Long Tailed Entity Extraction of Model Names using Distant Supervision

Swayatta Daw[1], Vikram Pudi[1]

[1]*Data Sciences and Analytics Center*
IIIT Hyderabad, India

Abstract

We introduce the task of long-tailed detection of model entities from scientific documents. We use distant supervision using an external Knowledge Base (KB) to generate synthetic training data and use a simple entity replacement technique to improve performance significantly by addressing the problem of overfitting in small sized datasets for supervised NER baselines. We introduce strong baselines for this task which are evaluated on our annotated gold standard dataset. We also release the distantly supervised silver labels generated using the KB. We introduce this model as part of a starting point for an end-to-end automated framework to extract relevant model names and link them with their respective cited papers from research documents. We believe this task will serve as an important starting point to map the research landscape in a scalable manner, needing minimal human intervention.

Keywords

Long-Tailed Entity, Entity Extraction, NER, Information Extraction, Scientific Literature,

1. Introduction

Long tailed entities are named entities which rarely occur in text documents. For these types of entities, the task of Named Entity Recognition (NER) is non-trivial. Recent approaches have aimed at solving the problem of NER using supervised training using deep learning models. However, supervised learning techniques require a large amount of token-level labelled data for NER tasks. Annotating a large number of tokens can be time-consuming, expensive and laborious. For real-life applications, the lack of labelled data has become a bottleneck on adopting deep learning models to NER tasks.

Most scientific named entities can be classified as long-tailed entities because of the rarity and domain-specificity of their occurrence. Recent work on NER in scientific documents has been concentrated around detecting biomedical named entities [1] or scientific entities like tasks, methods and datasets [2, 3, 4]. Some papers like [5] focus on the detection of a single specific entity-type (like dataset names) from scientific documents. Although previous work has focused on identifying methods [2, 3] as named entities, but what constitutes a method can have a significant variance when it comes to human annotated data. The authors [2] report the Kappa score of 76.9% for inter-annotator agreement in the SciERC dataset, which is widely used as a benchmark for scientific entity extraction.

BIR 2022: 12th International Workshop on Bibliometric-enhanced Information Retrieval at ECIR 2022, April 10, 2022, hybrid.

swayatta.daw@research.iiit.ac.in (S. Daw); vikram@iiit.ac.in (V. Pudi)

NER has traditionally been treated as a sequence labelling problem, using CRF [6] and HMM [7]. Recent approaches have used deep learning-based models [8] to address this task, which require a large amount of labelled data to train. The high cost of labelling remains the main challenge to train such models on rare long tailed entity types, where availability of labelled data is scarce. In order to address the label scarcity problem, several methods like Active Learning [9], Distant Supervision [10, 11, 12], Reinforcement Learning-based Distant Supervision [13, 14] have been proposed. [5] focused on detecting dataset mentions from scientific text and used data augmentation to overcome the label scarcity problem. In this paper, we leverage an external Knowledge Base and a large scale unlabelled corpora for our distantly supervised approach, using a simple entity replacement technique to prevent overfitting. In this paper, we introduce the task of detection of model entity names from scientific documents. Papers with Code (PwC[1]) is a community driven corpus that serves to automatically list models that solve particular subtasks, with links to the scientific research paper that introduced the model. Our aim is to build a similar but automated end-to-end pipeline that detects model names from scientific papers and benchmarks them against other similar models that solve the same task. We believe the task introduced in this paper (extraction of model names from scientific documents) to be a significant step forward towards the whole pipeline. This task is non-trivial mainly due to the lack of availability of token-level high-quality labelled data which is required for training deep learning models and the shortage of human annotated gold standard dataset for evaluation.

To address the above bottlenecks, we present a simple yet effective technique leveraging an external Knowledge Base and a large unlabelled corpora (both of which are cheap and easy to obtain) to generate our training dataset. We believe this simple technique can be easily extended to any other domain given the availability of a domain-specific Knowledge Base and unlabelled text corpora. Utilising this training set, we are able to establish a strong baseline for this task using a standard BERT-CRF model. In order to evaluate our performance for this task, we present a high quality human annotated gold standard evaluation dataset.

