Automated ESG report analysis by joint entity and relation extraction*

Adrien Ehrhardt[1,2][0000−0002−4448−3644] and Minh Tuan Nguyen[1,2]

[1] Groupe de Recherche Opérationnelle, Groupe Crédit Agricole, Montrouge, France
[2] École Polytechnique, Saclay, France
adrien.ehrhardt@credit-agricole-sa.fr

Abstract. The banking industry has lately been under pressure, notably from regulators and NGOs, to report various Environmental, Societal and Governance (ESG) metrics (*e.g.*, the carbon footprint of loans). For years at Crédit Agricole, a specialized division examined ESG and Corporate Social Responsibility (CSR) reports to ensure, *e.g.*, the bank's commitment to de-fund coal activities, and companies with social or environmental issues. With both an intensification of the aforementioned exterior pressure, and of the number of companies making such reports publicly available, the tedious process of going through each report has become unsustainable.
In this work, we present two adaptations of previously published models for joint entity and relation extraction. We train them on a private dataset consisting in ESG and CSR reports annotated internally at Crédit Agricole. We show that we are able to effectively detect entities such as coal activities and environmental or social issues, as well as relations between these entities, thus enabling the financial industry to quickly grasp the creditworthiness of clients and prospects w.r.t. ESG criteria.[3]

Keywords: Named Entity Recognition · Relation extraction · NLP.

1 Introduction

By deciding which projects and companies to fund, Corporate and Investment banks have many responsibilities, of which environmental concerns are among the latest, and stir passion.

For example, regulatory authorities (see *e.g.* [6]) and NGOs (see *e.g.* [14]) regularly push the industry towards more transparency in that regard.

For years at Crédit Agricole, a specialized division examined, among others, approx. 4,000 ESG and CSR reports each year from clients and prospects to ensure, *e.g.*, the bank's commitment to de-fund coal activities, and companies

* Supported by Groupe Crédit Agricole; analyses and opinions of the authors expressed in this work are their own. The authors wish to thank the ESG team at CACIB for the document annotations and their valuable comments.

[3] The resulting model is provided at https://github.com/adimajo/renard_joint

with social or environmental issues[4]. With an intensification of the aforementioned exterior pressure, signs of intricate relations between ESG metrics and creditworthiness (see *e.g.* [3]), the rise of "sustainable banking"[5], and of the number of companies making such reports publicly available, the tedious process of going through each report has become unsustainable.

The ClimLL dataset Following this situation, a random sample (not only current clients) of 31 ESG and CSR reports were annotated internally at Crédit Agricole. These include for example ArcelorMittal[6] and PSA[7]. We refer to this (private) dataset as ClimLL. Using this dataset, our aim is to derive a joint entity and relation extraction model able to replicate this manual annotation on unseen reports so as to accelerate their reading by analysts. Three annotators worked on these reports with 100 % overlap at first. Once they reliably reached over 80 % interrater reliability, this overlap was progressively reduced.

Data model A crucial part of any NLP project is the data model: deciding which concepts to label as entities and relations, such that each have a precise definition, and every annotator can unambiguously annotate the dataset. The data model for ClimLL is displayed on Figure 1: there are 8 entity types, among which "Coal Activity", "Environmental" and "Social Issues", as well as 5 relation types. This data model is likely to evolve rapidly in the coming years as the ESG metrics of interest will evolve. However, we feel it is a strong basis to build on.

Descriptive statistics More than 7,500 sentences from 372 paragraphs are split into a training and a test set containing 280 and 92 paragraphs respectively. All paragraphs from a given report belong to the same split. Sentence lengths range from 2 to 251 words (Figure 2 - left). In total, there are 28,751 entities and 5,864 relations (Figure 2 - center and right resp.). A sample sentence extracted from the dataset is displayed on Figure 3.

Proprietary model Using proprietary tools developed by IBM, a joint entity and relation extraction was "trained", which will serve as a baseline (see Section 3.4). This model was also used to determine that annotating these 31 reports was sufficient, as the F1 score on the test set stopped improving (see Appendix A).

[4] Some of these reports are becoming mandatory, *e.g.* in France as part of the "document d'enregistrement universel" required by the regulating authority, and audited.

