Detecting Omissions of Risk Factors in Company Annual Reports

Corentin Masson [1*] , **Syrielle Montariol**[1,2*]
[1] Université Paris-Saclay, CNRS, LIMSI, 91400, Orsay, France.
[2] Société Générale, Paris, France
corentin.masson@limsi.fr, syrielle.montariol@limsi.fr

Abstract

Regulators require most companies to publish yearly reports, describing their activities, results, future plans, and risk factors. Sometimes a risk factor can be omitted in a document, possibly –voluntarily or not– misleading the readers. In this paper, we introduce a task for detecting omitted risk factors in Annual Reports. This new task requires to catch the risks mentions in multiple sentences, and to identify the ones that are specific to a sector or a period. To address it, we use a neural architecture to extract risk sentences from documents and cluster the risk factors from these sentences. Finally, we generate synthetic risk factor omissions and propose a metric to evaluate the omission detection method.

1 Introduction

Risk analysis is a popular task in Business and Management research. While usually approached through expert knowledge and quantitative inputs [Kaplan and Garrick, 1981], it can benefit from the use of unstructured data such as legal and regulatory documents. One of the associated tasks is the automatic extraction of risk sentences.

Theoretically, a risk can be defined as a hazard with a potential for damage to an entity. Its meaning differs from the notion of uncertainty; in the former, one is able to quantify precisely the probability of occurrence and its potential impacts [Altham, 1983]. Therefore, a risk can be defined as a triplet composed of the potential event characterized as a risk, its quantitative counterparts such as the probability of occurrence, and its possible consequences [Kaplan and Garrick, 1981]. Thus, risk evocations can be identified by a topic-oriented summarization system able to detect occurrences of these triplets from natural language written documents such as Annual Reports (ARs).

Listed companies are regulated by the Financial Market in which their value is most traded in, often inducing the obligation to regularly publish information documents. ARs are supposed to exhaustively describe a company's current well-being, perspectives and the risks it is facing. In France, nearly 190 ARs are released each year from CAC40, CAC60 and CAC90 indexes (the principal French stock indexes from Euronext.).

To the best of our knowledge, few authors tackle risk sentences extraction from non-HTML indexed ARs [Liu *et al.*, 2018]; they often rely on XBRL[1]-indexed 10-K filings to identify risk factors markers [Huang and Li, 2011]. However, automatic analysis of such raw long documents can be beneficial for the Financial and Regulatory sectors. These documents represent the vast majority of ARs disclosed worldwide and are composed of an average of 3500 sentences with various sections and topics [AMF, 2020]. As for now, little has been done on extracting specific sections from Annual Reports or indexing them. In this paper, we focus on extracting and analysing the risk factors from these ARs.

In France, the financial market is regulated by the Financial Market Authority (AMF). In particular, disclosure of ARs depends on the "Code Monétaire et Financier" and on the "Doctrine" [2]. Companies must release every year a report containing all the requested information. If an element that might be important for a potential investor is missing from an AR, the company runs the risk of being accused of voluntarily omitting information, which is a specific kind of fraud.

From the extracted risk sentences, it is therefore possible to identify the possible omission of a risk in an AR by comparing its risk distribution to other ARs from the same sector and year. Therefore, in this paper, (1) we propose a new task for omitted risk factors detection from the DoRe Corpus [Masson and Paroubek, 2020], composed of European Companies ARs; and (2) we present a resolution method based on Neural Risk Sentences Extraction and Unsupervised Risk Factors Clustering. We hope to gather people to make the task grow.[3]

2 Related Works

The literature on corporate ARs analysis is plentiful in the financial research community. However, from the NLP perspective, research is more scarce and much more recent, while offering a wide range of applications from stock markets volatility prediction [Kogan *et al.*, 2009] to fraud detection. Today, financial reporting for companies faces a con-

[*] The authors contributed equally to this research.

[1] https://www.xbrl.org/the-standard/what/ixbrl/

[2] AMF guidance for righteous behavior on the market.

[3] Please contact us by email for access to the corpus.

tradiction: the huge increase in volume leads to more and more need of solution from the NLP community to analyse this unstructured data automatically. However, more reporting from more companies leads to more diversity in the shape of the documents; this lack of standardization and structure makes the analysis tougher and requires more complex methods [Lewis and Young, 2019].