Using our trained models, we create an automated framework of detecting model names of related work from research papers. We define related work as prior research work done by the scientific community for the same or a similar related task that has been investigated by the original paper. Our pipeline contains two steps: Firstly, we build a sentence intent classifier that classifies whether a citation sentence contains information regarding related work or not. Then we extract model names from the positively labelled sentences using our trained NER model and link them to their respective citation mentions using a string distance based technique, introduced by [15]. We believe this framework is a starting point to effectively map the entire research landscape in a scalable manner.

2. Annotation

In order to create our whole set of gold labels for the evaluation test-set, we randomly sample abstracts from a large set of arxiv Research papers[2]. We also introduce randomly sampled papers from DBLP citation dataset to add to the diversity of train and test-set selection.

[1] https://github.com/paperswithcode/paperswithcode-data
[2] https://www.kaggle.com/Cornell-University/arxiv

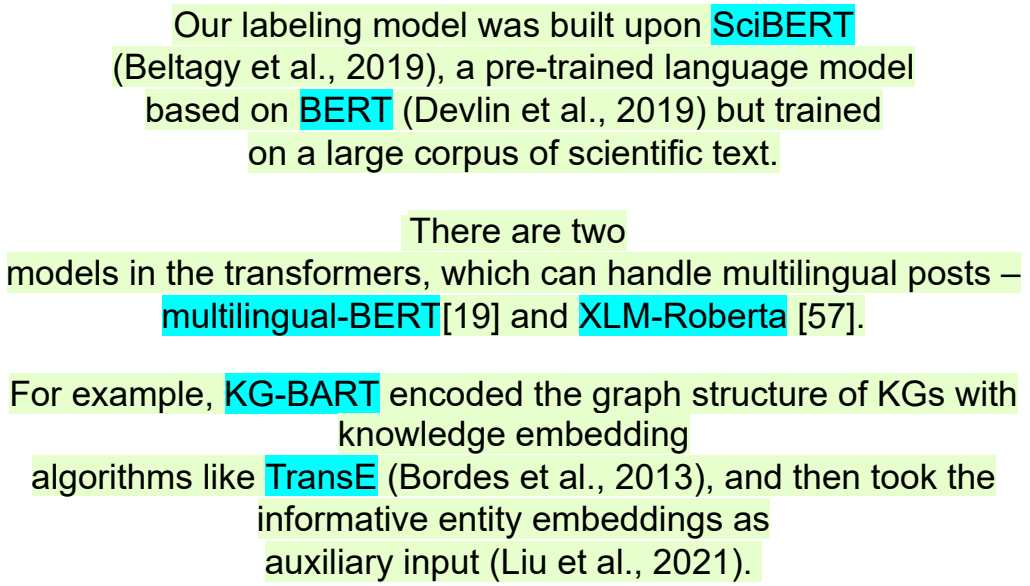

Figure 1: A few example sentences with annotated model named entities highlighted in blue. We only consider strict span matching while detection.

Considering our end goal of automating a high precision framework of extracting related model names and to minimise ambiguity, we consider only named models as model entities for this task. Few examples are - *BERT+BiLSTM+CRF, KG-BERT (with overlap)*, LSTM + Attention, DeepWalk. We consider both single named entities and a combination of multiple model named entities for annotation. A few example sentences with model entities are displayed in Figure 1.

We consider strict span matches for the model entity names, and do not consider any partial matches or synonyms. We also consider plural variants of entity names as matches.

