. This describes the benefits, product cost, product fees of recommended superannuation and also suggests the alternative strategies.
5	Risk	Risk profile, Asset allocation, Need analysis	Contains evaluation of clients(s) ability to risk for investments.
6	Disclosure	Fees, Ongoing services, Disclosures, Authority to proceed	Contains the cost of the services and advice.
7	Info	Generic, Product profiles, PDS, Research, Projections	Contains the list of benefits, cost and projections of the advice given by the advisor
8	Annexures	Annexures	Contains client(s) information regarding the current financial position, superannuation, insurance, and managed funds in a tabular format.

Table 1: The proposed SoA generic schema (SoA-S)

First, we extract line information that consists of text in all lines. Second, we also attempt to identify word information that contains all words, their positions and their heights in each line. Especially, the height of each word will be useful for identifying whether it is a candidate for heading or subheading, assuming that a heading has the highest height and a subheading has the second highest height in the document. We have found that this assumption is true across various SoA documents in general. Third, we also identify table information that shows all tables with their cell information that appear in the document. Further, we aim to extract page number information that all words, lines and tables appear in the document.

Map SoA Document into SoA-S: Given the information extracted in the previous step, in this step, our focus is to recognise headings and subheadings, and their associations based on page number information. Note that the information of the headings and subheadings are defined in SoA-S seen in Table 1. To achieve this, we explore information about the page number and the height value of each heading or subheading.

More specifically, a challenge here is how to distinguish words (or phrases) indicating heading, subheading, and body text indicating the main content under a heading or subheading. To address this, given SoA-S, we first identify candidates of headings and subheadings from the extracted lines, Second, we recognise what are the headings and subheadings from the candidates based on their heights. We observe that the heights of headings are higher than those of subheadings, and the heights of subheadings are higher than those of words or phrases in body text.

Our next step is to associate each heading with its relevant subheading(s). To do this, our key idea is to leverage the page number and the line information of each heading and subheading. For each subheading that appears in previous lines, we consider the closest heading, and associate the heading with the subheading together. Here, the term ‘closest’ means the minimum gap between the line numbers of the heading candidates and the subheading.

Build SoA JSON: In this step, our goal is to recognise paragraphs in the body text from a given SoA document, as well as associate the tables extracted from the SoA document, and paragraphs with the headings and subheadings identified while mapping the SoA document into SoA-S. Given a SoA document, we identify paragraphs using the heights of words (or phrases) based on information about lines that do not contain a heading or a subheading. Then, we associate the paragraphs and the tables with the corresponding heading and subheading that must contain these paragraphs and tables. The associated information will be presented in a SON JSON organised in a structured way so that it can be easily read and effectively handled by machines.

An example of a SoA JSON is presented in Figure 2. We see a heading, a subheading, a paragraph (tagged by ‘content’), two tables, and the start and end page numbers of the subheading. Note that, each table is represented by its table id. The actual table information is additionally organised by this table id in the JSON (it is not seen in the example). Also, we observe that the paragraph and tables are separated by a new line, and we organise each table in the paragraph by using the '[TABLE]' annotation with its id. This makes the paragraph more organised to facilitate QA-R to more efficiently understand the original SoA document for recommending right answers for SAQs.

```
{
   "heading": "Personal Insurance",
   "sub_heading": "Existing cover",
   "content": "We recommend you cancel the following existing insurance
policies \n [TABLE: c70d5065-9826] \n [TABLE: c23d4303-2343] \n",
   "start": 27,
   "end": 28,
   "table_id": [
      "c70d5065-9826",
      "c23d4303-2343"
   ]
}
```

Figure 2: An example SoA JSON

3.2. SoA Question-Answering Recommender: QA-R

In this section, we present how QA-R recommends right answers for SAQs. Based on inputs from domain experts, we propose to categorise SAQs into three levels in terms of their complexity: *easy-level*, *mid-level* and *hard-level complexity*.

