Towards Automated Auditing with Machine Learning

Rafet Sifa, Anna Ladi, Maren Pielka, Rajkumar Ramamurthy, Lars Hillebrand, Birgit Kirsch, David Biesner, Robin Stenzel, Thiago Bell, Max Lübbering, Ulrich Nütten, Christian Bauckhage
Fraunhofer IAIS
Germany

Ulrich Warning, Benedikt Fürst, Tim Dilmaghani Khameneh, Daniel Thom, Ilgar Huseynov, Roland Kahlert, Jennifer Schlums, Hisham Ismail, Bernd Kliem, Rüdiger Loitz
PriceWaterhouseCoopers GmbH WPG
Germany

ABSTRACT

We present the Automated List Inspection (ALI) tool that utilizes methods from machine learning, natural language processing, combined with domain expert knowledge to automate financial statement auditing. ALI is a content based context-aware recommender system, that matches relevant text passages from the notes to the financial statement to specific law regulations. In this paper, we present the architecture of the recommender tool which includes text mining, language modeling, unsupervised and supervised methods that range from binary classification models to deep recurrent neural networks. Next to our main findings, we present quantitative and qualitative comparisons of the algorithms as well as concepts for how to further extend the functionality of the tool.

KEYWORDS

Text Mining, Business Process Optimization, Automated Auditing

ACM Reference Format:
Rafet Sifa, Anna Ladi, Maren Pielka, Rajkumar Ramamurthy, Lars Hillebrand, Birgit Kirsch, David Biesner, Robin Stenzel, Thiago Bell, Max Lübbering, Ulrich Nütten, Christian Bauckhage and Ulrich Warning, Benedikt Fürst, Tim Dilmaghani Khameneh, Daniel Thom, Ilgar Huseynov, Roland Kahlert, Jennifer Schlums, Hisham Ismail, Bernd Kliem, Rüdiger Loitz. 2019. Towards Automated Auditing with Machine Learning. In *ACM Symposium on Document Engineering 2019 (DocEng '19), September 23–26, 2019, Berlin, Germany.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3342558.3345421

1 INTRODUCTION AND MOTIVATION

The purpose of an audit of financial statements is to enhance the degree of confidence of intended users in the financial statements by expressing an opinion by the auditor on whether the financial statements are prepared, in all material respects, in accordance with an applicable financial reporting framework. The auditing process requires a high degree of expert knowledge and judgment. However, it also includes recurring and time consuming tasks, which can be automated using artificial intelligence and machine learning techniques. In this work we focus on such a task, namely the audit of a company's notes and disclosures to the financial statement,

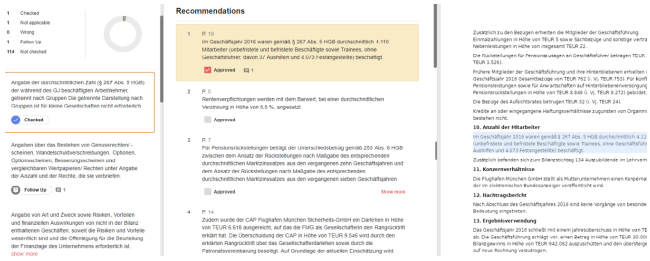


Figure 1: Screenshot of the engagement view of our accounting list recommender system. The left column contains progress indicators and checklist questions, the column in the middle displays the recommendations, and the column on the right highlights recommended results in the report.

which aim at providing further details and communicating the accounting methods and practices used by the company (examples under: https://www.bundesanzeiger.de/ebanzwww/wexsservlet). The audit of this part of the document involves but is not limited to the following tasks:

- ensuring the completeness of the financial statements, according to specific legal requirements
- ensuring the accuracy and valuation of the reported numbers and calculations in notes and primaries
- ensuring the consistency of information both within the notes to the financial statement, but also against external data sources.
- ensuring the classification and understandability of information as well as the occurrence of events and transactions disclosed in the notes.

In this work, we will present a machine learning based context-aware [13] recommender system toolbox that assists the auditor with the aforementioned tasks; our main goal is not to replace auditors, but to provide them with solutions that improve the speed and quality of the audit process. Our approach follows the paradigm of *Informed Machine Learning* [16], in the sense that it combines machine learning algorithms with expert knowledge to achieve optimal performance. ALI is designed to address all of the four tasks above. However, this paper focuses mainly on the first point, i.e. checking a statement in the notes to the financial statement (in the following referred to as *document*) against a list of legal requirements.