[5] The incorporation of ESG criteria alongside traditional financial metrics; see *e.g.* https://www.unepfi.org/banking/bankingprinciples/, https://www.ca-cib.com/our-solutions/sustainable-banking

[6] Available at https://corporate.arcelormittal.com/corporate-library

[7] Available at https://www.groupe-psa.com/en/newsroom/corporate-en/groupe-psa-publishes-its-csr-report/

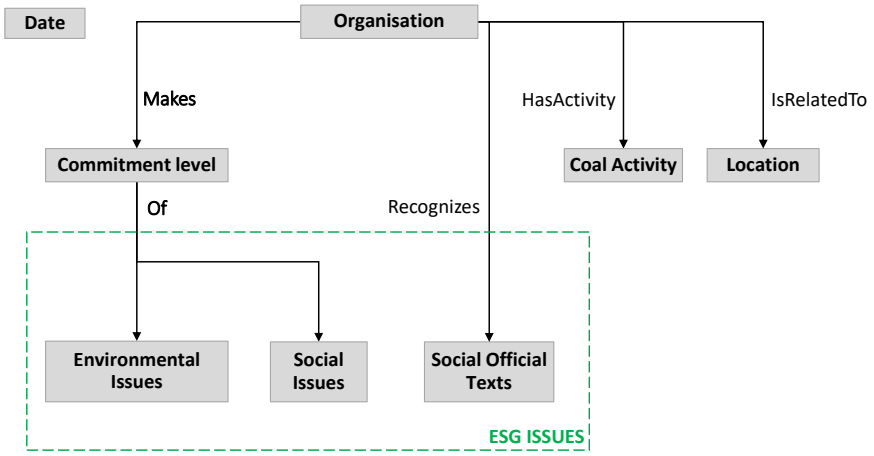

Fig. 1. The data model used to produce ClimLL (entities in boxes, relations as arrows).

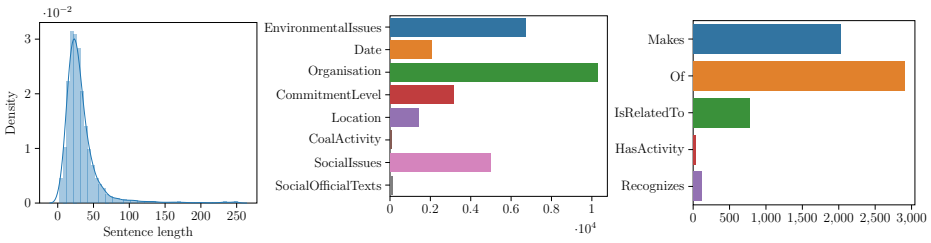

Fig. 2. Distribution of sentence length (left), entities (center) and relations (right).

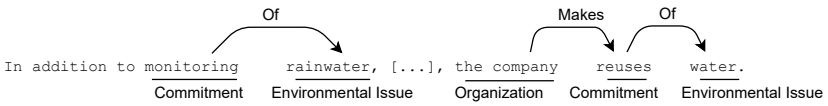

Fig. 3. A sample sentence from ClimLL.

In the next section, we present, adapt and implement two published works to solve this joint entity and relation extraction task. Section 3 is devoted to numerical experiments. We conclude this work in Section 4.

2 Related work: adaptation and application

The first challenge in Natural Language Processing (NLP) is to transform raw text into a meaningful numerical representation. This will be tackled in Section 2.1. Then, the problem of identifying entities and relations can be naturally decomposed into Named Entity Recognition (Section 2.2) and Relation Extraction (Section 2.3). Both problems will finally be tackled simultaneously in Section 2.4.

2.1 Representation algorithms

Tokenization An NLP task, including entity and relation extraction, usually starts with tokenization: transforming an arbitrary input text into a list of tokens from a fixed set, called vocabulary, which, in turn, can be transformed into embedded numerical vectors for computation purposes. During the tokenization process, a word can be broken down into multiple tokens (*e.g.*, "rainwater" $\longrightarrow$ ["rain", "##water"]) or transformed (e.g. to lowercase, lemmatization, stemming, etc).

Transformers In order to solve different NLP tasks, among which Named Entity Recognition and Relation Extraction, different models were designed, in particular neural networks, often based on recurrent neural networks (RNN) or Long Short-Term Memory (LSTM) [8], and trained on specific datasets. Such architectures usually take a long time to train because of two drawbacks: (1) they cannot process tokens in parallel and (2) they were designed and trained for each use-case, *i.e.* for each dataset, from scratch.