For investors and regulators, risk sections are important parts of ARs, as they contain information about the risks faced by the companies and how they handle it. [Mandi *et al.*, 2018] extract risk sentences from legal documents using Naive Bayes and Support Vector Machine on paragraph embeddings. [Dasgupta *et al.*, 2016] explore project management reports from companies to extract and map risk sentences between causes and consequences, using hand-crafted features and multiple Machine Learning methods. [Ekmekci *et al.*, 2019] performed a multi-document extractive summarization on a news corpus for a risk mining task. As it has not yet been done, we experiment extractive summarization on risk extraction task in ARs.

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information and overall meaning. In recent years, approaches to tackle this difficult and well-known NLP problem make use of increasingly complex algorithms ranging from dictionary-based approaches to Deep Learning techniques [Xiao and Carenini, 2019]. The current research trend deviates from general summarization to topic-oriented summarization [Krishna and Srinivasan, 2018], targeting a specific subject in the document such as risks in ARs in our case.

Focusing on detecting risk factors in ARs, topic modeling has been extensively used for this task in the literature [Zhu *et al.*, 2016; Chen *et al.*, 2017]. The evaluation is mostly done using intrinsic measures and by looking at the topics manually. Only [Huang and Li, 2011] manually define 25 risk factor categories, relying on ARs from the Securities Exchange Commission.

3 Pipeline

We propose a pipeline including a Risk Sentence Extractor module with Active Learning labeling framework and a Topics Modeling module to identify omitted risk factors.

3.1 Risk Sentences Extraction

As presented in Figure 1, each sentence in the document is processed sequentially using a fine-tuned French version of BERT [Devlin *et al.*, 2019] named Flaubert [Le *et al.*, 2020]. The goal is to compute the probability for each sentence to be a risk sentence using three modules: a Sentence Encoder, a Document Encoder and a Sentence Classifier.

Data Description

ARs are often disclosed in PDF format, which requires a lot of pre-processing (a notable exception are the 10-K filings [Kogan *et al.*, 2009]). ARs are extremely long documents: they contain an average of 3500 sentences and 27 different sub-sections. Due to the large size of each document, completely labeling a set of reports would take a considerable amount of time. To handle this, we propose to split the document into a set of disjoint sub-documents and label by hand a randomly selected subset of these sub-documents.

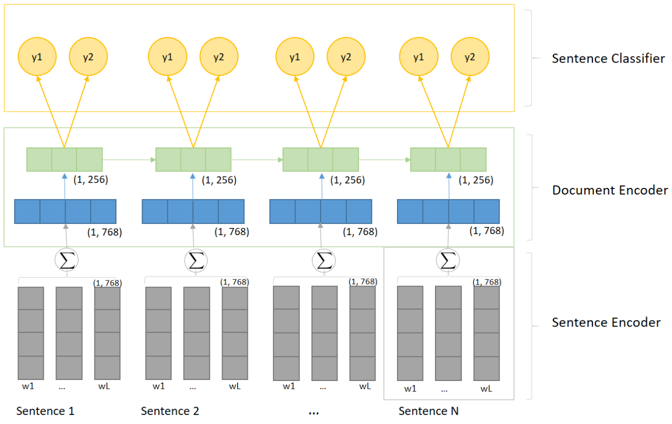

Figure 1: Risk Sentences Extraction architecture overview.

Model Architecture

The first module is a Sentence Encoder; its goal is to embed each sentence into a k-dimensional space without the information from the surrounding sentences. Due to the limited amount of labeled data, we use a FlauBERT pre-trained Language Model and fine-tune it for the extraction task, allowing it to get a good approximation of basic syntax and semantic features in higher layers [Jawahar *et al.*, 2019]. With N_D being the number of sentences in a document $D = (S_1, S_2, ..., S_{N_D})$ and M_i being the length of the sentence $S_i = (w_1, w_2, ..., w_{M_i})$, $SentEnc_i$ is the sum of the token embeddings computed by the fine-tuned FlauBert:

$$SentEnc_i = \sum_{j=1}^{M_i} \text{BERT}_{TokenEmb_j}(S_i)$$

We also experiment with a version where the sentence embeddings $SentEnc_i$ are computed using the `[CLS]` token from the FlauBert model. In both cases, each sentence is mapped into a v dimensional vector.