Our remaining presentation is organized as follows: We continue with a related work section covering machine learning backed process optimization and explain the relevant procedures in the

DocEng '19, September 23–26, 2019, Berlin, Germany

ACM ISBN 978-1-4503-6887-2/19/09…$15.00
https://doi.org/10.1145/3342558.3345421

audit of the notes to financial statements. Following that we present our recommender system along with our evaluations. The paper concludes with an outline of ongoing and further development and optimization steps.

2 RELATED WORK

Since our main focus in this work is on building a text-based recommender system, we next briefly review prior work on text analysis in the context of auditing. Indeed, the growing amount of textual data combined with continuous advances in natural language processing has significantly influenced recent research in this domain.

Work by Gepp et al. [5] identifies two closely related use cases for automated text analysis on annual reports: financial fraud detection and distress modelling. Concerning fraud detection, Hajek and Henriques [6] leverage Bayesian Belief Networks on a diverse feature set which comprises of financial statement-, annual report- and analyst forecast information. In contrast, a pure textual approach based on linguistic credibility analysis is proposed in [11] and [7]. Other researchers like Tsai and Wang [15] and Campbell et al. [2] employ sentiment analysis for financial risk prediction. They analyze relations between risk factor disclosures and financial sentiment words, to assess their information content. Following a similar goal, Matin et al. [10] estimate corporate distress probabilities by using a convolutional recurrent neural network on text segments in annual reports.

Considering these applications of text based machine learning models, it comes as a surprise that auditors today still heavily rely on manual analysis. To our knowledge, this paper is the first in providing evidence on how to successfully incorporate machine learning and natural language processing into auditing procedures and thereby increase their efficiency and quality.

3 PROCESSING AUDIT CHECKLISTS WITH MACHINE LEARNING

One of the core components of the system we have developed is the automated processing of a disclosure checklist. An audit checklist is a list of legal requirements[1] with which a company's financial statements need to comply. A requirement can be as simple as *"Disclosing the name of the reporting entity"* or as quite complex as asking for a condition that refers to specific (sometimes judgmental) information spread throughout the text.

The current process involves the auditor reading through the notes to the financial statement and determining for every requirement where the relevant text passage is. This task is time consuming, as the size of the checklist can vary from a couple of hundred to a couple of thousand requirements, according to the type of company, the industrial branch, and the financial reporting framework. Similarly, the notes to the financial statements is a document that can also vary in length from a few pages to several hundred pages.

Following the assumption that the requirements must be answered in a structured and consistent manner, a machine learning algorithm can be trained in order to learn which text passage corresponds to which requirement. The paradigm of a recommender system allows the auditor to make the final decision by confirming the correct recommendations.

In the following sections we present the algorithms and the developed framework that recommends for every requirement the relevant text passage using a combination of supervised and unsupervised methods.

[1]The list of legal requirements for two different reporting standards (german GAAP and IFRS) can be found under http://www.gesetze-im-internet.de/hgb/ and https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:02008R1126-20180101&from=EN

4 RECOMMENDER SYSTEM

Having explained the general steps taken when processing accounting checklists, we will turn our attention to explaining the main architecture of our recommender system tool. We define two main entities: a set of requirements and a document that is under audit. We will use the term *requirement* r_i to describe a specific legal requirement that has to be answered in the document we are analyzing. Additionally, we assume that the document can be represented as a set of non-overlapping text blocks. Such a block, which can be a paragraph, title or table, will in the following be referred to as a *structure of interest* (SoI) s_j.

Given a set of n requirements $R = \{r_i | i \in [0, n]\}$ and a set of m SoIs $S = \{s_j | j \in [0, m]\}$, our recommender system ranks for every requirement in R, the SoIs in S with respect to the content relevance. The system consists of three main components (that we describe in detail below): text pre-processing, data representation methods, and SoI-ranking models. These components are responsible for 1) standardizing the natural language using linguistic or domain specific methods, 2) parsing the documents and representing each requirement and SoI for further processing and 3) utilizing unsupervised and supervised machine learning methods to rank the SoI for each requirement.