QA-R takes a rule-based approach that exploits rules to look for evidence from the corresponding SoA JSON of a given SoA document, in order to answer SAQs. Such rules have been defined, relying on lexical and semantic hints on context, via in-depth knowledge about the general semantics of SoA documents guided by domain experts. Although statistical approaches (e.g., machine learning) have been widely used for question-answering,due to the complex nature of SAQs, we have chosen a rule-based approach for implementing QA-R. The answers generated by QA-R are then used to assess and audit the advice as well as compliance of a given SoA document.

Easy-level Complexity SAQs: Easy-level complexity SAQs involve *keyword-based searching* of text in an SoA document. Some examples of easy-level SAQs include:

- ***SAQ1****: Whether SMSF is advised or not?*
- ***SAQ2****: Is the plan insurance planning only?*
- ***SAQ3****: Is the major content within the first 10 pages?*
- ***SAQ4****: Does the Scope of Advice demonstrate the client's interests have been prioritised?*

Given an easy-level SAQ, QA-R scans text under the relevant heading(s) or subheading(s) to identify whether the keywords indicating the right answers are presented in a given SoA document. Such keywords with their synonyms and their positions are predefined as rules. Thus, QA-R identifies the keywords and examines associations between the keywords including their synonyms within their contexts, considering their headings and subheadings.

More specifically, as presented in Figure 3, QA-R performs the following steps. First, as input to QA-R, a SoA JSON and a SoA vocabulary are provided. According to the given rules, QA-R searches for the required heading(s)/subheading(s) and extracts the text contents (i.e., paragraphs) of them. Then QA-R identifies the required keyword(s) with their synonyms provided by the input vocabulary from the text contents to recommend the answer. If the required keyword(s) are not found, the answer is 'No', otherwise, QA-R checks the relationships between the identified keywords based on the given rules which tries to match according to the context of the keywords. If conditions for satisfied, the answer is 'Yes', otherwise 'No'.

For example, for **SAQ1**, QA-R first searches for the 'scope of advice' heading. Under this heading, it then searches for the keywords such as 'SMSF' or its synonyms, 'Self Managed Super Fund', 'Self-managed' or 'Self managed superannuation'. We also consider the context surrounding these keywords using a rule to determine whether these keywords are 'advised' or 'not'. Thus, we identify whether these keywords are co-occurred with the keyword 'advice' (including its synonyms) or 'not advice' (including its synonyms) within the context. Here, given a keyword, we consider a context as the set of three sentences: the sentence encompassing the keyword, its prior sentence and its post sentence. Depending on the result of the keyword search, QA-R recommends the correct answer.

Mid-level Complexity SAQs: For QA-R to be able to answer mid-level complexity SAQs, context-dependent semantic analysis of text in an SoA document is required. Some examples of mid-level SAQs are:

- ***SAQ5****: Does the advisor recommend the strategy before recommending a product?*
- ***SAQ6****: Where an ongoing review fee/commission is being charged, is there a section offering an ongoing review service?*

The steps of identifying the answer mid-level SAQs follows a similar process to Figure 3. Except in the

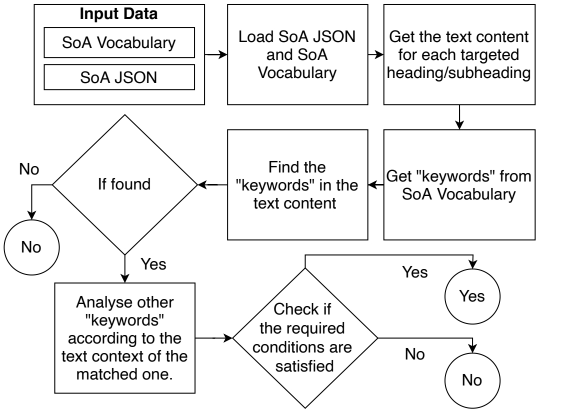

Input Data
SoA Vocabulary
SoA JSON
Load SoA JSON and SoA Vocabulary
Get the text content for each targeted heading/subheading
Get "keywords" from SoA Vocabulary
Find the "keywords" in the text content
If found
No
No
Yes
Analyse other "keywords" according to the text context of the matched one.
Check if the required conditions are satisfied
Yes
Yes
No
No

Figure 3: The process for finding answers for easy-level and mid-level SAQs

"Analyse other keywords step" the QA-R performs an additional semantic analysis based on the context of the matched keywords.