Risk evocations are often split into multiple sentences. For example, in Figure 2, the first sentence displays the risk factor while the second depicts the uncertainty with *'if'* and *'might'* along with the potential impact (*'affect its market share in a near future'*).

The sector is driven by innovation from newcomers. If the Group does not keep with the process, it might affect its market share in a near future.

Figure 2: Example of risk evocation.

We want our model to be able to extract all parts of the risk evocation. In order to extract sentence embedding taking into account the surrounding sentences (context sentences), we apply a forward LSTM layer at the document level, each sentence being considered as a token whose embedding comes from the Sentence Encoder. We take the hidden state of each sentence as the context sentence embedding.

$$DocEnc_i = LSTM(SentEnc_1, SentEnc_2, ..., SentEnc_{M_i})$$

As decoder, we add one linear layer with dropout for regularization. Its input comes directly from the contextualized sentence embeddings computed through the Document Encoder module, followed by a softmax layer to compute probabilities.

$$P(y_i = 1) = Softmax(Linear(DocEnc_1, ..., DocEnc_{N_D}))$$

For training, our loss function is a L2-penalized binary cross-entropy loss.

$$\mathcal{L} = -\sum_{d=1}^{N}\sum_{i=1}^{N_d}(y_i log(p_i) + (1 - y_i)log(1 - p_i)) + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

Active learning

To our knowledge, there is no freely available dataset for risk sentences extraction in French nor in English, leaving us with a considerable labeling task. Randomly selecting sub-documents to label would be biased toward non-risk sentences and therefore would make the dataset asymmetric. Thus, we implement a Pool-Based Query-By-Committee [Settles, 2010] Active Learning approach using dropout masks for committee models generation and compute stochastic predictions for each sentence [Tsymbalov *et al.*, 2018]. It allows to select the most informative sub-documents to label and increase the accuracy of the model for these sentences which are near the segmentation frontier.

With $L = \{D_1^L, D_2^L, ..., D_{N_L}^L\}$ the set of labeled sub-documents and $U = \{D_1^U, D_2^U, ..., D_{N_U}^U\}$ the set of unlabeled sub-documents, the framework – or Learner, as called in the Active Literature – looks for x^*, the most informative sentence with the selected query strategy. Our committee $H = \{h_1, h_2, ..., h_T\}$ is composed of T models. At each Active Learning iteration, a model is trained on the already labeled data. Then, T different dropout masks are applied on the classification layer of the Sentence Classifier module in order to generate T different model. They are used to compute stochastic predictions for each sentence in each sub-document.

Using the predictions for each sentence, we can compute the uncertainty score. As the Least Confidence, Sample Margin and Entropy measures are equivalent in the binary case, we compute the approximated Least Confidence measure using votes from the committee H for probability estimation p_i for each sentence. The uncertainty measure of a given sub-document is the average uncertainty score of all its sentences.

$$LS(D) = \frac{1}{N_D^U}\sum_{i=1}^{N_D^U}|p_i - 0.5|$$

where $p_i = P(y_i = 1|X_i)$

The learner ranks sub-documents by decreasing uncertainty measure and queries the M most informative sentences to the Oracle following : $x^* = \arg\max_{D^U} LS(D^U)$. The process is then iterated until a stop criterion is met, such as an insufficient increase of accuracy between two iterations.

3.2 Risk Omission Detection

We use the set of risk sentences extracted from the ARs to detect if a risk factor was omitted in a document.

Motivation & Pipeline

All companies describe different types of risks in their ARs, often through a “risk factors” section. To detect if an AR is missing a risk factor that should have been reported, we would need to define a list of risks factors for all the companies. However, the regulators do not enforce any normalisation nor provide a list of risks to report. Thus, the number and the type or risks reported vary a lot in the different documents. Consequently, we have to use unsupervised methods to capture them.