To overcome the former drawback, the Transformer architecture was proposed [18]. It is a neural network with an encoder and a decoder. Each encoder layer contains a multi-headed self-attention and a feed-forward sub-layer. Each decoder layer contains a masked multi-headed self-attention, a multi-headed attention, and a feed-forward sub-layer. An attention is a map from a query vector Q and a key-value vector pair K (of dimension d_k), V to a weighted sum of the the components of the value V. The weights are given by a function of the query and the key as in Equation (1). The self-attention layer corresponds to the special case where the query Q and the key K are the same: this removes the need for a recurrent neural network and enables parallel computation. Furthermore, the self-attention layer offers shorter paths between long-distance dependencies in text.

$$\text{Attention}(Q, K, V) := \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V. \tag{1}$$

BERT To overcome the latter drawback of RNNs (retraining on each dataset), the Bidirectional Encoder Representations from Transformer (BERT) model was proposed [4] and trained on the Masked Language Modeling (MLM) [17] and the Next Sentence Prediction (NSP) tasks. During training, BERT takes two tokenized sentences as input, where some tokens are masked, along with "special" tokens (`[CLS]` at the beginning and `[SEP]` at the end of each sentence, see Figure 4). Through MLM, the model learns to predict the masked tokens and, through NSP, the representation of the `[CLS]` token is used to predict whether two input sentences are consecutive in the original document. Hence, through these two tasks, the model is able to learn the context surrounding each token and across sentences instead of only one direction.

Fig. 4. An example of BERT input in MLM.

Subsequently, by taking the result of its last hidden layer, BERT is used as a representation algorithm: an input sentence is transformed into the consecutive 768-dimensional numerical representation of its tokens. Simple models are then used after this transformation to solve "downstream" NLP tasks [4].

2.2 Named Entity Recognition (NER)

One of these downstream tasks is NER: classifying words into pre-defined classes; for example the "Organization" class is said to "span" over the two words "the company" in Figure 3.

The approaches range from grammar-based or vocabulary-based to machine learning (see *e.g.* [15,12,4]). In particular, the numerical representation of the first token of each word, given by BERT, can be used as an input to a simple classifier, *e.g.* logistic regression [4]: we will focus on this approach in what follows.

First, the input sentence is tokenized into token ids using the tokenizer provided by BERT, *e.g.*, "rainwater" $\longrightarrow$ ["rain", "##water"] $\longrightarrow$ [4458, 4669]).

For entity recognition, the BERT representation of the first token id of each word serves as an observation w.r.t. the aforementioned simple classifier, while the entity of this word ("Environmental Issue") is the class label, straightforwardly providing a design matrix and allowing the estimation of this classifier. Figure 5 (left) provides an illustration of the tokenization and NER processes. The different formats for providing entity labels are discussed in Appendix B.

2.3 Relation Extraction (RE)

Following the same approach for RE, we implemented the one-pass Multiple Relation Extraction (MRE, see Appendix C) model [21], where the tokens' BERT

representation and the true entities serve as input. The average BERT representation of all tokens of each entity span is computed. Each pair of "averaged" entities is then classified into a relation or marked as "not a relation" using a softmax (Figure 5 - right).

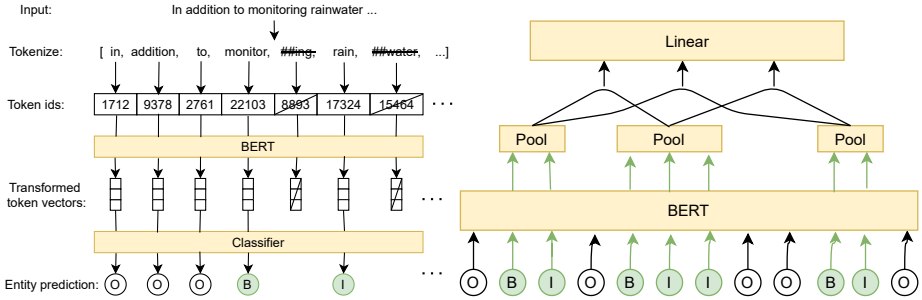

Fig. 5. Tokenization and NER illustration (left), MRE model [21] (right).

2.4 Joint Entity and Relation Extraction (Joint NER & RE)

NER-RE pipeline To combine both previous tasks, the "natural" idea is to use a pipeline where the entities are identified first before extracting the relations. Thus, the predicted entities from the NER model described in Section 2.2 are given as input to the RE model described in Section 2.3 in place of the true entities.

Consequently, this two-stage procedure is vulnerable to compound error and does not consider the connection between entities and relations.