For example, to answer **SAQ5** and **SAQ6**, QA-R searches for the required keywords. Based on the keywords' locations, QA-R analyses the context of the sentences in the text content belong to matched keywords. For example, in **SAQ5**, QA-R first extracts the paragraph in the 'Summary of advice' subheading in which QA-R identifies the sentences with new recommended products by the RFA. This subheading contains the client's current products and the newly recommended products. To distinguish between current and newly recommend product additional semantic analysis is performed by the QA-R. To do so, QA-R first analyses the context, if a product name (e.g., Netwealth, Domacom, AMG, ING, MyNorth) is identified and finds if this keyword (i.e., product name) is related to another keyword 'new'. If the product name is associated to the 'new' keyword, QA-R checks the context-depended semantic whether the text content indicates a newly recommended product by the following rules, (1) If only one product name is found in the text content, the answer will be 'No' as there is no clear evidence whether this is an 'old' or 'new' product name. (2) If more than one product name is found, then based on the context-dependent semantic of the 'new' keyword, QA-R identifies the association of it with newly recommended products. if the recommended products are matched with the context then the answer will be 'Yes' otherwise 'No'. (3) If no product name is found, the answer will be 'No'.

Hard-level Complexity SAQs: Inducing answers for hard-level complexity SAQs involves a flow-based analysis. In such analysis, the rules are predefined in a decision tree. Here, QA-R follows a step-by-step data extraction process (which is explained with an example later in this section) according to the predefined rules in the decision tree to answer the SAQs. Some examples of hard-level SAQs are:

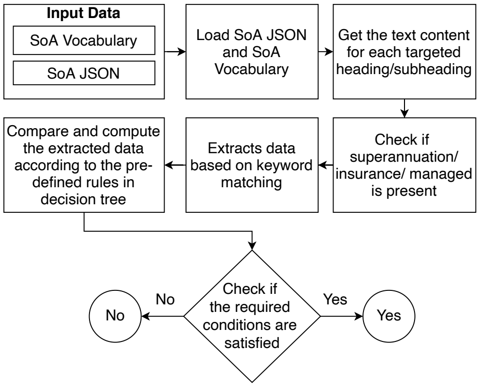

Input Data
SoA Vocabulary
SoA JSON
Load SoA JSON and SoA Vocabulary
Get the text content for each targeted heading/subheading
Check if superannuation/ insurance/ managed is present
Extracts data based on keyword matching
Compare and compute the extracted data according to the pre-defined rules in decision tree
Check if the required conditions are satisfied
No
No
Yes
Yes

Figure 4: The process for finding answers for hard-level SAQs

- ***SAQ7***: *Has adviser compared the recommended strategy with one or more adequate or appropriate alternatives?*
- ***SAQ8***: *Does the plan demonstrate the benefits of the recommended strategy are superior to alternatives?*

As presented in Figure 4, QA-R performs the following to answer hard-level SAQs. First, given the SoA vocabulary and SoA JSON, QA-R identifies targeted heading(s)/subheading(s) predefined by given rules. QA-R in particular checks if any of the three strategies (*superannuation*, *insurance* and *managed funds*) are recommended in the SoA JSON by analysing the subheadings under the "Strategy" heading. If any of these strategies are found, QA-R extracts relevant data (e.g. current product name, current superannuation amount, recommended superannuation amount) from the SoA JSON to find the answer for a given SAQ. The extracted data is then analysed through the step-by-step process in the decision tree. The example of some step-by-step process are: numerical comparison such as compare current and recommended superannuation amount, numerical computation such as compute the recommended superannuation benefit, pattern matching such as client satisfaction on the current product. Overall, QA-R checkd if all such conditions (please see example below) are satisfied for the SAQs.

