Zero-Shot Text Matching for Automated Auditing using Sentence Transformers

1st David Biesner  
Fraunhofer IAIS and University of Bonn  
Sankt Augustin and Bonn, Germany  
david.biesner@iais.fraunhofer.de

2nd Maren Pielka  
Fraunhofer IAIS  
Sankt Augustin, Germany

3rd Rajkumar Ramamurthy  
Fraunhofer IAIS and University of Bonn  
Sankt Augustin and Bonn, Germany

4th Tim Dilmaghani  
PWC GmbH WPG  
Frankfurt a. M., Germany

5th Bernd Kliem  
PWC GmbH WPG  
Frankfurt a. M., Germany

6th Rüdiger Loitz  
PWC GmbH WPG  
Frankfurt a. M., Germany

7th Rafet Sifa  
Fraunhofer IAIS  
Sankt Augustin, Germany

Abstract—Natural language processing methods have several applications in automated auditing, including document or passage classification, information retrieval, and question answering. However, training such models requires a large amount of annotated data which is scarce in industrial settings. At the same time, techniques like zero-shot and unsupervised learning allow for application of models pre-trained using general domain data to unseen domains.

In this work, we study the efficiency of unsupervised text matching using Sentence-Bert, a transformer-based model, by applying it to the semantic similarity of financial passages. Experimental results show that this model is robust to documents from in- and out-of-domain data.

Index Terms—NLP, Transfer Learning, BERT, Text Classification

I. INTRODUCTION

The auditing process of financial reports involves proof-reading and correcting yearly financial statements by trained auditors. The auditors present an opinion on whether the report issued by the audited company is prepared according to all legal requirements of the applicable financial reporting framework such as International Financial Reports Standards (IFRS) and Handelsgesetzbuch (HGB). This process requires expert knowledge, judgment and experience by the auditors.

However, a large part of the described process proves itself recurrent, manual and time-consuming. For example, consider the evaluation of a financial report to the guidelines given by a regulatory framework. The guidelines are generally presented as a series of checklist requirements that must be addressed in the report. For each such checklist item, the designated auditor has to first find the relevant section in the document and then use their expert judgement to answer the compliance question raised by the checklist item. Finding pertinent sections of a large document can be time-consuming since many items do not lend themselves to a simple lookup function and relevant passages can be spread throughout the text. For regulatory frameworks with a couple hundred to over a thousand requirements and report lengths of up to several hundred pages, the simple process of finding relevant text passages can take up a large portion of an auditor’s time.

One possible approach to addressing this issue is using machine learning algorithms for text recommendation. That is, given a specific requirement text, the goal of the prediction model is to recommend one or more relevant text passages, ideally as a list ranked by relevance to the desired requirement. Previous work [1, 2] has formulated this problem as a text classification problem, in which the model determines the relevance of a text passage from a financial report to a specific requirement in the checklist. First, the text passage is encoded to a latent representation (using TF-IDF [1] or transformer-based language models [2]). Then this representation is used by a classification model (a multi-layer perceptron or a recurrent neural network) to assign relevance scores to each requirement from the checklist. Once relevance scores for all pairs of passage-requirement are computed, the results can be ranked by relevance for each requirement.

The detriment of this process is two-fold: on the one hand, training such a model requires annotated data, i.e., financial reports with text paragraphs annotated as relevant for specific requirements or a list of relevant text passages from various reports for each requirement. Curating this data is time-consuming and therefore expensive. On the other hand, these models deal with a fixed set of possible requirements from a checklist. Once a classification layer or a group of requirement classifiers is trained, adding new requirements to predict is only possible by retraining of the partial or the entire model. However, legal frameworks change over time and it is therefore possible that a model trained on data from a particular year might not be usable in successive years.

Additionally, these models ignore a critical aspect of the auditing process: requirements are not only target labels for the classification model, they are written text and therefore contain information that might be useful for a model to understand the task better. A human auditor does not memorize which text passages are relevant for which checklist entry number but rather processes the requirement text and tries to understand which text passages might match. Therefore, a model that considers text from a textual passage as well as the requirement to compute relevance scores is desired.
To alleviate these issues, we propose a recommendation approach that can be trained entirely unsupervised, without annotated data, and can recommend target requirements without ever seeing them during training. Based on textual semantic similarity, our proposed model encodes both the requirement and paragraph text to latent representations and calculates the cosine similarity between the two vectors to get a relevance score. Semantically close texts are assigned latent vectors in similar regions of latent space with high cosine similarity, texts that are not related are assigned latent vectors with low similarity. This method can be used entirely without domain-specific training. A language model pre-trained on general language data will assign latent vectors with the latent space similarity property described above to any text and can be used to rank text paragraphs for relevance out of the box. Further training of the model to better encode and match domain-specific texts can be done both in a supervised and unsupervised manner.