Joint extraction Indeed, entities are sometimes defined by the types of relations they are in, as exemplified by these two article titles from CNN: "Bloomberg buys BusinessWeek" and "Bloomberg will sell his company if elected, adviser says". Based on the relations mentioned in the titles, the former "Bloomberg" is a company while the latter refers to a person. Hence, to overcome the disadvantages of the NER-RE pipeline, there have been various studies on alternative methods for joint entity and relation extraction. The Hierarchical Framework [16] consists of two levels of reinforcement learning. The higher level traverses the text to extract and classify relations while the lower level identifies entities within each extracted relation. The Encoder-Decoder architecture [13] implements multiple bidirectional LSTM. However, both methods only detect whether or not a token is part of an entity and do not classify the entity type. Other approaches which are able to extract and classify both entities and relations are, for example, a reinforcement learning multi-turn question answering model [9] or DYGIE [11]. Nevertheless, these methods depend on heavily-engineered architectures and require a lot of resources to train.

Recently, the popularity of BERT-based methods for joint entity and relation extraction has risen. DYGIE++ [20] and Span-based joint entity and relation extraction (SpERT) [5] both make use of BERT. These methods can extract multiple relations from the input sentence while SpERT is span-based and can thus also extract overlapping entities (see Appendix B). It achieved high performance on public joint entity and relation extraction datasets. We focus on the latter in what follows.

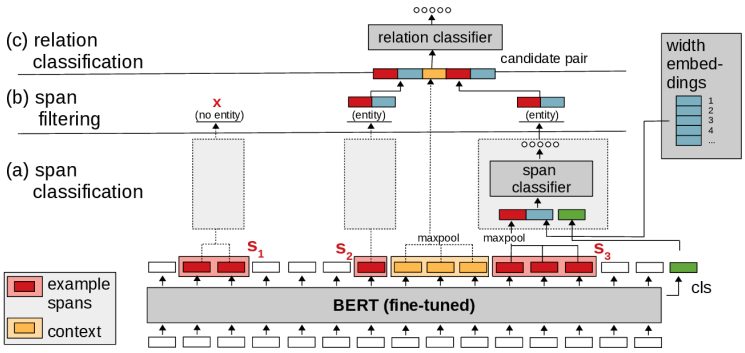

Fig. 6. The SpERT architecture [5], shared under CC BY-NC 4.0.

SpERT First, similar to the NER-RE pipeline described above, the input sentence is tokenized and passed through BERT. Next, token representations from "candidate" entity spans (see below) are max-pooled (the maximum value per coordinate), concatenated with a trainable width-embedding vector and the representation of the `[CLS]` token. A softmax layer classifies the concatenated vectors into entity classes or non-entity. Spans classified as non-entity are filtered out. Then, pairs of classified entity spans are once again max-pooled and concatenated with the width-embedding vector for relation classification. The representations of the words between the two entities are also max-pooled and concatenated, which act as a context. The relation classifier is a sigmoid layer and a filter threshold, such that, potentially, multiple relations can be predicted per pair of predicted entities. The whole process is displayed on Figure 6.

Candidates During the training process, undersampling is performed such that candidate entity spans and relations consist in the true ones and a fixed number of negative spans generated randomly. On the other hand, during prediction and evaluation, candidate entities are all spans within the sentence and candidate relations are all pairs of predicted entity spans. In order to reduce complexity, the length of the spans is limited to 10 words.

It is worth noting that even though entity spans are spans on lists of tokens, and words may be tokenized into several tokens, the candidate entity spans must contain all tokens of each word.

3 Numerical experiments

First, we experiment with the entity recognition (see Section 2.2) and relation extraction (see Section 2.3) models of the NER-RE pipeline separately in Section 3.2 and 3.3 respectively. Then, we combine them, as suggested in Section 2.4, and compare the performance with our implementation of SpERT (we describe in Appendix D how it differs from the original implementation) on the public datasets CoNLL04 and SciERC (described in the next section), as well as our proprietary dataset ClimLL.

3.1 Public datasets

CoNLL04 contains 1,400+ sentences with annotated entities and relations extracted from news sources to represent daily life language. There are 4 classes of entities: Location, Person, Organization and Other, which cannot overlap. Each sentence may contain multiple relations; however, there is no cross-sentence relation. There are 5 types of relations: Located_In, Work_For, Live_In, OrgBased_In, and Kill. We divide the dataset into the same training, development, and test subsets as in the existing literature [2].