Rules in decision-tree have been defined in QA-R for three strategies: insurance, superannuation and managed fund, to induces answers for **SAQ7** and **SAQ8**. For example, in **SAQ8**, the rules in insurance category extract the content for each subheading under strategy heading from the SoA JSON. It also extracts client(s) name (to whom the financial advice group provides the advice). Multiple client names may occur in the SoA document such as partners. QA-R extracts the clients' insurance (e.g., current insurance and recommended insurance), recommended insurance products, alternatives products and alternative strategies (e.g., Stepped vs Level, Inside/Outside Super, Any/Own Occupation and Self-insure), annual amount, annual premium cost, industrial insurance product and premium type for both current and recommended insurance for the client(s). Some insurance types require special information which is only relevant for those insurance types. For example, QA-R extracts the information 'waiting and benefit periods' and 'agreed/indemnity' only for Income Protection (IP) insurance. As another example, it extracts the information about 'any/own occupation' for Total and Permanent Disability (TPD) insurance. Afterwards, QA-R extracts more complex information such as client's goals, purposes for discounting client's current insurance, and provides the evidence showing that the adviser has recommended the lower cost products comparing to the client's current insurance.

Using the extracted information, QA-R applies the following steps to answer the question for the insurance category: **Step 1**: If the client has an existing insurance, go to step 2. Otherwise, the answer for **SAQ8** (insurance) will be 'No'. **Step 2**: If the adviser considers the client's existing insurance, go to step 3 Otherwise, go to step 6. **Step 3**: If there is an evidence showing the adviser has recommended lower cost products, go to step 5. Otherwise, go to step 4. **Step 4**: If there are identified causes for recommended products, go to step 5. Otherwise, go to step 6. **Step 5**: If the adviser compared products in terms of dollar, go to step 7 . Otherwise, go to step 6. **Step 6**: If there are identified causes for discounting current products, go to step 7. Otherwise, the answer for **SAQ8** (insurance) will be 'No'. **Step 7**: If there are evidence showing the satisfaction in the client goal(s), the answer will be 'Yes'. Otherwise, the answer will be 'No'. Similarly the answers from all categories are aggregated to induce the final answer for **SAQ8**.

QA-R provides answers for hard-level complexity SAQs with the evidence of relevant context (e.g., current product, recommended product, recommended cost) and corresponding identified values for each of the three categories (insurance, superannuation, managed fund).

4. Evaluation

In this section, we evaluate SRAuditor using real-life SoA documents. Our goal here is to measure the performance of SRAuditor for assessing and auditing SoA documents against manual assessments conducted by domain experts (i.e., a financial adviser group).

4.1. Evaluation Settings

We evaluate SRAuditor using eight SAQs presented in Section 3.2 on 200 SoA documents. These SoA documents follow two templates: Midwinter [5] and XPlan [6]. Our sample data contain 100 SoA documents from each template. Since SoA documents differ in structure (i.e., different headings and subheadings) based on template, all 200 SoA documents are mapped to the generic template structure presented in Table 1 in consultation with domain experts from Fourth Line. The experts included two financial advisors, a commercial lawyer who advises in this area, and a professor of law, all of whom are on staff or in the management of Fourth Line, the commercial partner in the project. We have also formally engaged 5 client contacts of Fourth Line, all of whom are in charge of risk and compliance for the groups which hold the financial services licences that the advisors work under.

Though both templates contains SoA information, they have significant structural variations. Midwinter mostly presents information using lines of text (i.e., paragraphs), however, XPlan represents similar information in a two-columns tabular format where the left column indicates the 'title' and the right column contains the 'description' of the information. Moreover, key information to find the answer for a SAQ is located under different headings/subheadings in different templates. To handle this, SoA-T first extracts the information according to the process descried in Section 3 and then each SoA document is converted to the generic structure in Table 1. Some examples of mapping key information in the two templates to our generic schema (SoA-S) are presented below.

