Weakly Supervised Named Entity Recognition for Carbon Storage Using Deep Neural Networks

René Gómez Londoño[1,3], Sylvain Wlodarczyk[1], Molood Arman[1,2,3], Francesca Bugiotti[2,3](✉), and Nacéra Bennacer Seghouani[2,3]

[1] Services Pétroliers Schlumberger, 34000 Montpellier, France
{swlodarczyk,marman2}@slb.com
[2] Paris-Saclay University, CNRS, LISN, 91405 Orsay, France
{francesca.bugiotti,nacera.seghouani}@lri.fr
[3] CentraleSupélec, Paris-Saclay University, 91405 Orsay, France
rene.gomez@student-cs.fr

Abstract. Applying Transfer-Learning based on pre-trained language models has become popular in Natural Language Processing. In this paper, we present a weakly supervised Named Entity Recognition system that uses a pre-trained BERT model and applies two consecutive fine tuning steps. We aim to reduce the amount of human labour required for annotating data by proposing a framework which starts by creating a data set that uses lexicons and pattern recognition on documents. This first noisy data set is used in the first fine tuning step. Then, we apply a second fine tuning step on a small manually refined subset of data. We apply and compare our system with the standard fine tuning BERT approach on large amount of old scanned document. Those documents are North Sea Oil & Gas reports and the knowledge extraction would be used to assess the possibility of future carbon sequestration. Furthermore, we empirically demonstrate the flexibility of our framework showing that it can be applied to entity-identifications in other domains.

Keywords: Natural language processing · Named entity recognition · Deep neural networks · Stratigraphy

1 Introduction

Carbon sequestration in the North Sea is a way to reduce the global warming to below 1.5 °C. Several Northern European countries are currently engaging in solutions to store carbon under the North Sea in old Oil & Gas reservoirs. One of the difficulty in carbon storage is to entirely reassess the ancient reservoirs by interpreting many documents such as end of well reports, or core laboratory reports written during the long life cycle of the reservoir. Those documents are very heterogeneous and many of them are accessible only thanks to OCR techniques that do not provide clean data. In this case of study, the geologists study the rock strata and categorize them given the information embedded in those documents. Multiple analyses are performed in the domain of stratigraphy, that is the study of the physical and temporal relationships between rock layers or strata.

P. Pascal and D. Ienco (Eds.): DS 2022, LNAI 13601, pp. 227–242, 2022.
https://doi.org/10.1007/978-3-031-18840-4_17

For running this analysis, a source of information that is fundamental but generally underused is the set of geological well reports accumulated and produced during the whole history of a reservoir. Before the digital transformation of Oil & Gas industry, these analyses were run on a manually-converted subset of these documents. Nowadays, thanks to cloud computing and new technologies, it could be possible to handle a large amount of heterogeneous data and exploit a valuable source of historical information. Also, from the computational point of view, the analysis becomes more complex to evaluate, and analysis needs all useful data to be considered.

Those documents are underused because the geologists and the petrophysicists need to convert the information manually into structured tables. Usually, from these structured tables, they can populate the numerical models. These documents do not follow a given structure, and old documents are often written by typewriters and are accessible thanks to OCR techniques that do not provide clean data.

Name Entity Recognition (NER) [7] identifies the mentioned entities in unstructured texts and classifies them into target categories. Extracting the correct entities in the domain of the stratigraphy is capital information to evaluate a reservoir. Referring to our context, we can select as classification categories the period, the age, the era, the formation, etc. In the literature the performance of language models based on the Deep Neural Network (DNN) transformers architecture has produced interesting results in information extraction for many specific domains. The problem, however, is to provide the network with the necessary amount of labelled data required for the training phase. A recent state-of-the-art method for NER is to fine-tune a pre-trained BERT model using a labelled dataset with the corresponding entities we want to identify.