From the sets of risk sentences, we create a mapping of the risks depending on the sector and the year of the ARs. The distribution of risks per year can also allow to identify emerging risks, while the distribution per sector allows to identify the risks that are specific to a sector. We can either work on the data at the sentence level using sentence clustering or at the document level by doing topic modeling. We present the two approaches in the following section.

Sentences clustering

We cluster the risk sentences of all documents together to identify the types of risks across the full corpus. We use the sentence representations from the risk sentence extraction step using FlauBERT.

Moreover, we can assume that successive sentences, or sentences that are close in the document, have a high probability to deal with the same risk factor. Thus, the surrounding sentences as well as their distance to the target sentence can add valuable information to the clustering. We use the representation of the surrounding sentences as features for the clustering, by doing element-wise sum with the representation of the main sentence, weighted by a factor of their distance to the main sentence. The distance is computed according to the number of sentences: two successive sentences have a distance $d = 1$, etc. Then, the weight of each sentence is computed as the inverse of its distance to the main sentence augmented by one: $w = \frac{1}{d+1}$.

For the clustering, we use the K-means algorithm. The number of clusters k is chosen according to the literature on risk factors in ARs. To ease the interpretation of the different clusters of risk sentences, we use a method to detect keywords in the clusters. We consider each cluster of sentences as a document and the set of clusters as a corpus. To identify the most representative words in a cluster, we compute the tf-idf (Term Frequency - Inverse Document Frequency) score of each word in the clusters. We exclude stopwords and words that can be found in 50% of the clusters or more. The words with the highest score in each cluster are used to label it.

Topic Model on Documents

We challenge the previous method using a popular topic modeling algorithm: the Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003]. Each document is characterised by a probability distribution over a set of topics, while each topic is characterised by a probability distribution over all the words of the vocabulary. Therefore, the top words per topic are used as a set of keywords to describe it. The number of topics is the same as the number of clusters for the sentence clustering with K-Means.

Intrinsic Evaluation Measures

We compute several measures, all relying on a list of keywords characterising each topic or cluster.

First, the Normalized Point-wise Mutual Information (NPMI) [Aletras and Stevenson, 2013] measures the topic coherence. It relies on word co-ocurrences to measure the level of relatedness of the top k words characterizing each topic. We also use external knowledge – pre-trained Word2Vec embeddings [Mikolov *et al.*, 2013] [4] – to evaluate topic coherence. Similarly to [Ding *et al.*, 2018], we compute the pairwise cosine similarity between the vectors of the top k words characterizing each topic, and average it for all topics. We call this second topic coherence measure TC-W2V. For the two measures, we use a relatively low k ($k = 10$). A high NPMI or TC-W2V measure indicates an interpretable model.

These two measures are completed by a topic uniqueness (TU) measure [Nan *et al.*, 2019] for the top k keywords, representing the diversity of the topics. For a given topic t, with $cnt(i)$ being the number of times the word i appears in the top words of all the topics, the TU is computed as:

$$TU_t = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{cnt(i)}$$

We take the global TU measure as the average TU for all topics. The higher the TU measure is (close to 1), the higher the variety of topics. We use $k = 25$ for this measure.

Risk Omission Detection Task

The extrinsic evaluation is done using the detection of omissions as downstream task. We want to detect if a company omitted or under-reported a risk in one of its reports, by observing the risks reported in the document, and comparing it with the ones reported in other documents of the same year and the same sector.

First, we generate synthetic risk omissions in our corpus. We randomly sample a small set of ARs, manually select a section of each document describing one type of risk, and remove it. Our goal is double: to detect that a risk factor is missing in the altered document, and to identify the risk associated with the removed section.

To tackle this problem, we compute a measure relying on a binarized version of the topic distribution of a document. Indeed, both the topic model and the sentence clustering methods output a distribution of risks (respectively topics or cluster) for each document. We consider that a document includes a topic (or a cluster) if the proportion of the topic (or the number of sentences belonging to the cluster) is higher than a threshold ϵ. Below this threshold, we consider that the document does not report the risk characterised by that topic. Then, for each sector and for each year, we extract the set of "typical" topics: the ones that are present in most documents for that sector or year, and therefore are expected to appear in all documents of the same sector and year.