Such a model is then capable of matching texts to entirely new requirements it has never seen during training, simply by following the steps above and encoding the new requirement text.

To alleviate these issues, we propose a recommendation approach that can be trained entirely unsupervised, without annotated data, and can recommend target requirements without ever seeing them during training. Based on textual semantic similarity, our proposed model encodes both the requirement and paragraph text to latent representations and calculates the cosine similarity between the two vectors to get a relevance score. Semantically close texts are assigned latent vectors in similar regions of latent space with high cosine similarity, texts that are not related are assigned latent vectors with low similarity. This method can be used entirely without domain-specific training. A language model pre-trained on general language data will assign latent vectors with the latent space similarity property described above to any text and can be used to rank text paragraphs for relevance out of the box. Further training of the model to better encode and match domain-specific texts can be done both in a supervised and unsupervised manner.

Such a model is then capable of matching texts to entirely new requirements it has never seen during training, simply by following the steps above and encoding the new requirement text.

To alleviate these issues, we propose a recommendation approach that can be trained entirely unsupervised, without annotated data, and can recommend target requirements without ever seeing them during training. Based on textual semantic similarity, our proposed model encodes both the requirement and paragraph text to latent representations and calculates the cosine similarity between the two vectors to get a relevance score. Semantically close texts are assigned latent vectors in similar regions of latent space with high cosine similarity, texts that are not related are assigned latent vectors with low similarity. This method can be used entirely without domain-specific training. A language model pre-trained on general language data will assign latent vectors with the latent space similarity property described above to any text and can be used to rank text paragraphs for relevance out of the box. Further training of the model to better encode and match domain-specific texts can be done both in a supervised and unsupervised manner.

Such a model is then capable of matching texts to entirely new requirements it has never seen during training, simply by following the steps above and encoding the new requirement text.
In this section, we briefly describe the data we used in our experiments. See Table I for an overview of all datasets and training, validation and test splits.

For unsupervised training, we provide two datasets of German and English language. The German dataset consists of financial reports from BundesAnzeiger¹ (BANZ). The English dataset consists of financial reports from the US Securities and Exchange Commission² (SEC). We only use these datasets for training, not evaluation and therefore do not split them. See Table I for a reference on number of paragraphs and word counts.

For supervised training, we annotate two datasets of financial reports in German and English language. Those data sets were provided to us by an auditing firm, who collaborated with us for this project. We annotate each paragraph as, if applicable, corresponding to one or more individual checklist items from the IFRS regulatory checklist with a total of 1305 items. See Figure 2 for sample checklist items with corresponding description text. We keep only annotated paragraphs in the dataset, see Table I for an overview of number of paragraphs and word count in each split. We additionally describe the number of individual requirements that are part of some annotation in the corresponding split.

For model testing, we create two test splits. One test set stems from the same distribution as the training set. This test set contains no report texts that are also in the training set, but the requirement annotations are for the same requirements as the annotations in the training set. That way, the model has already seen report texts that match the requirements in question. We call these test sets test seen (see Table I).

The other type of test set contains new report texts and requirement annotations that the model has not seen during training, i.e. no report text in the training dataset has any annotation of these requirements. To predict reasonable text matches, the model must understand the requirement texts itself and match the fitting texts correctly. A binary or multilabel classification model trained to predict the requirements in the training dataset cannot predict any of these requirements.

V. TRAINING AND TRANSFER LEARNING

The proposed method allows for various types of supervised and unsupervised learning. We therefore train the model in multiple steps on various datasets and evaluate the performance after each training stage.

A. Language Model Training

The underlying BERT architecture of the SentenceBERT model itself is a language model, meaning it is trained to reconstruct un-annotated text input and learn from datasets of raw text. For details on the training of a BERT language model we refer to [3].

| Dataset | Split | Paragraphs | Words | Requirements |
|---------|-------|------------|-------|--------------|
| BANZ    | -     | 70 203     | 1 144 226 | -             |
| SEC     | -     | 70 977     | 1 274 397 | -             |
| IFRS_DE | train | 20 522     | 1 343 917 | 758           |
|         | val   | 2281       | 160 180  | 761           |
|         | test seen | 2818    | 175 747  | 758           |
|         | test unseen | 177     | 12 593   | 8             |
| IFRS_EN | train | 12 486     | 813 854  | 660           |
|         | val   | 1744       | 119 562  | 491           |
|         | test seen | 1869    | 140 768  | 660           |
|         | test unseen | 272     | 21 333   | 7             |

¹https://www.bundesanzeiger.de/
²https://www.sec.gov/dera/data/financial-statement-and-notes-data-set.html
BERT is available pretrained on a variety of datasets. We consider the model bert-base-multilingual-cased\(^3\), which is trained on Wikipedia text from 104 languages. The model should be able to process any type of input text in a reasonable manner to provide a baseline for further training. Results for models based on this language model are shown in Table IIa.