In our approach, we create this labelled dataset with a weakly supervised approach by using lexicons and labelling functions. This labelled dataset can be very large but also noisy as it comes from scanned documents and weak supervision. The hyperparameters of this first stage will be adapted to the "noisy" nature of the dataset. We then manually correct a very small subset of the noisy dataset and apply a second fine-tuning step with adapted hyperparameters. By comparing the results with a one-step fine-tuning approach, including the manually corrected dataset, we show that this workflow improves the results of precision by two (2) percentage points and recall by five (5) points. Increasing 5 points in recall means gaining a huge amount of information as we have massive data to process. We propose and test three language models with a human-reviewed data set. We present results for three Name Entity Recognition models, including a light version and compare with the state-of-the-art fine-tuned BERT model. Our results show a precision of 90%, recall of 96%, and F1 score of 93%. We finally provide some recommendations to apply our approach in other domains.

This paper is structured as follows. In Sect. 2 we describe the objectives, and we identify the main contributions of our approach. In Sect. 3 we introduce the fundamentals of our research focusing on the concepts related to Name Entity Recognition. In Sect. 4, we detail our methodology. In Sect. 5 we present the evaluation of the methodology.

In Sect. 6 we discuss related work, and we compare this research to the existing literature. Finally, in Sect. 7 we draw conclusions and some limitations and open challenges that remain subject for future work.

2 Overview

The objective of this research is to build a Named Entity Recognition system using Deep Neural Networks with a weakly supervised training process. To avoid complex feature engineering or continuous labelling and extraction work from the domain experts, we use a deep neural network-based approach. In the context of interest, training data is not available and annotating data is a labour-intensive task for geologists. To overcome such an obstacle, we decided to rely on a distant supervision approach to create noisy labels using external resources like regular expressions and dictionaries. It is a common scenario for a geologist to extract information from a report using regular expressions. Each regular expression identifies an entity and defines a sequence of characters that is used as a search pattern in each report. Multiple chunks of text could match the given search pattern, even text that is not a valid entity. The geologist might not realize this mismatch and erroneous entities are commonly identified (False-Positive). Such matches in NLP tools can produce alignment errors in the labels. As a second scenario, suppose instead using dictionaries related to the energy domain. The matching process should be straightforward and precise. Even in this scenario, False-Positives are commonly produced because of polysemy: words in the entity dictionary might be used in another context with a different meaning.

These two cases demonstrate that additional effort is required for cleaning the results by using pure text matching to extract the final entities. This would drastically hurt the system's scalability. To solve this problem, we use training data to build a deep neural network model that produces clean results and helps us by the generalization capacity of language models to detect unseen entities based on the contextual representation of their tokens Table 1.

The problem we introduced is studied in our domain but is common to many domains [14,21]. In Fig. 1 we show an example where NER is presented as a

Table 1. Sequence to Sequence Task Classification.

Tokens	BIO	BILOU
Diego	B-PER	B-PER
Armando	I-PER	I-PER
Maradonna	I-PER	I-PER
was	O	O
born	O	O
in	O	O
1960	B-DATE	U-DATE

Tokens-entities:

Diego Armando Maradona PERSON

was born in **1960** DATE .

sequence classification task. Specifically, we treat it as a sequence-to-sequence problem: given a token sequence (a sentence) as input, we produce the corresponding sequence of labels as output.

The approach is flexible enough to incorporate new target entities without labour-intense human annotation and sufficiently robust to reduce the necessity of result post-processing. The methodology is composed of the following steps:

1. The first step is the creation of a noisy training set for Named Entity Recognition. Given a set of documents, we aim to facilitate the text extraction task to generate a noisy training set on large data sets using dictionaries and regular expressions. The goal is to build an approach that can be run on distributed processing frameworks.
2. Given a noisy training set, we aim to use transfer learning to evaluate different DNN models incorporating contextual representations and using training techniques to avoid learning the noisy labels.
3. Given a set of pre-trained language models we want to evaluate the performances using a test set reviewed by human annotators. The evaluation shall be done having precision, recall and F1 score as metrics adapted for sequence evaluation.