We aim to minimise ambiguity by considering only those model named entities that we can verify about in Google Scholar and Semantic Scholar. We follow the process of identifying a candidate model name and reviewing the existing Computer Science literature to verify whether it is a model name entity or not by identifying its usage in the literature. A simple criteria that we use is to observe if the model(or a variant of the model) has been mentioned in a Results table and compared with baselines/other related models, in previous literature. Only after this thorough review, we annotate a named entity as a model name. We discard any sentence if a model named entity within the sentence does not follow the defining criteria. Hence, we believe we reduce the ambiguity sufficiently enough to allow for a single annotator for the entire annotation process. All the annotations has been done by a graduate NLP researcher who is also a co-author of this paper. The overall statistics of the training and test set has been provided in Table 2.

Table 1
Statistics of the train-set and the annotated test-set

	Sentences	Tokens	Entities	Unique Entities	Avg # tokens per sentence	Avg # Entities per sentence
Train	7800	232600	19012	14748	29.82	2.44
Test	1000	22873	3647	1249	22.87	3.65
Total	8800	255473	22659	15672	29.03	2.57

3. Training Set Creation with Entity Replacement

For the unlabelled corpus, we use the arxiv dataset containing $\tilde{2}$27,000 abstracts from various domains of Computer Science. We use the Papers with Code (PwC) corpus as a reference Knowledge Base to obtain a total of 14,748 model entity names. We use this list of named entities and create a set of distant silver labels by extracting the corresponding sentences out of the arxiv dataset that contain the same entity mention. We aim for exact match while also considering plural forms of the entity words. We obtain a total of 7800 sentences that contain a model named entity mention.

We plot a model entity vs frequency of occurrence in the entire corpus of our obtained sentences. We provide the plot in Figure 2. We notice that the distribution is long-tailed in nature, which is consistent with our hypothesis about scientific named entities as discussed in the Introduction section. This means that there are certain pre-dominant popular models that occur most frequently in the literature. The distribution tapers down and takes a long-tailed form, where most of the entities have a much significant lower number of occurrence in the literature. This can be attributed to the wide-spread use of certain models (like CNN), in the existing Computer Science research literature.

However, such a skewed distribution is unsuitable for training supervised custom NER models. The models tend to memorise and overfit for certain named entities. Hence, we use a simple entity replacement technique to deal with this bottleneck. More specifically, we detect the entity span of the model named entity in the occurring sentence. Then, we replace this entity with another entity from the entire set of model entities obtained from the Knowledge Base. We execute the process keeping the number of entity distributions to atmost 2 to maintain uniformity. After the entire process is completed, the entire train sentences set is set with uniformly distributed entities.

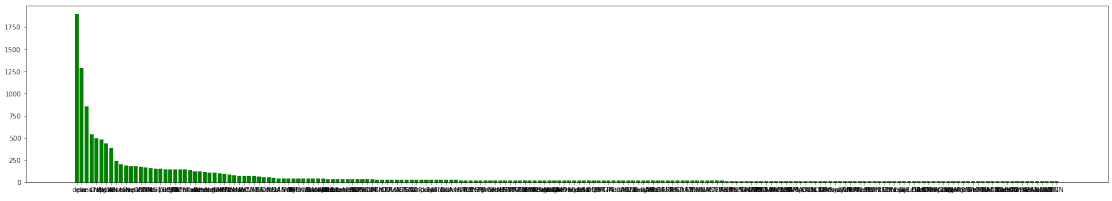

Figure 2: Distribution of entity occurrence frequency in the training dataset pre-replacement

Table 2
Result on Evaluation Dataset

Model	Precision	Recall	F1
BiLSTM + CRF (w/o replacement)	0.205	0.519	0.294
BERT + CRF (w/o replacement)	0.389	0.310	0.345
SciBERT+CRF (w/o replacement)	0.391	0.312	0.346
BERT+CRF (with replacement)	0.575	0.563	0.569
BiLSTM + CRF (with replacement)	0.628	0.631	0.629
SciBERT+CRF (with replacement)	**0.641**	**0.632**	**0.636**

4. Distantly Supervised NER Model

We aim to classify each token into its candidate labels among the BIO-tags. We use pre-trained BERT-based contextualised embeddings to capture the distributed representations from the sequence of tokens. We aim to detect the entire entity span and classify the entity span into specific entity types. We formulate this as a sequence labelling task, where we classify the sequence of tokens into a sequence of tokens. We consider the entire training sentences as the distantly labelled training data.