SciERC focuses on scientific language extracted from paper abstracts. It contains 500 annotated paragraphs with 2,500+ sentences. There can be multiple relations within a sentence but not across sentences. The entities are span-based, which means different entities can overlap. There are 6 entity classes (Task, Method, Metric, Material, Generic, OtherScientificTerm) and 7 relation classes (Conjunction, Feature-of, Hyponym-of, Used-for, Part-of, Compare, Evaluate-for). We use the same training, development and test subsets provided by the dataset authors [10].

3.2 Named Entity Recognition

In this section, the NER model (Figure 5 - left) is trained on the ClimLL dataset. We used 5 different classifiers: k-nearest neighbor (5 neighbors), decision tree (unlimited depth), random forest (20 trees with unlimited depth), and two neural networks with 1 hidden layer of size 512 and 1024 respectively.

Furthermore, BERT has an input size limit of 512 tokens and was pre-trained mostly on inputs of size less than 128. Since ClimLL offers input paragraphs which are typically tokenized into more than 512 tokens, we experimented with two separation methods to breakdown these paragraphs: (1) split the paragraphs sentence by sentence and pass them to BERT one by one or (2) break the input paragraphs down into chunks of 128 tokens and pass to BERT.

Results are displayed in Table 1. The top-performing classifiers are the k-nearest neighbor and the two neural networks. Even though random forest achieved the highest precision, it has a much lower recall and, thus, a lower F1 score compared to the others. Furthermore, we can see that the entity recognition model produced similar results compared to the IBM model.

Table 1. NER results on ClimLL (test set, micro-average).

	(1) Sentence-by-sentence			(2) 128-by-128		
Classifier	Precision	Recall	F1	Precision	Recall	F1
k-nearest neighbor	0.79	**0.80**	**0.79**	0.79	**0.80**	**0.79**
Decision tree	0.42	0.46	0.44	0.43	0.47	0.45
Random forest	**0.94**	0.39	0.52	**0.93**	0.40	0.53
Neural network (512)	0.80	0.75	0.77	0.84	0.74	0.78
Neural network (1024)	0.81	0.76	**0.79**	0.83	0.75	0.78
IBM	0.87	0.70	0.78			

It is rather surprising that the 128-by-128 separation achieves similar performance compared to the sentence-by-sentence approach because these 128-token chunks are cutting through the sentences. This result led us to believe that neighboring sentences may carry useful information to detect entities in a given sentence. In what follows, we will use sentence-by-sentence separation.

Also, the results of this section are already overwhelming: using now standard and open-source NLP tools such as BERT and very simple supervised classification models thereafter, we are able to detect the concepts of interest, *e.g.*, coal activities, which could already prove useful to the financial industry.

3.3 Multiple Relation Extraction

For relation extraction, we trained the MRE model [21] (Figure 5 - right) on CoNLL04 and ClimLL, where sentences and true entities are passed as input. The model was trained for 100 epochs on each dataset. We did not fine-tune any other hyper-parameters. The micro-averaged and macro-averaged results are shown in Table 2. They are inferior to state-of-the-art [5] (0.74 macro F1 with SpERT[8]), due to limitations pointed out in Section 2.4.

3.4 Joint Entity and Relation Extraction

After experimenting with the NER and RE tasks, we combined the NER model with a neural network (1 hidden layer of size 1024) classifier and the MRE model to create a NER-RE pipeline for joint entity and relation extraction as described in Section 2.4. We benchmarked this pipeline and SpERT on CoNLL04

[8] https://paperswithcode.com/sota/relation-extraction-on-conll04

Table 2. MRE results on test set.

Dataset	Average	Precision	Recall	F1
ClimLL	Micro	0.61	0.54	0.57
	Macro	0.55	0.54	0.54
CoNLL04	Micro	0.65	0.58	0.61
	Macro	0.66	0.61	0.63

and ClimLL. On SciERC, we could only test SpERT because the dataset contains overlapping entities. For SpERT, we chose a learning rate of $5e^{-6}$ on ClimLL and $5e^{-5}$ on the other two public datasets. The rest of the hyper-parameters are the same as suggested by the authors [5]. Some training metrics are made available in Appendix E.

During the evaluation, an entity is considered correct if its span (the begin and end position) and its predicted type match its true value. A relation is considered correct if both of its entities (spans and types) together with the predicted relation type are all correct.

The evaluation results are presented in Table 3. Overall, SpERT outperformed the NER-RE pipeline as predicted. The performance of SpERT on public datasets matches the results shown in the original paper[8].

Table 3. Joint entity and relation extraction results on test set.