Additionally, we finetune the bert-base-multilingual-cased language model on a dataset of financial language data (BANZ and SEC, see Table I). We theorize that a language model trained on domain specific data will improve matching performance after further training steps. We show results for model based on this language model in Table IIIb.

**B. Unsupervised Training**

The second training step consists of training the sentence embeddings in an unsupervised manner. During this training stage, we do not differentiate between report texts and requirements and consider both as individual input texts. We consider two training mechanisms: Simple Contrastive Learning of Sentence Embeddings (SimCSE [11]) and Transformer-based Denoising AutoEncoder (TSDAE [12]).

During the SimCSE training, we aim to improve the sentence embeddings by enforcing that each embedding occupies its own region in latent space, meaning that embeddings for two different input sentences are far apart in latent space. This has shown to significantly improve the performance of sentence embedding models in downstream tasks [11]. For training the model via SimCSE, we embed all sentences and requirements in a batch of documents and calculate the cosine similarity between each pair of texts. We add some noise using dropout in the encoding process, such that two embeddings of the same text are not exactly alike. We train the model to maximize the cosine similarity between two embeddings of the same text and minimize the cosine similarity between embeddings from different texts. For details on this training we refer to [11].

The second unsupervised training approach, TSDAE, is similar to the language modeling training described in Section V-A. For this, the model is adjusted into an encoder-decoder-structure. The encoder is given by the SentenceBERT model, initialized either from the general language (bert-base-multilingual-cased) or the financial data model. During training, noise is added to each input text by masking a fraction of all tokens (we mask 60% in our experiments), encoding the noisy text and reconstructing the embedding using the decoder module. After training, the decoder module is discarded and the encoder is used for prediction or further finetuning. For details on this training we refer to [12].

We train the model using both methods on two un-annotated datasets of German (BANZ in Table I) and English (SEC) language until convergence, and select the model with the best matching score on the validation set for further training or evaluation.

\(^3\)https://huggingface.co/bert-base-multilingual-cased

**C. Supervised Training**

During the last training step, we aim to utilize the annotated datasets, i.e. the datasets of financial reports in German and English with report paragraphs annotated as matching a certain requirement text. The training procedure is based on the same SimCSE method described in Secion V-B. In this case, we do not want to only enforce two noisy encodings of the same text to be close in latent space, but the encodings of matching report text and requirement pairs to be close, i.e. have a large cosine similarity. In the same manner, we enforce non-matching text and requirement to have encodings with small cosine similarity. For this, we encode all report texts and requirements from a batch of annotated data and calculate the cosine similarity between each report text and each requirement text. Unlike the unsupervised training, we do not calculate the cosine similarity between two encodings of reports texts or two encodings of requirement texts. We train the model to maximize the cosine similarity of matching text and requirement pairs and minimize the cosine similarity of all other pairs. For further details on the supervised training we again refer to [11].

We train the model on German, English or German and English annotated reports (DE, EN and DE+EN in Table I, respectively) until convergence and select the model with the best matching score on the validation set for evaluation.

**VI. EVALUATION AND RESULTS**

We evaluate all trained models on two types of hold-out test sets, the test set with seen requirements and the test set with unseen requirements, as described in Section IV.

We evaluate each type of test set in German, English and combined German and English language (DE, EN and DE+EN in Table I, respectively).

We collect all evaluation results in Tables IIa and IIb. For each training step, we compute the one-shot recall on the top-5 recommendations given by the model, which is a custom performance measure that has been designed to reflect user experience. For each annotated report text, we let the model predict the matching score for all requirements and select the 5 highest rated requirements as prediction. We assign a score of 1.0 for this sample if a correct requirement is part of this prediction and average the score over the dataset. A 90% one-shot recall signifies that in 90% of all cases, a correct requirement was part of the top-5 predictions.

Comparing both subtables IIa and IIb, we see that pretraining the language model on financial data significantly improves classification for two of the three unseen requirement datasets, while the best metrics for seen requirement dataset are very similar, increasing or decreasing by less than 1 percentage point depending on the dataset. Also note that the same training procedures lead to similar results for both language models, i.e. TSDAE unsupervised training and DE supervised training performs best on the DE seen test set and TSDAE unsupervised training and DE+EN supervised training best on both other seen test sets. Although this effect is less present when skipping the supervised training and not occurring
when skipping both unsupervised and supervised training, we conclude a net benefit from pretraining the language model on domain specific data.

For consecutive evaluations we only consider the model pretrained on financial language data in Table IIb.

In order to evaluate the effect of both unsupervised training methods, SimCSE and TSDAE, we first note that only unsupervised training results in poor performance on the test sets, with a maximum of 17.81% one-shot recall on a seen test set. We therefore concentrate on the effect of the unsupervised training method when continuing training in a supervised manner, see the bottom half of Table IIb.