We experiment with multiple baselines which are standard for the sequence labelling process.

- **BiLSTM + CRF:** This BiLSTM-CRF model captures the contextual representations and encodes them into a bidirectional hidden state using BiLSTM. The CRF layer models the dependency among a sequence of tokens by considering the entire sequence label probability distribution.
- **SciBERT + CRF:** This model contains pre-trained SciBERT [16] embeddings trained on large scientific corpus. The SciBERT-embeddings are passed onto a CRF layer that models each sequence probability distribution.
- **BERT+CRF:** This consists of a pretrained BERT-model and a CRF layer to model sequence-level dependencies.

We evaluate the models on the gold test labels. We find that SciBERT model in combination with CRF provides the best performance. We also find that the entity replacement technique is particularly effective when dealing with long tailed entity distributions. We find that this simple technique offers a significant boost in performance across all models. It is effective in countering the overfitting bottleneck and successfully prevents memorisation of named entities. We rely only on distant labels to obtain strong performance on gold labels. We illustrate our best performing model in Figure 3.

5. Sentence Intent Classifier

We train a classifier to detect whether a sentence contains relevant information regarding models that solve a similar task as specified in the target research paper. For a target scientific document, we define a relevant model name as a model that the author has cited, which solves a task that is similar or relevant to the original task that the target paper is solving. To create an