First, we count the number of documents mentioning each risk. Then, we binarize it: if the number of documents mentioning the risk is lower than half of the total number of documents in the sector/year, then the risk is considered as not important for the sector/year and we do not select it. We compare this list of "expected" topics with the list of topics reported in each document. It allows to identify the documents where a risk is absent but should have been reported, because it is a risk common to most documents for that sector or year.

For the second step, we check whether the missing topic detected by our method is the same as the one removed from the selected document. We use the fitted LDA and the fitted K-Means algorithm to predict the topics (the clusters) which can be found in the set of sentences that were removed from the selected documents. If there is at least one topic in common between the set of "missing" topics in the document, and the set of topics predicted from the removed sections, we consider that the omission has been correctly detected.

In order to evaluate the ability of our methods to tackle the task, we define the accuracy measure as the proportion of correctly detected omissions among the 20 altered documents. This measure can be computed by using the documents of the same sector or of the same year as comparison; we name it *Binary-sector* and *Binary-year* accuracies. We also compute a joint measure, taking into account both the expected topics from the year and the ones from the sector: *Binary-all*.

4 Experiment

4.1 Data Preparation

Preparation for Risk Extraction

For labeling, we selected a random subset of 50 ARs from the whole DoRe Corpus containing French and Belgian companies with large, mid and small capitalization from various sectors. These documents are converted from PDF to TXT format using MuPDF [5], some were unusable and excluded after conversion, such as the 2018 AR from AIR LIQUIDE. We then extracted start and end offset of sentences from these documents using Stanza[6] from StanfordNLP team; we chose it for its accuracy and relative speed. All of these preprocessing steps induce errors; that is why we add some custom rules to filter out unusable sentences based on number of letters / sentence length ratios and counts of line-breaks in a sentence. To handle the cold start of our Active Learning approach, we label up to 1000 sentences in successive groups of 5 from the 4 first documents in the random sample. The labeling rule is to label a sentence as Risk sentence if it includes the notion of uncertainty, and if at least one other element from the Risk triplet is present. We take into account the surrounding sentences to check whether the missing element

[4] We use pre-trained French word embeddings on the Wikipedia Corpus: http://fauconnier.github.io

[5] https://mupdf.com/

[6] https://stanfordnlp.github.io/stanza/

	Accuracy	F1	Recall
Iteration 1	0.8412	0.7373	0.7236
Iteration 2	0.8002	0.6403	0.6863
Iteration 3	0.8331	0.7483	0.6771
Iteration 4	0.8721	0.7767	0.8034
Iteration 5	0.8845	0.8158	0.7723
Iteration 6	**0.8969**	**0.8269**	**0.8216**

Table 1: Performance measures for each active learning iteration.

	Accuracy	F1	Precision	Recall
BERT CLS	0.8398	0.7679	**0.8968**	0.6715
BERT Sum	**0.8969**	**0.8269**	0.8323	**0.7723**

Table 2: Final results of both models after the final Active Learning iteration.

from the triplet is present in a sentence around the current one; if it is the case, we also label this second one as risk.

The initial set of 200 sub-documents is composed of groups of 5 successive sentences. We apply zero-padding to those with less than 5 sentences. We are unable to label a set of risk sentences representative of all potential risk topics from different sectors due to the dimensionality of the data; to evaluate the ability of the algorithm to detect risks even outside the sectors it has seen previously, we split the dataset into two parts and put sub-documents from two of the four first labeled ARs into the test set. This test set containing 70 sub-documents is used to follow the evolution of the performance metrics at each Active Learning iteration. It also allows the metrics during the Active Learning to be less sensitive to randomness of the split due to the low amount of data.

Active Learning

From these selected data, we train the first model in our Active Learning pipeline. The parameters for our Query-By-Committee approach are the dropout probability of classification layers weights set to $p = 0.5$ and the number of models in the committee H set to $T = 15$ for computation feasibility.

We iterate 6 times and have 39% of risk sentences in the labeled sample. We can see in Table 1 that the metrics globally increase during iterations while it is still subject to instability due to the lack of data. A solution to stabilize the results could be to add a cross-validation step, but it is computationally expensive.