- 'Executive Summary', a heading in Midwinter, contains client's goals, scope, summary and outcomes of the advice as subheadings. However, in XPlan such information is located under different headings namely 'Goals and Objectives', 'Scope of our Advice',' Your Current Situation' etc. This information are mapped to 'No. 2' in Table 1.

- 'Personal Details' represents client's personal information, current lifestyle assets, current superannuation, current insurance, current managed fund, liabilities, net worth and others. This information is located under the Appendix heading in XPlan, however, is placed under the 'Your current situation' heading in Midwinter.

This information is mapped to the heading and sub-headings described in 'No. 3' in Table 1.

- 'Strategy' covers the advice that the client needs for different aspects such as product recommendation, risks, alternatives and consequences. 'Risk' performs the analysis on the client's risk profile, asset allocation and needs analysis. This information are located under different headings in Midwinter, however, can be found in table under the Appendix header in XPlan. This is mapped to the heading and sub-headings described in 'No. 4' in Table 1.
- 'Disclosure' shows fees of the advice as well as the client's authority to proceed the advice. 'Additional Information' contains pieces of information related to research, projections and product's profiles. This information can be found under the Appendix heading in both Midwinter and XPlan. This is mapped according to 'No. 6' in Table 1.

The 200 SoA documents were manually audited by a financial adviser group. This group marked the correct answers (Yes or No) of the easy-level and mid-level complexity SAQs. For the hard-level complexity SAQs, the correct answers were marked according to the three categories (i.e., insurance, superannuation and managed fund) along with the final answers (Yes or No). These answers were used as the ground-truth answers that were compared to the recommended answers of QA-R.

To produce the quality of the recommended answers, a confusion matrix for both templates (Midwinter and XPlan) is produced using true positive (*tp*), true negative (*tn*), false positive (*fp*) and false negative (*fn*) values. Finally, as the quality metric, we used 'accuracy' by calculating (*tp*+*tn*)/(*tp*+*tn*+*fp*+*fn*) for the 100 SoA documents of each template. Afterwards, the averaged accuracy is calculated by taking the mean accuracy across the two templates. We used only accuracy as our evaluation measure as this is based on the confusion matrix of the one-to-one comparison between QA-R generated yes/no answer which is designed using rules encoded in decision trees and human audited yes/no answer of each SAQ for each SoA. Therefore, accuracy was sufficient for our case.

4.2. Evaluation Results

We first present the evaluation results for the easy-level and mid-level complexity SAQs, followed by the results for the hard-level complexity SAQs. Moreover, for human-level evaluation we have developed a web user interface (Web-UI) of SRAuditor. This Web-UI (see Figures 5 and 6) can be used by a domain expert to cross check his/her audited answer and with the answer recommended by QA-R in SRAuditor for a SAQ. The Web-UI also assists domain experts to verify the correctness of the answer by providing option to visualise the detail evidence for the recommended answer.

Assessment for easy and mid-level complexity SAQs: Table 2 shows the accuracy for the six SAQs (SAQ1 - SAQ6) on the Midwinter and XPlan templates. The key observations from Table 2 are summarised as follows. First, the performance of first 3 easy-level SAQs (1 to 3) is very high (95% or above on average). Second, the performance of SAQ4 was not good (only 70% on average). The reason of this low accuracy stems from the ambiguity in co-occurrence keywords in several SoA documents - that is many 'yes' answers are predicted as 'no'. Such ambiguity occurs due to frequent use of some common keywords in the vocabulary (e.g., cash, debt) across both superannuation and managed funds. Third, the performance of mid-level SAQs (5 and 6) is also high (90% or above on average).