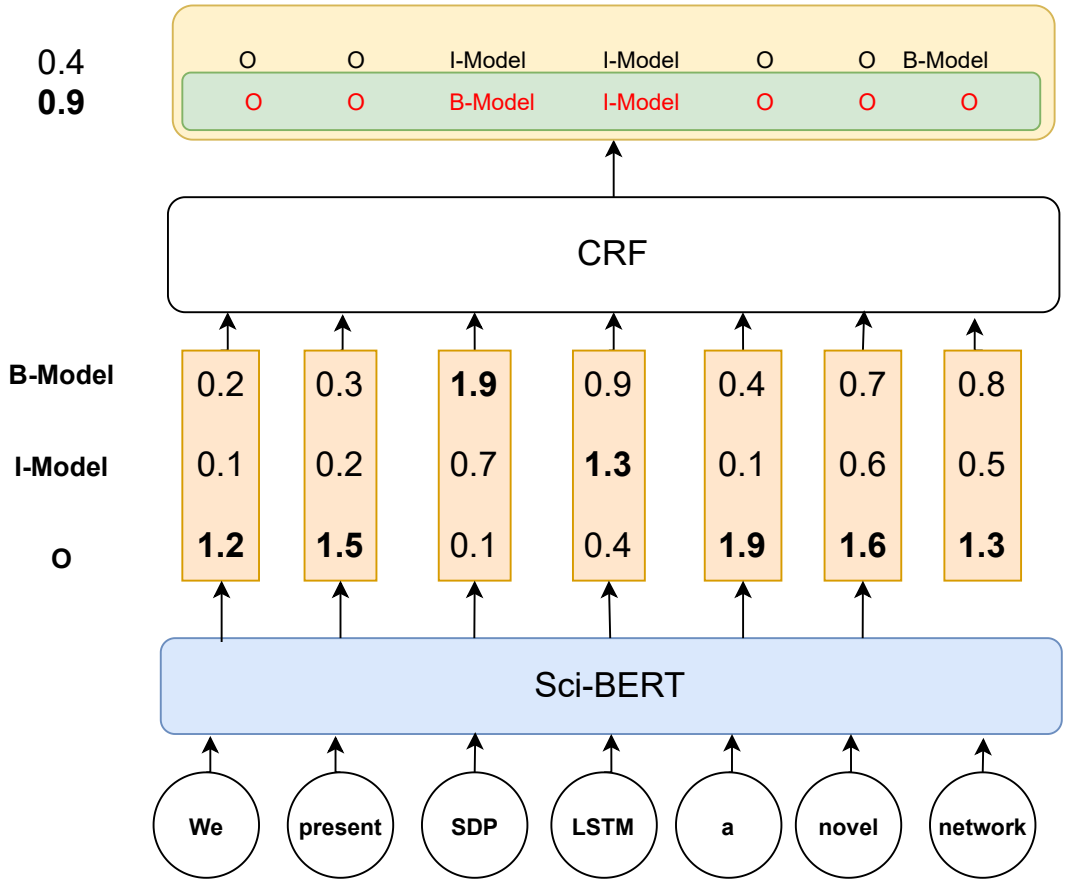

Figure 3: Our SciBERT-CRF Model for sequence tagging

automatically labelled dataset, we iterate over all sentences in the research corpora. If a sentence contains the words - 'Related Work' or 'Previous Work' or 'Baseline', then we take 15 sentences occurring after it. We assign positive labels to sentences containing model entity mentions by referring to our KB. We consider the maximal span for entity matching between our unlabelled text and KB. For creating negative samples, we randomly sample from all sentences and make sure the above words are absent and it also does not contain model entity mentions. We keep an equal distribution of positive and negative labels. An example of a positive and negative label is shown in Figure 4.

5.1. Training the classifier

The most commonly used approach of averaging BERT embeddings or using the output of the first token (the [CLS] token) yields subpar sentence representations [17]. Hence, we choose Sentence-BERT [17], a modification of the pretrained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can

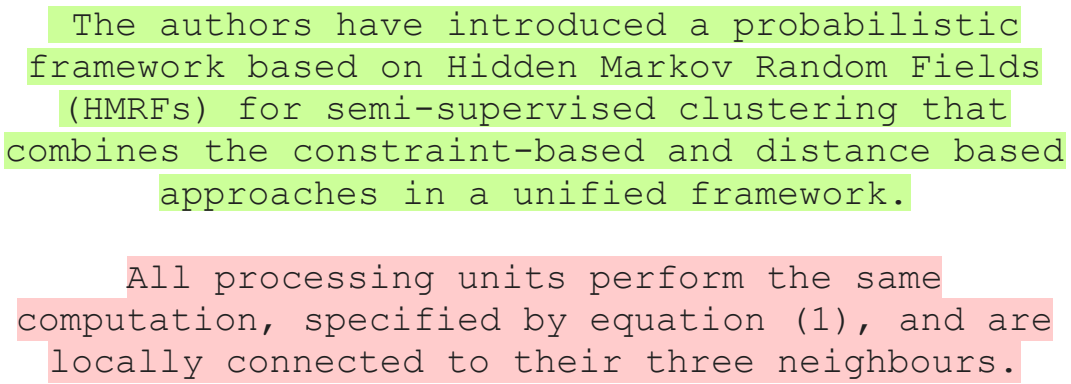

Figure 4: Sentence Intent : Positive label labelled with green, negative labelled with red

be compared using dot product. It takes a sentence as input and returns the corresponding sentence-level representation as output. We use Sentence-BERT to encode the sentences and use Logistic Regression as our binary classifier to train it on 15,518 labelled sentences, containing both citation and non-citation sentences. Positive and negative samples are equally distributed. The sentence dataset size is kept small to avoid compromising on the quality of the labels. The train-test split followed is 75-25. The testset accuracy (which, again, consists of both citation and non-citation sentences) is 86.41%.

6. Entity Citation Linker

For the entity citation linking, we iterate between all possible extracted entities and citation combination and get their closeness score, which is the string distance between an entity and the citation occurrence. We first take all the citations and keep the closest entity per citation. Then, we take all the entities and keep the closest citations per entity. This linking process is able to accurately link most of the extracted entities with their closest citations, as demonstrated by [15].