Preprocessing for risk clustering

We focus on the CAC40 companies. We have 388 annual reports from 40 companies, spanning 12 sectors and 12 years (from 2008 to 2019). From the risk sentences extraction step, we have for each document, a set of risk-related sentences and their position in the document. On average, the extracted risk-related sentences correspond to 3.6% of the full document (minimum proportion = 1.3%, maximum = 14.1%). Each document is associated with a year and a company, which belongs to one of the 12 sectors. For both the topic modeling and the sentence clustering methods, the number of topics can be chosen by relying on the literature. Following [Huang and Li, 2011], we use $k = 25$ topics.

We apply a heavy processing step to all the risk sentences, in order to get a document as clean as possible to extract the most important keywords for each topic more efficiently. From the set of risk sentences, we first clean all errors resulting from the transition from pdf to text (divided words, merged characters...). Then, we exclude the sentences that have less than 60% of letters (too many symbols, spaces or digits in a sentence usually means that a portion of a data table was extracted). We delete numbers and symbols from the remaining sentences. We also remove French stopwords, words of less than 2 characters, words found in less than 15 documents and words found in more than 80% of the documents. Finally, we lemmatize all the words. [7]

4.2 Results

Risk Sentence Classification

We train two models for risk sentences classification, differing in the method to compute non-contextualized sentence embeddings. The first one (BERT Sum) is computed from the sum of the hidden-states of the last attention layer from the fine-tuned FlauBert model. The second model (BERT CLS) uses the CLS token, even though the Extractive Summarization literature tends to conclude that the second attempt is less accurate [Xiao and Carenini, 2019]. Regarding the architecture, we set the Document Encoder LSTM hidden-states to 256, the Classifier Linear layer dropout probability to 0.5, the L2 penalization parameter of the loss function to 0.01 and the learning rate to 1.e$\hat{-}$5. The model is optimized by Adam-Optimizer for 150 epochs with batch size of 16. We keep as best model the one having the best validation accuracy, and test it on the previously created test set (not used during Active Learning nor training).

Table 2 presents the final results of both models after the last Active Learning iteration. Even if the (BERT CLS) Precision is better (0.8968), the increase in the recall (+0.1008) for (BERT Sum) makes it the best model for the task with the current amount of data. Table 1 shows the results of the Active Learning step, increasing the F1 score by 0.0785 (10% increase in only 5 iterations). We believe that with a greater amount of data, the model can still increase its performance and gain a better capacity to identify unknown risk factors.

For each document, the risk sentences extracted by the model from each sub-document are concatenated to create the topic-oriented summary.

Risk Clustering

In order to identify the different risk factors from the topic-oriented summary, we use the unsupervised methods described in section 3.2.

On the one hand, we apply Online LDA [Hoffman *et al.*, 2010][8] to the set of risk sentences after preprocessing. On

[7] For lemmatization, we use the `LefffLemmatizer()` from `Spacy`: https://pypi.org/project/spacy-lefff/

[8] Using `Gensim` implementation: https://radimrehurek.com/gensim/models/ldamulticore.html

	NPMI (k=10)	TC-W2V (k=10)	TU (k=25)
LDA	**-0.153**	0.175	**0.691**
KM	-0.240	**0.186**	0.652

Table 3: Intrinsic measures of topic modeling and sentence clustering quality.

the other hand, we apply K-Means to the set of sentence embeddings extracted from the Sentence Encoder. We experiment with K-Means of sentences embeddings (KM), Augmented K-Means using weighted embeddings of surrounding sentences with window = 2 (KM2), and Augmented K-Means with window = 4 (KM4). As a preliminary measure of quality, we compute the silhouette score of the K-Means clusterings. The score is the highest for the Augmented K-Means with a window of 4 sentence (score = 0.178), slightly lower with a window of 2 sentences (score = 0.162), and even lower for the standard K-means (score = 0.147).

From the LDA, we have a set of keywords describing each topic. Some topic examples along with an interpretation of the associated risk factor are presented in Table 5. To be able to compare it with the sentence clustering, we extract keywords from the sentence clusters from the K-Means algorithm, using the aforementioned tf-idf method (section 3.2. Then, we compute the three intrinsic measures for both LDA and K-Means to evaluate the quality of the topic model and the clustering (Table 3). The measures for the Augmented K-Means are almost the same as for the standard K-Means.