We see that the best TSDAE model outperforms the best SimCSE model by a significant margin on all seen test sets (improvement of 3 to 5 percentage points) and two of the three unseen test sets (improvement of 7 to 10 percentage points). For the remaining unseen test set the metric scores of the best models are very similar (46.32% for TSDAE and 47.45% for SimCSE). As a result of this evaluation, we conclude that TSDAE is the more fitting unsupervised training method.

Finally, we consider the effect of supervised training data language on the test performance. In this evaluation, we see no significant trend towards any specific data language setup. While the model trained on German language only performs best on the German seen test set, the model trained on English language only does not outperform the other data setups in any test set. But training on any English language data is clearly necessary for the seen test sets in English or combined English and German. Peculiarly, the model trained only on German data performs best on the English only unseen test set, with a significant margin against the next best model.

We conclude that, if available, training on data from both languages is recommended if the use case requires processing of texts in both languages. However, more testing of this hypothesis is required.

In general, the evaluation shows that the proposed model architecture performs very well on the task of predicting requirements available in training data (over 90% one-shot recall on all seen datasets). Additionally, the methods produce reasonable results when faced with new requirements, which
a multilabel prediction model could not process at all.

In practice, such a model can be applied to replace a multilabel prediction model with little to no loss in performance. When faced with a new or adjusted set of requirements, this model would not need to be retrained entirely but could be applied directly to the new task and improved by supervised training when training data is available.

VII. CONCLUSION AND OUTLOOK

In this study, we analyzed the potential of sentence embedding methods for text matching based recommendation models in the domain of financial reports. We trained a SentenceBERT model in a multi-stage training approach to embed sentences from financial reports and the text from corresponding regulatory checklist items and used the model to recommend labels that have been seen during training and labels that have never been seen during training.

We found that the proposed method performs competitively on seen labels and reasonably well on unseen labels. This suggests these models to be useful in a setting were a subset of the labels or the entire label set might change between retrainings of a model.

Open points for future research include:

- The application of the models in languages other than German and English: How does the model perform on other languages it has never seen in training? Does additional language model training in the target language improve performance?
- Combining the pretraining methods TSDAE and SimCSE: Both are unsupervised, but training a model using both methods either simultaneously or consecutively is possible and might improve performance.
- Splitting the model at some point during the training stages into an encoder model for report texts and an encoder model for requirement texts. Having dedicated models for both might increase matching performance due to further domain specialization.

We look forward to deploying the models developed during the course of this study in existing smart auditing software and expect a resulting optimization of machine learning workflows.

VIII. ACKNOWLEDGEMENTS

In parts, this research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence, LAMARR22B.

REFERENCES

[1] R. Sifa, A. Ladi, et al., “Towards automated auditing with machine learning,” in Proceedings of the ACM Symposium on Document Engineering 2019, DocEng ’19, (New York, NY, USA), Association for Computing Machinery, 2019.
[2] R. Ramamurthy, M. Pielka, et al., “Alibert: Improved automated list inspection (ali) with bert,” in Proceedings of the 21st ACM Symposium on Document Engineering, DocEng ’21, (New York, NY, USA), Association for Computing Machinery, 2021.
[3] J. Devlin, M.-W. Chang, et al., “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019.
[4] L. Hillebrand, T. Deußer, et al., “KPI-BERT: A joint named entity recognition and relation extraction model for financial reports,” in Proc. ICPR, 2022.
[5] Y. Cao, H. Li, et al., “Towards automatic numerical cross-checking: Extracting formulas from text,” in Proc. WWW, 2018.
[6] D. Biesner, R. Ramamurthy, et al., “Anonymization of german financial documents using neural network-based language models with contextual word representations,” Springer International Journal of Data Science and Analytics, 2021.
[7] M. Tsai and C. Wang, “On the risk prediction and analysis of soft information in finance reports,” European Journal of Operational Research, 2017.
[8] J. Campbell, H. Chen, et al., “The information content of mandatory risk factor disclosures in corporate filings,” Review of Accounting Studies, 2014.
[9] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 11 2019.
[10] A. Vaswani, N. Shazeer, et al., “Attention is all you need,” 2017.
[11] T. Gao et al., “Simcse: Simple contrastive learning of sentence embeddings,” 2021.
[12] K. Wang et al., “Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning,” 2021.
[13] N. Reimers and I. Gurevych, “Making monolingual sentence embeddings multilingual using knowledge distillation,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 11 2020.
[14] C. L. Chapman, L. Hillebrand, et al., “Towards generating financial reports from tabular data using transformers,” in Machine Learning and Knowledge Extraction, (Cham), pp. 221–232, Springer International Publishing, 2022.