7. Pipeline Formation

We show two end-to-end pipelines in this paper. First, we show the entire training process for both model entity extraction and sentence intent classification. We use the same unlabelled corpora and Knowledge Base(KB) for both training processes. The automatic data labelling

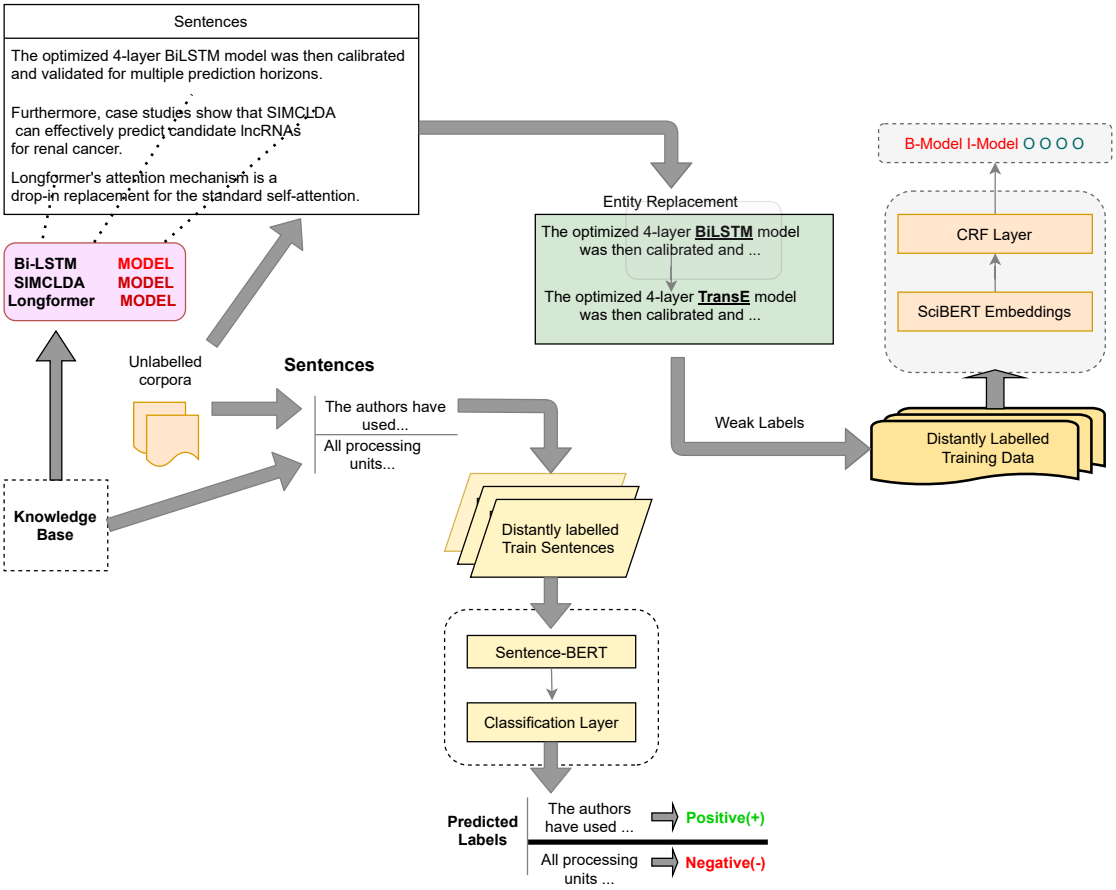

Figure 5: Training Pipeline for the Sentence Intent Classifier and NER model using unlabelled corpora and KB

process using the external KB followed by entity replacement is shown in Figure 5. This transformed dataset is used as distantly supervised training labels for input to the SciBERT-CRF NER model. Also, the sentence intent classifier approach is illustrated, where the KB is utilised to obtain weak binary labelled sentences from 'Related Work' section to train the classifier.