4.1 Text Pre-processing

In this work, in extension to the typical character- and word-based (for instance removing digits and lower-casing text) and linguistic (stemming, lemmatization) text pre-processing steps [9], we have utilized application specific pre-processing steps in cooperation between data scientists and domain experts. This includes but is not limited to detecting and normalizing currencies indicators, dates and legal references (e.g. paragraph indices), detecting and protecting specific (e.g. accounting) terms from further processing, detecting section titles. This domain specific pre-processing not only standardizes the text but also enriches it semantically.

4.2 Document Representation and Language Modeling

There are numerous ways to represent textual data for content matching and recommender systems which can be categorized based on the way textual similarities are measured [9, 13]. Our recommender system can represent a given text block in terms of:

- N-**Grams**: a set of overlapping character n-grams.
- **Bag-of-Words**: a joint vector-space representation of all the words that are contained in it. In this representation, the word order is not taken into account. A typical choice is to represent text blocks using their term frequency (TF)

or their term frequency - inverse document frequency (TF-IDF) representations[9, 12]. Furthermore, matrix factorization (MF) methods[4, 13] can be built on top of such representations to automatically capture and model latent semantic structures and reduce the dimensionality of the data.

- **Neural Language Models**: a vector embedding produced by context aware neural language models (for example the so-called *doc2vec* paragraph embeddings [8]).

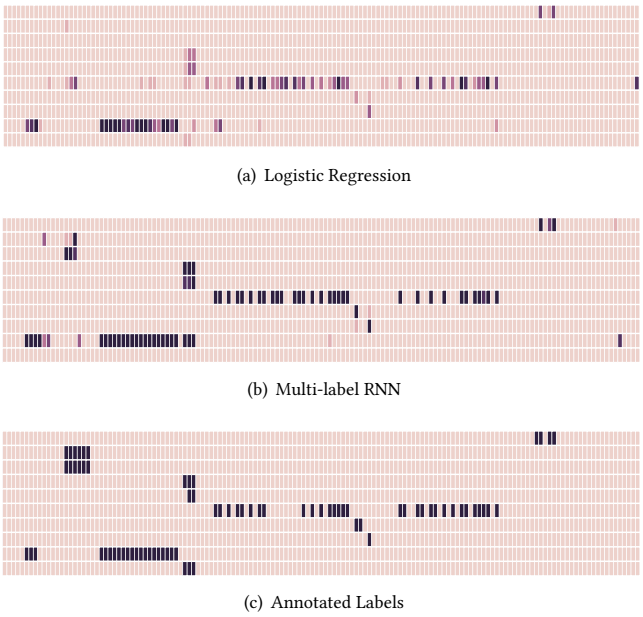
4.3 Ranking Models

In the following we categorized our ranking models based on training them with the available labels as unsupervised and supervised models. The advantage of models belonging to the former categories lies in the fact that they can be immediately deployed in case of change in the checklist. On the other hand as we will observe in our evaluations, they are not as performant as the models belonging to the latter category.

4.3.1 Unsupervised ranking. Representing both SoIs and requirements in the selected space (see section 4.2), the similarity of a given requirement to every SoI is computed using a similarity measure which can be chosen to be the Jaccard- or Tversky-index in the case of an n-gram representation, or the cosine similarity in the case of a vector space representations. Following that, we rank the SoI based on the computed similarity for the given requirement. The unsupervised ranking is based on the assumption that the text of the SoI resembles the text of the requirement itself (or a model answer for the requirement if available). Even though this assumption does not always hold, the unsupervised ranking is still useful in the case of new or rare requirements for which there are no previously seen examples.

4.3.2 Supervised Ranking. Given an SoI representation (such as the TF-IDF or lower dimensional MF vector), a probabilistic supervised ranking model can be trained to predict the binary relevance of that SoI to each of the requirements. Once such a model has been trained, during inference, for a given document consisting of m SoIs, for each requirement, we could retrieve all relevant SoIs by simply sorting them based on their probabilities. We tackled this by considering the following schemes: 1) training n binary classifiers, where each model predicts the relevance of given SoI to a requirement and 2) considering a multi-label classifier that predicts the relevance to all requirements using a single neural network.