			NER			Joint NER & RE		
Dataset	Average	Model	Precision	Recall	F1	Precision	Recall	F1
CoNLL04	Micro	NER-RE	0.75	0.80	0.77	0.36	0.47	0.41
		SpERT	**0.86**	**0.91**	**0.89**	**0.71**	**0.70**	**0.70**
	Macro	NER-RE	0.71	0.74	0.73	0.41	0.51	0.45
		SpERT	**0.84**	**0.88**	**0.86**	**0.72**	**0.71**	**0.71**
SciERC	Micro	SpERT	0.64	0.72	0.68	0.31	0.45	0.37
	Macro	SpERT	0.65	0.71	0.68	0.34	0.41	0.35
ClimLL	Micro	NER-RE	0.67	0.68	0.68	0.23	0.18	0.20
		SpERT	**0.75**	**0.79**	**0.77**	**0.36**	**0.44**	**0.40**
	Macro	NER-RE	0.63	0.67	0.64	0.21	0.22	0.21
		SpERT	**0.75**	**0.78**	**0.77**	**0.46**	**0.58**	**0.50**

However, we can also observe that the performance on ClimLL and SciERC cannot match the performance on CoNLL04. This may be because BERT was trained on a general language vocabulary. A much better performance was achieved on SciERC using SciBERT [5], which is a version of BERT pre-trained on scientific vocabulary. Since the ESG reports annotated in ClimLL were most likely also written with a different, more formal language, future work may improve the model performance on ClimLL by fine-tuning BERT and / or pre-training it on a specialized vocabulary.

Comparison with IBM model We also compare the performance of SpERT on ClimLL with the IBM model. In order to do this, we evaluate the entity prediction at the word-level (each word is predicted to be either non-entity or an entity type if it is part of a span of this entity type), since this is how the IBM model is evaluated. The results in Table 4 show that SpERT outperformed the IBM model (+4 % in NER and +7 % in Joint NER & RE).

Table 4. Comparison with IBM model (test set, micro-average).

	NER			Joint NER & RE		
Model	Precision	Recall	F1	Precision	Recall	F1
IBM	**0.87**	0.70	0.78	**0.52**	0.24	0.33
SpERT	0.80	**0.84**	**0.82**	0.36	**0.44**	**0.40**

4 Conclusion

With little human and computational resources, we were able to annotate a sufficiently large dataset of ESG and CSR reports and to train two open-source joint entity and relation extraction models. The SpERT model yields superior performance than the current proprietary model at Crédit Agricole and is now used daily, allowing analysts to go through more reports and concentrate on their most useful parts, participating in a broader awareness of the bank's environmental and societal role.

Both models discussed in this work, as well as code to reproduce the results on the public datasets, are publicly available at https://github.com/adimajo/renard_joint. We hope this will empower other institutions to incorporate (further) ESG criteria in their decisions.

Finally, we identified future research directions which may improve the performance of SpERT on ClimLL: incorporating the context of neighboring sentences into its input and pre-training BERT on a specialized vocabulary set, as exemplified by SciBERT.

References

1. Baldini Soares, L., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: Distributional similarity for relation learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2895–2905. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1279, https://www.aclweb.org/anthology/P19-1279
2. Bekoulis, G., Deleu, J., Demeester, T., Develder, C.: Joint entity recognition and relation extraction as a multi-head selection problem. Expert Systems with Applications **114**, 34–45 (2018)