For the automated framework, we use a two stage pipeline. It takes a scientific research paper as input and obtains the citation sentences from it. Using the trained sentence intent classifier, we segregate the sentences into positive and negative labels. Only positively labelled sentences are passed into the next stage of the pipeline. We use the trained SciBERT-CRF NER model to extract entity mentions from the sentences. The model mentions are then linked with their respective citations using our Entity-Citation linker. The framework is illustrated in Figure 6.

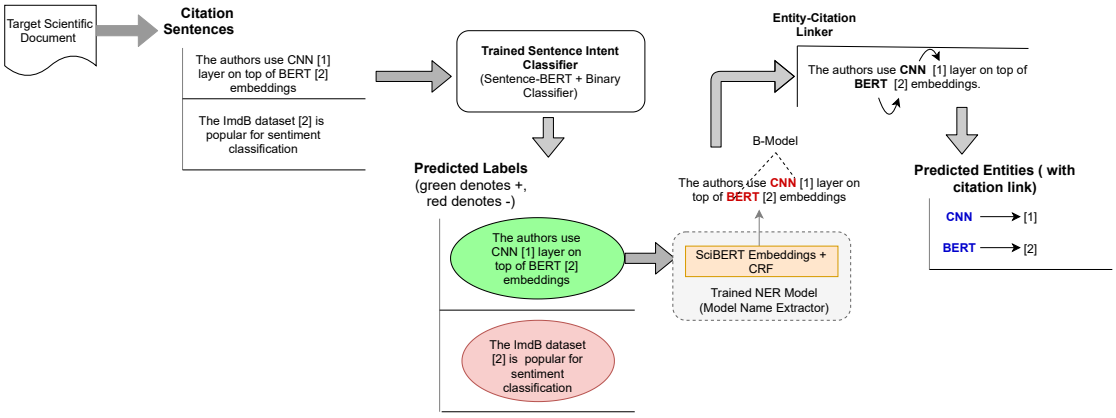

Figure 6: Entire Automated Framework utilising the trained models. The input is a scientific document and the output from the pipeline is a set of predicted model entities linked with their citation.

8. Error Analysis

We conduct error analysis for model entity extraction, sentence intent classification and entity citation linking. Some precision error is introduced into the model because for the training set we consider the maximum span of each entity and the I-Model entity occurrence (a token that lies inside a named entity) is high. We find in our evaluation dataset, the number of B-Model entities is massively more, which leads to the model misclassifying an O as an I for few sentences.

Also, due to the usage of citation sentences in the evaluation dataset, our model recognises the citation marker occurring right after the entity as an I-Model. Also, most of the citation sentences in the evaluation dataset has a large number of named entities occurring adjacently, as seen in many citation contexts. The model, which is trained on sentences from abstracts only, is unable to recognise all of them as entities sometimes.

For the sentence intent classification, our classifier often recognises sentences containing dataset names as a positive label. This can be attributed to the fact that citation sentences that refer to different datasets often have a similar structure to those citing model names of prior work. Lastly, for the entity citation linker, sometimes an entity that is associated with a citation marker occurs in the initial part of a sentence and its not the closest to the citation. This can lead to missed out or incorrect linking.

9. Implementation details

We use PyTorch framework to implement our NER model. We use the pre-trained SciBERT tokenizer and embeddings as input to a dropout layer with a dropout probability of 0.5 to prevent overfitting. We use a learning rate of 1e-5 and train all models for 20 epochs. We pass the output from the dropout layer through a linear layer with input dimension same as the hidden

dimension of SciBERT embeddings (768) and output dimension same as the number of labels (4). We train the BiLSTM-CRF model for 20 epochs. We annotate the evaluation dataset in the standard CoNLL BIO format. For Sentence-BERT, we use pretrained models available in Pytorch. We use the DBLP corpus consisting of $\tilde{4}$3K papers as our unlabelled research corpora to obtain the distant labels for training the classifier. For the Knowledge Base (KB), we use PwC public data corpus. For the CRF layer, we use allennlp[3] models library. We use regular expressions to extract citation sentences from papers that are written in the Springer LNCS/LNAI format.