The measures show that the sentence clustering method leads to a higher extrinsic topic coherence (TC-W2V) than the topic model, but lower intrinsic topic coherence (NPMI). Moreover, the TU measure is lower for K-Means, meaning that the clusters are less diversified.

Risk Omission Detection

We use the same models for the risk omission detection task. In order to generate synthetic omissions in ARs, we randomly sample and alter 20 ARs of the CAC40 companies, by manually removing a section describing one risk factor; and we add these altered documents to our corpus. We choose risk sections of different sizes, describing different types of risks; for example, we remove the *System security and cyber attack* section in the 2018 AR from ATOS, and the *Risk of delay and error in product deployment* section in the 2017 report from DASSAULT SYSTEMES.

After fitting the LDA and the K-Means on the corpus, we obtain the distribution of risks in the altered documents and the average distribution of risks for each sector and year. According to the method described in section 3.2, we binarize these vector and compare them in order to identify the list of missing topics in the altered documents. Then, using the topic model and clustering fitted on the full corpus, we predict the distribution of risks in the sections that were removed from the selected documents. Finally, we can compute the accuracy measures described in section 3.2 using the LDA, the standard K-Means and the Augmented K-Means with windows of size 2 and 4 (Table 4).

	LDA	KM	KM2	KM4
Binary - sector	0.2	0.7	**0.8**	**0.8**
Binary - year	0.2	**0.55**	0.4	0.4
Binary - all	0.4	0.75	**0.8**	**0.8**

Table 4: Accuracy measures for the risk omission detection task on the manually altered documents.

Risk factor	Example of keywords
reputation	agency, advertiser, publicity, affect, negatively
patent	property, intellectual, licence, brand, software
energy	oil, exploration, hydrocarbon, well, damage

Table 5: Translation of keywords examples using LDA with 25 topics, and manually associated risk factor.

Augmenting the K-Means algorithm by using the surrounding sentences, even though it improved the silhouette score, does not lead to a clear improvement for this task. However, the LDA leads to much lower accuracy compared to the K-Means algorithm. It might be linked with the low extrinsic topic coherence of the LDA compared to K-Means.

5 Conclusion

In this paper, we introduced the task of risk omission detection and proposed a pipeline to tackle it. First, we extract risk sentences from company annual reports using an Encoder-Classifier architecture on top of contextualised embeddings from the BERT model. Then, we use unsupervised methods to extract the risk distribution of each annual report.

We generate synthetic risk factor omissions in a sample of ARs in a straightforward way, propose a method to detect them, and a metric to evaluate the method. We conclude that a sentence-level analysis, by clustering sentence representation extracted with BERT, is more adapted than LDA to address the task. Augmenting the sentence clustering by using a weighted sum of the representations of the surroundings of a sentence can further increase its quality. The low performance of the LDA might be overcame using more advanced topic modelling methods [Nan *et al.*, 2019], possibly relying on word embeddings [Dieng *et al.*, 2019].

However, the risk sentence extraction step could be improved with more Active Learning iterations, for the model to learn more about the notions of uncertainty and the impacts than about the risk factors that has already been observed during training. It could also be improved by increasing the number of sentences in each sub-document and transferring information between consecutive sub-documents in an AR.

Acknowledgments

We address our deepest thanks to François Hu (ENSAE-CREST, Société Générale) for his valuable advices on Active Learning, to Patrick Paroubeck (LIMSI-CNRS) and Alexandre Allauzen (ESPCI, University Paris-Dauphine) for their support and comments on the paper.

References

[Aletras and Stevenson, 2013] Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *IWCS 2013*, pages 13–22, Potsdam, Germany, March 2013. ACL.

[Altham, 1983] J. E. J. Altham. Ethics of risk. *Proceedings of the Aristotelian Society*, 84:15–29, 1983.

[AMF, 2020] AMF. Annual report regulation, February 2020. AMF indications on information submission.

[Blei *et al.*, 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[Chen *et al.*, 2017] Yu Chen, Md Rabbani, Aparna Gupta, and Mohammed Zaki. Comparative text analytics via topic modeling in banking. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 11 2017.