Considering the former, for each requirement $r_i \in R$, we train a binary classifier which takes the SoI representation as its input and predicts relevance probability to the requirement. In our case, we chose this to be a logistic regression (LR) classifier that is trained to minimize the logistic loss. Note that, here, each requirement is treated independently of the others (i.e. each requirement has its own model). Turning our attention to the latter scheme, as one SoI could belong to many requirements, we also chose to train a multi-label classifier that maps a given SoI to a binary relevance vector w.r.t. all the requirements. In our case, we chose this to be a feed forward neural network model that is trained to minimize the binary cross-entropy loss. In addition, to incorporate structural dependencies between the requirements and the SoIs (for instance the order of appearance), we considered a multi-label recurrent neural network (RNN), which we implemented as a Gated Recurrent Units (GRUs) [3] network.



(a) Logistic Regression

(b) Multi-label RNN

(c) Annotated Labels

Figure 2: Visualization of document parsing and mapping to requirements using heatmaps. The x-axis corresponds to SoIs in a document and the y-axis corresponds to requirements. (a) predictions from a logistic regression model, (b) predictions from a multi-class RNN, and (c) ground truth provided by domain experts.

5 EVALUATIONS

Existing popular evaluation metrics in the recommender systems literature (see, for instance, the examples in [13]) such as precision and recall can be used to evaluate a recommender system. However for this application we also want to evaluate the recommendations, given the constraint that a limited number of results, L, can be presented to the user. Our metric is based on the *sensitivity* measure, which in the context of machine learning, is commonly defined as the ratio of true positives among all positive examples. Our modified sensitivity measure is calculated on a requirement-level taking into account the L top-ranked recommended SoIs for each requirement:

$$Sensitivity = \frac{|\text{top } L \text{ recommendations } \cap M \text{ annotations}|}{\min(L, M)}, \quad (1)$$

where L is a parameter usually chosen to be 3 or 5 (for evaluation), and M is the list of annotated passages in the notes to the financial statements. It is important to note here that L is fixed, while M can vary between requirements. This metric accounts for the fact that we always predict L examples, making it possible to reach a perfect score if M is larger than L.

We evaluated our recommender systems by comparing supervised and unsupervised approaches as well as different data representations and pre-processing options on a data set of 150 financial

Model	Performance
Unsupervised model (n-gram-similarity)	0.801
Binary supervised model (logistic regression)	0.857
Multi-label supervised model (NN)	0.854
Multi-label supervised model (RNN)	0.791

Table 1: Weighted average sensitivity per requirement with L=3, w.r.t. requirement frequency in all reports

reports in German language. In this section we report only on our most significant findings.

Considering an illustrative example, in Fig. 2, we show a visualization of predictions from our supervised classifiers using heat maps for an example document. Each heat map represents a document consisting of several SoIs (x-axis) which are mapped to requirements (y-axis) by a classifier. In Figure 2 (a) and (b), we can see mappings obtained from logistic regression and recurrent neural networks, respectively. Fig 2 (c) shows the actual annotations by an expert. As we can see, the predictions from both classifiers are very close to the actual mappings while context-aware RNNs capture sequential information better compared to logistic regression.

In order to measure the overall recommendation performance of the models we investigated, we split our set of 150 documents into training and test sets. A static training/test set split was selected over cross-validation, so that the results could be analyzed both quantitatively and qualitatively on the same data set. For the supervised model, we used 80 % of the whole data set (120 reports) to train the models and the remaining 20 % (30 reports) to evaluate the models. Additionally, we extended our training set using automatically generated annotations. We used our unsupervised models to recommend candidate SoIs for requirements where a low number of annotated examples was available. These candidates where then evaluated and finally manually annotated by domain experts. In case of our unsupervised method, we are evaluating using the same 20 % reports as above so as to ensure consistency and comparability.