10. Conclusion and future work

We have introduced a novel task of long-tailed model entity recognition from scientific documents. We test our gold standard evaluation set on multiple baselines. We also find that a simple strategy of entity replacement works well on small labelled datasets for distant supervision. We hope to extend this technique to different types of entities with low labelled data availability. We integrate our model in the automated pipeline framework to extract model names from scientific research documents and link them to their respective citations. For future work, we aim to utilise this pipeline on a large research corpus to obtain a map of benchmarked model names linked with their respective papers on a much larger scale. We believe our work will serve as an important starting point for mapping the entire research landscape of computer science.

References

[1] V. Kocaman, D. Talby, Biomedical named entity recognition at scale, CoRR abs/2011.06315 (2020). URL: https://arxiv.org/abs/2011.06315. arXiv:2011.06315.

[2] Y. Luan, L. He, M. Ostendorf, H. Hajishirzi, Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3219–3232. URL: https://aclanthology.org/D18-1360. doi:10.18653/v1/D18-1360.

[3] S. Jain, M. van Zuylen, H. Hajishirzi, I. Beltagy, Scirex: A challenge dataset for document-level information extraction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. arXiv:2005.00512.

[4] S. Mesbah, C. Lofi, M. V. Torre, A. Bozzon, G.-J. Houben, Tse-ner: An iterative approach for long-tail entity extraction in scientific publications, in: International Semantic Web Conference, Springer, 2018, pp. 127–143.

[5] Q. Liu, P. cheng Li, W. Lu, Q. Cheng, Long-tail dataset entity recognition based on data augmentation, in: EEKE@JCDL, 2020.

[6] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: ICML, 2001.

[7] H. L. Chieu, H. Ng, Named entity recognition with a maximum entropy approach, in: CoNLL, 2003.

[3] https://github.com/allenai/allennlp-models

[8] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, ArXiv abs/1812.09449 (2018).

[9] S. Goldberg, D. Z. Wang, C. Grant, A probabilistically integrated system for crowd-assisted text labeling and extraction, J. Data and Information Quality 8 (2017). URL: https://doi.org/10.1145/3012003. doi:10.1145/3012003.

[10] X. Wang, Y. Guan, Y. Zhang, Q. Li, J. Han, Pattern-enhanced named entity recognition with distant supervision, in: 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 818–827. doi:10.1109/BigData50022.2020.9378052.

[11] C. Liang, Y. Yu, H. Jiang, S. Er, R. Wang, T. Zhao, C. Zhang, BOND: bert-assisted open-domain named entity recognition with distant supervision, CoRR abs/2006.15509 (2020). URL: https://arxiv.org/abs/2006.15509. arXiv:2006.15509.

[12] M. A. Hedderich, L. Lange, D. Klakow, ANEA: distant supervision for low-resource named entity recognition, CoRR abs/2102.13129 (2021). URL: https://arxiv.org/abs/2102.13129. arXiv:2102.13129.

[13] F. Nooralahzadeh, J. T. Lønning, L. Øvrelid, Reinforcement-based denoising of distantly supervised NER with partial annotation, in: Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 225–233. URL: https://aclanthology.org/D19-6125. doi:10.18653/v1/D19-6125.

[14] Y. Yang, W. Chen, Z. Li, Z. He, M. Zhang, Distantly supervised NER with partial annotation learning and reinforcement learning, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 2159–2169. URL: https://aclanthology.org/C18-1183.

[15] S. Ganguly, V. Pudi, Competing algorithm detection from research papers, in: Proceedings of the 3rd IKDD Conference on Data Science, 2016, CODS '16, Association for Computing Machinery, New York, NY, USA, 2016. doi:10.1145/2888451.2888473.

[16] I. Beltagy, K. Lo, A. Cohan, Scibert: A pretrained language model for scientific text, arXiv preprint arXiv:1903.10676 (2019). URL: https://www.aclweb.org/anthology/D19-1371/.

[17] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://www.aclweb.org/anthology/D19-1410/.