SAQ	Midwinter	XPlan	Average
1	96%	94%	95%
2	96%	93%	95%
3	95%	97%	96%
4	74%	66%	70%
5	93%	89%	91%
6	89%	91%	90%

Table 2: Accuracy of QA-R for the easy-level and mid-level complexity SAQs (SAQ1 - SAQ6)

Assessment for hard-level complexity SAQs: As discussed previously, answering hard-level complexity SAQs involves significant qualitative assessment by a financial advisor group. Thus, it is a labour-intensive and time-consuming process even for a domain expert to generate the ground-truth answers. Further, to recommend correct answer to such SAQs, three categories (i.e., insurance, superannuation, managed fund) are required to be audited separately. Thus, it can be very hard to generate many benchmark samples for hard-level complexity SAQs. In our experiment, we choose 40 SoA documents in both templates with the ground-truth answers for our evaluation. Here, we report accuracy of two categories (insurance and superannuation) evaluated over these 40 documents (20 each from Midwinter and XPlan) for SAQ7 and SAQ8. The evaluation results are presented in Table 3 . We observe the following from Table 2. First, for SAQ7, QA-R performs equally to the ground-truth in terms of the superannuation. In the insurance category, only one Midwinter document answer was incorrect, however, the overall accuracy is very high reached 98% across both templates. Second, For SAQ8, insurance and superannuation category produces the average accuracy around 85% and 87% respectively. Third, the average accuracy for hard-level SAQs observed higher than some easy and mid-level SAQs. This is simply because the number of benchmark samples used to evaluate hard-level SAQs was lower (40) than those of the easy and mid level SAQs (200). One of our future works is to do more exact performance comparison using more ground-truth samples.

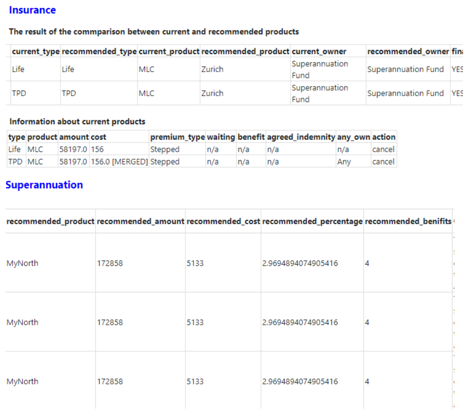
Cat.	Midwinter		XPlan		Average	
SAQ	7	8	7	8	7	8
Insur.	96%	86%	100%	85%	98%	85%
Super	100%	94%	100%	80%	100%	87%

Table 3: Accuracy of QA-R for the Insurance (Insur.) and Superannuation (Super) categories for SAQ7 and SAQ8.

User Interface Human-level Evaluation: A domain expert can use our developed Web-UI to validate the manually audited answer against recommended answer by SRAuditor. The Web-UI has option for its user to choose a SoA document of a specific template and a SAQ. Then, the Web-UI presents the recommended answer for the selected SAQ and the evidence (by highlighting the keywords and associated contexts) about how the answer was generated. In our human-evaluation, there were some cases SRAuditor identified human errors. For example, SRAuditor predicted the correct answer with the detail evidence which was missed by domain expert in manual assessment. This can be further verified with the help of evidence shown in the Web-UI.

Figure 5 shows the UI displaying the keywords along with contexts evidence for SAQ2. In the 'Choose a question' panel a SAQ is selected (here SAQ2). The 'Answer' panel shows the recommend answer (which is 'No') along with the chosen SoA JSON containing a hierarchical structure of relevant contexts (heading, subheadings, paragraph) in the SoA. The evidence of the answer is highlighted in 'More Detailed Evidence' panel. Here the detail evidence shows what is advised and what is not advised. Under what is advised on text, insurance is not identified so the answer became 'No'.

Figure 6 presents a partial view of the UI for displaying the evidence for a hard-level complexity SAQ. Here some extracted information using the rules for insurance (current insurance, recommender insurance, current product) and superannuation (recommended product, recommended super amount) category are presented. Such detailed information can help a RFA to improve the understanding the induced answer.