3. Devalle, A., Fiandrino, S., Cantino, V.: The linkage between esg performance and credit ratings: A firm-level perspective analysis. International Journal of Business and Management **12**, 53 (08 2017). https://doi.org/10.5539/ijbm.v12n9p53
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Eberts, M., Ulges, A.: Span-based joint entity and relation extraction with transformer pre-training. 24th European Conference on Artificial Intelligence (2020)
6. de Guindos, L.: Shining a light on climate risks: the ecb's economy-wide climate stress test (2021), https://www.ecb.europa.eu/press/blog/date/2021/html/ecb.blog210318~3bbc68ffc5.en.html
7. Han, X., Wang, L.: A novel document-level relation extraction method based on bert and entity information. IEEE Access **8**, 96912–96919 (2020)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
9. Li, X., Yin, F., Sun, Z., Li, X., Yuan, A., Chai, D., Zhou, M., Li, J.: Entity-relation extraction as multi-turn question answering. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 1340–1350. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1129, https://www.aclweb.org/anthology/P19-1129
10. Luan, Y., He, L., Ostendorf, M., Hajishirzi, H.: Multi-task identification of entities, relations, and coreferencefor scientific knowledge graph construction. In: Proc. Conf. Empirical Methods Natural Language Process. (EMNLP) (2018)
11. Luan, Y., Wadden, D., He, L., Shah, A., Ostendorf, M., Hajishirzi, H.: A general framework for information extraction using dynamic span graphs. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 3036–3046. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1308, https://www.aclweb.org/anthology/N19-1308
12. Martins, P.H., Marinho, Z., Martins, A.F.T.: Joint learning of named entity recognition and entity linking. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. pp. 190–196. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-2026, https://www.aclweb.org/anthology/P19-2026
13. Nayak, T., Ng, H.T.: Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 8528–8535 (2020)
14. Poidatz, A.: Banques : des engagements climat à prendre au 4ème degré (2020), https://www.oxfamfrance.org/rapports/banques-des-engagements-climat-a-prendre-au-4eme-degre/
15. Straková, J., Straka, M., Hajic, J.: Neural architectures for nested NER through linearization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5326–5331. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1527, https://www.aclweb.org/anthology/P19-1527
16. Takanobu, R., Zhang, T., Liu, J., Huang, M.: A hierarchical framework for relation extraction with reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 7072–7079 (2019)

17. Taylor, W.L.: "cloze procedure": A new tool for measuring readability. Journalism quarterly **30**(4), 415–433 (1953)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
19. Verga, P., Strubell, E., McCallum, A.: Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 872–884. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-1080, https://www.aclweb.org/anthology/N18-1080
20. Wadden, D., Wennberg, U., Luan, Y., Hajishirzi, H.: Entity, relation, and event extraction with contextualized span representations. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 5784–5789. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1585, https://www.aclweb.org/anthology/D19-1585
21. Wang, H., Tan, M., Yu, M., Chang, S., Wang, D., Xu, K., Guo, X., Potdar, S.: Extracting multiple-relations in one-pass with pre-trained transformers. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 1371–1377. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1132, https://www.aclweb.org/anthology/P19-1132
22. Wang, J., Lu, W.: Two are better than one: Joint entity and relation extraction with table-sequence encoders. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1706–1721. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.133, https://www.aclweb.org/anthology/2020.emnlp-main.133

A Evolution of test F1 for the IBM model

The annotation of 372 paragraphs (see Section 1) was deemed sufficient as the F1 scores for NER and Joint NER & RE stopped improving using the IBM proprietary model, as can be seen in Figure 7.

B Named Entity Recognition representation

A popular output representation of NER is BIO (Begin, In, Out) embedding, where each word is marked as the beginning, inside, or outside of an entity (see *e.g.* [19,22]); however, this representation does not allow overlapping entities. On the other hand, span-based methods [5], which classify spans of words, can

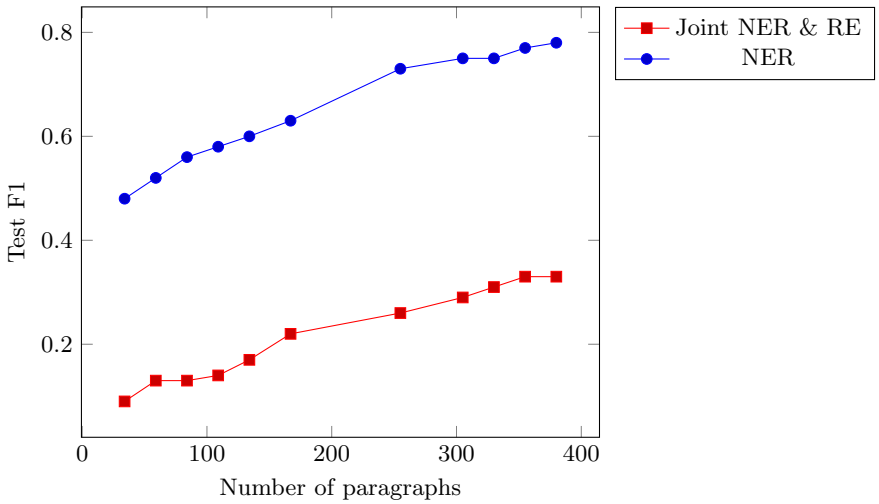

Fig. 7. Evolution of F1 scores w.r.t. the number of paragraphs.