[Dasgupta *et al.*, 2016] Tirthankar Dasgupta, Lipika Dey, Prasenjit Dey, and Rupsa Saha. A framework for mining enterprise risk and risk factors from news documents. In *COLING 2016*, pages 180–184, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL 2019*, pages 4171–4186, Minneapolis, Minnesota, June 2019. ACL.

[Dieng *et al.*, 2019] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. Topic modeling in embedding spaces. 2019.

[Ding *et al.*, 2018] Ran Ding, Ramesh Nallapati, and Bing Xiang. Coherence-aware neural topic modeling. In *EMNLP 2018*, pages 830–836, Brussels, Belgium, October-November 2018. ACL.

[Ekmekci *et al.*, 2019] Berk Ekmekci, Eleanor Hagerman, and Blake Howald. Specificity-based sentence ordering for multi-document extractive risk summarization, 2019.

[Hoffman *et al.*, 2010] Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *NeurIPS 2010*, page 856–864, Red Hook, NY, USA, 2010. Curran Associates Inc.

[Huang and Li, 2011] Ke-Wei Huang and Zhuolun Li. A multilabel text classification algorithm for labeling risk factors in sec form 10-k. *ACM Trans. Management Inf. Syst.*, 2:18, 10 2011.

[Jawahar *et al.*, 2019] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *ACL 2019*, pages 3651–3657, Florence, Italy, 2019. ACL.

[Kaplan and Garrick, 1981] Stanley Kaplan and B. John Garrick. On the quantitative definition of risk. *Risk Analysis*, 1(1):11–27, 1981.

[Kogan *et al.*, 2009] S. Kogan, D. Levin, B.R. Routledge, J.S. Sagi, and N.A. Smith. Predicting risk from financial reports with regression. ACL, 2009.

[Krishna and Srinivasan, 2018] Kundan Krishna and Balaji Vasan Srinivasan. Generating topic-oriented summaries using neural attention. In *the NAACL 2018*, pages 1697–1705, New Orleans, Louisiana, June 2018. ACL.

[Le *et al.*, 2020] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. Flaubert: Unsupervised language model pre-training for french. In *LREC*. ACL, 2020.

[Lewis and Young, 2019] Craig Lewis and Steven Young. Fad or future? automated analysis of financial text and its implications for corporate reporting. *Accounting and Business Research*, 49(5):587–615, 2019.

[Liu *et al.*, 2018] Yu-Wen Liu, Liang-Chih Liu, Chuan-Ju Wang, and Ming-Feng Tsai. RiskFinder: A sentence-level risk detector for financial reports. In *NAACL 2018*, pages 81–85, New Orleans, Louisiana, June 2018. ACL.

[Mandi *et al.*, 2018] Jayanta Mandi, Dipankar Chakrabarti, Neelam Patodia, Udayan Bhattacharya, and Indranil Mitra. Use of artificial intelligence to analyse risk in legal documents for a better decision support. In *TENCON 2018*, Jeju, Korea (South), 10 2018.

[Masson and Paroubek, 2020] Corentin Masson and Patrick Paroubek. Nlp analytics in finance with dore: A french 250m tokens corpus of corporate annual reports. In *LREC 2020*, pages 2254–2260, Marseille, France, May 2020. European Language Resources Association.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. pages 3111–3119. NIPS, 2013.

[Nan *et al.*, 2019] Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. Topic modeling with Wasserstein autoencoders. In *ACL 2019*, pages 6345–6381, Florence, Italy, July 2019. ACL.

[Settles, 2010] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin, Madison, July 2010.

[Tsymbalov *et al.*, 2018] Evgenii Tsymbalov, Maxim Panov, and Alexander Shapeev. Dropout-based active learning for regression. pages 247–258, Cham, 2018. Springer.

[Xiao and Carenini, 2019] Wen Xiao and Giuseppe Carenini. Extractive summarization of long documents by combining global and local context. In *EMNLP-IJCNLP 2019*, pages 3011–3021, Hong Kong, China, 2019. ACL.

[Zhu *et al.*, 2016] Xiaodi Zhu, Steve Yang, and Somayeh Moazeni. Firm risk identification through topic analysis of textual financial disclosures. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 12 2016.