Insurance

The result of the commparison between current and recommended products

current_type	recommended_type	current_product	recommended_product	current_owner	recommended_owner	fin
Life	Life	MLC	Zurich	Superannuation Fund	Superannuation Fund	YES
TPD	TPD	MLC	Zurich	Superannuation Fund	Superannuation Fund	YES

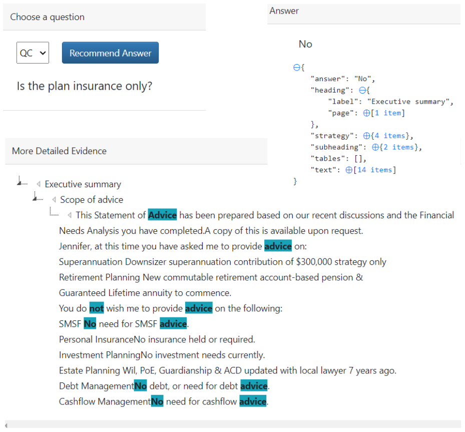
Information about current products

type	product	amount	cost	premium_type	waiting	benefit	agreed_indemnity	any_own	action
Life	MLC	58197.0	156	Stepped	n/a	n/a	n/a	n/a	cancel
TPD	MLC	58197.0	156.0 [MERGED]	Stepped	n/a	n/a	n/a	Any	cancel

Superannuation

recommended_product	recommended_amount	recommended_cost	recommended_percentage	recommended_benifits
MyNorth	172858	5133	2.9694894074905416	4
MyNorth	172858	5133	2.9694894074905416	4
MyNorth	172858	5133	2.9694894074905416	4

Figure 5: : An example of the evidence along with recommend answer by QA-R for an easy-level SAQ



Is the plan insurance only?

No

```
⊖{
    "answer": "No",
    "heading": ⊖{
        "label": "Executive summary",
        "page": ⊕[1 item]
    },
    "strategy": ⊕{4 items},
    "subheading": ⊕{2 items},
    "tables": [],
    "text": ⊕[14 items]
}
```

Executive summary

Scope of advice

This Statement of Advice has been prepared based on our recent discussions and the Financial Needs Analysis you have completed.A copy of this is available upon request.
Jennifer, at this time you have asked me to provide advice on:
Superannuation Downsizer superannuation contribution of $300,000 strategy only
Retirement Planning New commutable retirement account-based pension &
Guaranteed Lifetime annuity to commence.
You do not wish me to provide advice on the following:
SMSF No need for SMSF advice.
Personal InsuranceNo insurance held or required.
Investment PlanningNo investment needs currently.
Estate Planning Wil, PoE, Guardianship & ACD updated with local lawyer 7 years ago.
Debt ManagementNo debt, or need for debt advice.
Cashflow ManagementNo need for cashflow advice.

Figure 6: Example evidence for a hard-level SAQ

4.3. Discussion

SRAuditor can be used internally by a compliance company (i.e., Fourth Line) which works with risk and compliance managers of financial advice group who are responsible for the quality and accuracy of a SoA. The reports generated by SRAuditor can also be useful for compliance professionals within the compliance company to identify deficiencies in a SoA.

A key purpose of designing SRAuditor is to minimise the regulatory and compliance risk. The failures in this area generally mean penalties enforced by regulators on the groups, although they occasionally may involve litigation. Compliance at scale for thousands of SoAs is not primarily a matter for lawyers but rather those within the risk and compliance function of corporations.

The compliance products of Fourth Line (of which SRAuditor is a part) have been designed by lawyers and are overseen by them. Specifically, the accuracy of the review generated by SRAuditor has been checked by commercial lawyers working for Fourth Line.