It is worth mentioning that, simple n-gram similarity ranking already works well for many requirements, where the SoI answering a requirement is formulated in a very similar way to the requirement itself. For others, it is not sufficient, as the SoIs do not have text in common with the requirement. Using supervised classifiers, we get good results for most of those requirements, given that they are sufficiently represented in our data. We considered TF-IDF and MF vector as inputs to the binary classifier and multi-label classifiers, respectively. The neural network architecture is chosen to consist of two hidden layers with 300 and 200 units. An overview on the evaluation is presented in table 1. We are using the weighted average sensitivity (defined in(1)) over all requirements as a performance criteria, where the weight for each requirement is the number of reports it appears in. Prior to this step, the performance has already been averaged over all SoIs and reports for each requirement.

6 CONCLUSION AND OUTLOOK

In this application paper, we have presented a core component of the ALI tool, namely a context-aware recommender system for the auditing of financial statements. Text mining and Machine Learning techniques are used in a novel domain in order to reduce the manual effort of the auditor. The developed system is generic and can be easily tuned and adapted to other checklists and applications with similar formulations. We are currently further optimizing our system, considering more state-of-the-art data representation models [1] as well as domain-specific training for our supervised models involving non-differentiable custom ranking loss functions using the Simultaneous Perturbation Stochastic Approximation (SPSA) [14] optimization scheme.

Checking the report against legal requirements is only a single step of the audit of financial statements. An equally important aspect is checking the document for consistency: It is not enough for a piece of relevant information to be there, rather, that information must be complete, correct, and consistent throughout a document. A corresponding consistency check module is currently under development and combines machine learning, intelligent document parsing, and rule based algorithms to extract and verify information in tables and text.

7 ACKNOWLEDGMENTS

We would like to thank the Helix team at PwC Germany for the integration and deployment of the developed recommender system. This research is partly supported by the Fraunhofer Center for Machine Learning.

REFERENCES

[1] A. Akbik, D. Blythe, and R. Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proc. Int. Conf. on Computational Linguistics*.
[2] J.L. Campbell, H. Chen, D.S. Dhaliwal, H. Lu, and L.B. Steele. 2014. The information content of mandatory risk factor disclosures in corporate filings. *Review of Accounting Studies* 19, 1 (2014).
[3] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
[4] G.W. Furnas, S. Deerwester, S.T. Dumais, T.K. Landauer, R.A. Harshman, L.A. Streeter, and K.E. Lochbaum. 1988. Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure. In *Proc. SIGIR*. ACM.
[5] A. Gepp, M.K. Linnenluecke, T.J. O'Neill, and T. Smith. 2018. Big data techniques in auditing research and practice: Current trends and future opportunities. *Journal of Accounting Literature* 40 (2018).
[6] P. Hajek and R. Henriques. 2017. Mining corporate annual reports for intelligent detection of financial statement fraud–A comparative study of machine learning methods. *Knowledge-Based Systems* 128 (2017).
[7] S.L. Humpherys, K.C. Moffitt, M.B. Burns, J.K. Burgoon, and W.F. Felix. 2011. Identification of fraudulent financial statements using linguistic credibility analysis. *Decision Support Systems* 50, 3 (2011).
[8] Q. Le and T. Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proc. ICML*.
[9] C. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
[10] R. Matin, C. Hansen, C. Hansen, and P. Mølgaard. 2018. Predicting Distresses using Deep Learning of Text Segments in Annual Reports. *arXiv:1811.05270* (2018).
[11] L. Purda and D. Skillicorn. 2015. Accounting variables, deception, and a bag of words: Assessing the tools of fraud detection. *Contemporary Accounting Research* 32, 3 (2015).
[12] J. Ramos. 2003. *Using TF-IDF to Determine Word Relevance in Document Queries*. Technical Report. Rutgers University Department of Computer Science.
[13] F. Ricci, L. Rokach, and B. Shapira. 2011. *Recommender Systems Handbook*. Springer.
[14] J.C. Spall. 1992. Multivariate Stochastic Approximation using a Simultaneous Perturbation Gradient Approximation. *IEEE Trans. Automat. Control* 37, 3 (1992).
[15] M. Tsai and C. Wang. 2017. On the risk prediction and analysis of soft information in finance reports. *European Journal of Operational Research* 257, 1 (2017).
[16] L. von Rueden, S. Mayer, J. Garcke, C. Bauckhage, and J. Schuecker. 2019. Informed Machine Learning – Towards a Taxonomy of Explicit Integration of Knowledge into Machine Learning. *arXiv:1903.12394* (2019).