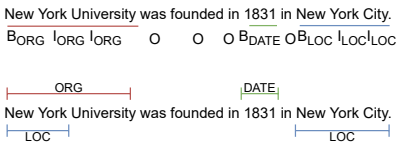

Fig. 8. Examples of BIO (above) and span-based (below) representations.

extract the spans of these overlapping entities. Figure 8 gives examples of BIO and span-based entity representations.

In the ClimLL dataset, even though the entities are presented in the span-based format in the dataset, there is no overlapping entity. Thus, it is also possible to convert to BIO format. Multiple relations can exist in the same sentence but relations cannot span across sentences. This facilitates splitting the paragraphs by sentence.

C Single versus Multiple Relation Extraction

Relation extraction algorithms are divided into two categories: Single Relation Extraction [1] (SRE) algorithms which expect only one relation per input sentence and multiple relation extraction [21,7] (MRE) where multiple relations may exist in a single input sentence (Figure 9).

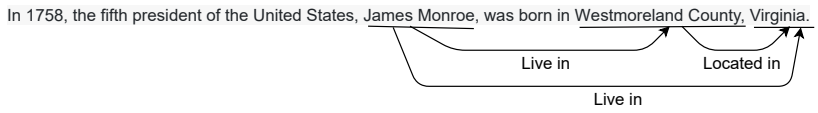

Fig. 9. Multiple relations example.

In this work, multiple relations are considered.

D SpERT

D.1 Address a shortcoming in evaluation

While re-implementing the model, we noticed that, in the evaluation process, SpERT considers an incorrectly predicted entity span or relation as two negative observations. An example is presented in Figure 10, where the model returns a set of predicted entities with "SpaceX" incorrectly classified as a person. In this case, the original evaluation process would iterate through the union of the true entity and the predicted entity sets. If an entity (including its span and type) is only presented in one of the sets, then it is considered to be classified as non-entity in the other. With this approach, "SpaceX" is considered to be incorrectly classified twice.

Thus, instead of iterating through the union of the true entity and predicted entity sets that include both entity spans and types, we only consider the union of the true entity spans with the predicted entity spans. Similarly for relations, we only take the union of the true and predicted spans of the source and target entity pair. As a result, we obtain a more accurate evaluation step.

Input SpaceX was founded by Elon Musk in 2002.

Prediction

True entities	Predicted entities
(SpaceX, Organization)	(SpaceX, Person)
(Elon Musk, Person)	(Elon Musk, Person)

Evaluation

Original

Entities	True entities	Predicted entities
SpaceX	Organization	No entity
SpaceX	No entity	Person
Elon Musk	Person	Person

Our version

Entities	True entities	Predicted entities
SpaceX	Organization	Person
Elon Musk	Person	Person

Accuracy 0.33 0.5

Fig. 10. Illustration of a better evaluation process for SpERT.

D.2 Proposed improvements

Furthermore, we also proposed two improvements to the prediction stages. Because SpERT classifies spans into entities, when dealing with datasets in BIO representation, it has to discard overlapping entities. In the original implementation, predicted entity spans are looped through in no specific order and any span that overlaps with previous spans is discarded. We suggest, instead, prioritizing discarding spans with low classification confidence.

Secondly, we noticed that the true pairs of entity types are not considered in the relation prediction stage of the original SpERT: For example, the model can only predict that an entity pair has a "live in" relation if the source entity is a person and the target entity is a location, irrespective of the probability given by the relation prediction stage. Thus, we modified the model so that it only predicts a relation if this relation fits the types of the source and target entities.

E Evolution of loss functions

The entity and relation losses as well as the F1 score on the validation set throughout the training process (30 epochs) of SpERT on ClimLL are displayed on Figure 11. Both entity and relation losses reached their minimum after only a few epochs while the validation F1 score kept improving.

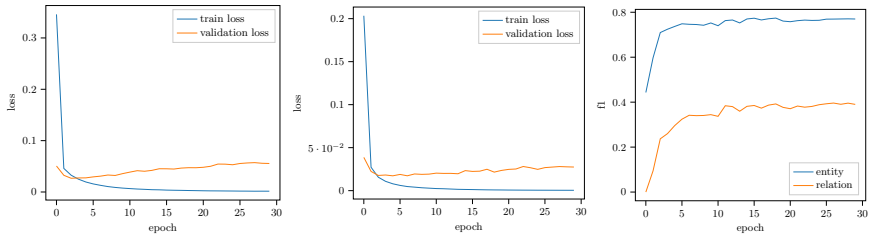

Fig. 11. Entity loss (left), relation loss (center) and F1-score on ClimLL w.r.t. training epochs (right).