5. Conclusion

In the digital era, demand for automated assessment and auditing of regulatory documents such as SoA is growing rapidly and has gained significant attention with advances in NLP techniques. In this paper, we proposed SRAuditor, a NLP framework for automated assessment of SoA documents to support automated auditing. SRAuditor incorporates two major components: 1) SoA transformer (SoA-T) that transforms SoA documents into the proposed generic SoA scheme (SoA-S) and eventually builds a machine-interpretable JSON object (SoA JSON); and 2) the question-answering recommender (QA-R) that can recommend right answers for SoA audit questions (SAQs). Our experimental evaluation using 200 real-world SoA documents validated SRAuditor's accuracy and its ability to support automated auditing. SRAuditor produced an accuracy of over 95% for easy- and medium-level complexity SAQs while producing an average accuracy more than 80% for complex-level SAQs. The results validate the efficacy of the rule-based approach used by SRAuditor to support automated auditing.

There are also significant implications for legal and compliance practitioners from the methodology described in this paper. For compliance practitioners, our approach promises the ability to manage regulatory risk at scale in an environment where regulators are demanding greater compliance but where compliance costs need to be controlled. For legal practitioners, our approach demonstrates that modern NLP techniques can be combined with traditional knowledge engineering techniques to generate semantic understanding of legal/regulatory text.

SRAuditor is evaluated only for a limited number of questions which is one of the limitation of this approach. Therefore, as future work, we plan to extend and test SRAuditor to provide answers for a large number of regulatory compliance assessment questions and expand its support for more SoA templates.

Acknowledgement

This research was supported by the industry funding provided by Fourth Line Pty Limited and by use of the Nectar Research Cloud, a collaborative Australian research platform supported by the NCRIS-funded Australian Research Data Commons (ARDC).

References

[1] K. L. Choi, "Demand for financial advice doubled in the last five years: Investment trends 2020 financial advice report."

[2] The Australian Securities and Investments Commission, "Asic enforcement outcomes: January to june 2018." `https://www.paraplanner.com.au/example-statement-of-advice-pages.html`.

[3] S. A. Chen, A. J. Makarucha, N. Alam, W. Sherchan, S. Harris, G. Yiapanis, and C. J. Butler, "Evaluating goal-advice appropriateness for personal financial advice," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1–5, 2019.

[4] AISC, "Example statement of advice pages." `https://www.paraplanner.com.au/example-statement-of-advice-pages.html`, 2017.

[5] Midwinter. `https://www.midwinter.com.au/`, 2020.

[6] Xplan. `https://www.iress.com/software/financial-advice/xplan/`, 2020.

[7] M. Spruit and D. Ferati, "Text mining business policy documents: Applied data science in finance," *International Journal of Business Intelligence Research (IJBIR)*, vol. 11, no. 2, pp. 1–19, 2020.

[8] W. Breymann, N. Bundi, J. Heitz, J. Micheler, and K. Stockinger, "Large-scale data-driven financial risk assessment," in *Applied Data Science*, pp. 387–408, Springer, 2019.

[9] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, and D. C. L. Ngo, "Text mining for market prediction: A systematic review," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7653–7670, 2014.

[10] M. R. Vargas, B. S. De Lima, and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles," in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 60–65, IEEE, 2017.

[11] J. Huang and M. Chen, "Domain adaptation approach for credit risk analysis," in *Proceedings of the 2018 International Conference on Software Engineering and Information Management*, pp. 104–107, 2018.

[12] J. Zhang and N. M. El-Gohary, "Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking," *Journal of Computing in Civil Engineering*, vol. 30, no. 2, p. 04015014, 2016.

[13] I. E. Fisher, M. R. Garnsey, and M. E. Hughes, "Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research," *Intelligent Systems in Accounting, Finance and Management*, vol. 23, no. 3, pp. 157–214, 2016.

[14] G. Boella, L. Di Caro, and V. Leone, "Semi-automatic knowledge population in a legal document management system," *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 227–251, 2019.

[15] AWS, "Amazon textract." `https://aws.amazon.com/textract/`.