2.1 Contributions

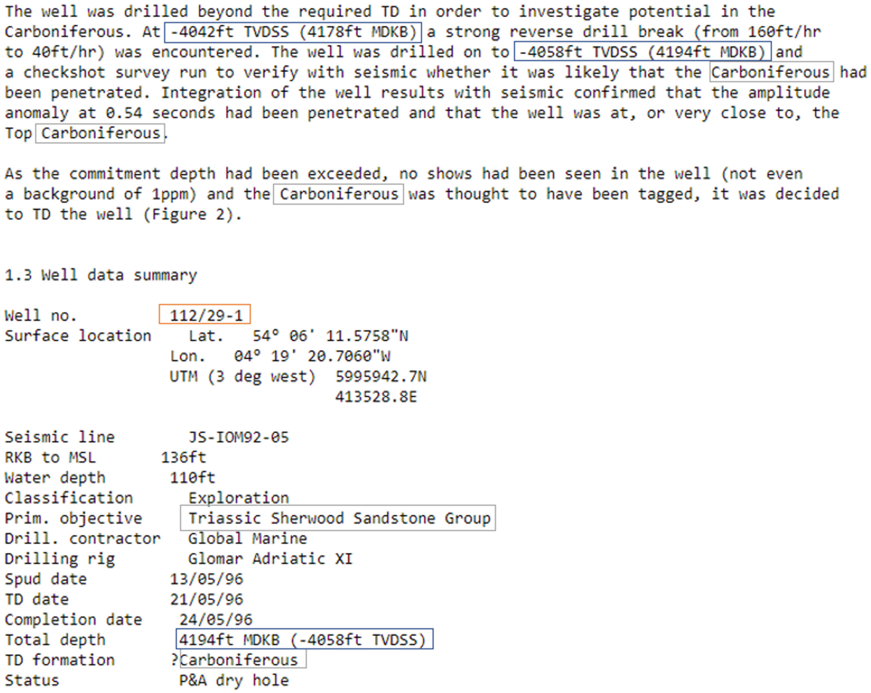
Given the described challenges, the methodology steps, and the technical constraints, our research achieves the following contributions:

1. The definition of a Named Entity Recognition System, establishing a baseline for future model benchmarking.
2. The implementation of a distributed framework enables data labelling using NER annotation schemas (like BIO and BILOU).
3. The implementation of a detailed two fine-tuning process of a pre-trained BERT model using in the first step, a large and noisy dataset created automatically and in the second step a small and clean human reviewed dataset. The hyperparameters are adapted in each step to fit the specific nature of each training data.
4. The evaluation of the approach utilizing sequence evaluation criteria from CoNLL (precision, recall, and F1 score adapted for text sequences) against human-reviewed data sets.

Furthermore, the same pipeline can be applied to other domains without a huge effort by changing the dictionaries and regular expressions.

3 Background

The main task of this project is to generate a framework to facilitate noisy data set creation, model training, and evaluation for a Named Entity Recognition system. For this purpose in our domain, we focus on a set of entities whose identification is a recurring challenge, given the nature of the geological reports.



The well was drilled beyond the required TD in order to investigate potential in the Carboniferous. At -4042ft TVDSS (4178ft MDKB) a strong reverse drill break (from 160ft/hr to 40ft/hr) was encountered. The well was drilled on to -4058ft TVDSS (4194ft MDKB) and a checkshot survey run to verify with seismic whether it was likely that the Carboniferous had been penetrated. Integration of the well results with seismic confirmed that the amplitude anomaly at 0.54 seconds had been penetrated and that the well was at, or very close to, the Top Carboniferous.

As the commitment depth had been exceeded, no shows had been seen in the well (not even a background of 1ppm) and the Carboniferous was thought to have been tagged, it was decided to TD the well (Figure 2).

1.3 Well data summary

Well no.	112/29-1
Surface location	Lat. 54° 06' 11.5758"N Lon. 04° 19' 20.7060"W UTM (3 deg west) 5995942.7N 413528.8E
Seismic line	JS-IOM92-05
RKB to MSL	136ft
Water depth	110ft
Classification	Exploration
Prim. objective	Triassic Sherwood Sandstone Group
Drill. contractor	Global Marine
Drilling rig	Glomar Adriatic XI
Spud date	13/05/96
TD date	21/05/96
Completion date	24/05/96
Total depth	4194ft MDKB (-4058ft TVDSS)
TD formation	?Carboniferous
Status	P&A dry hole

Fig. 1. An example of an end of well report scanned and converted to pdf format. We manually highlighted the various entities we would like to identify such as the DEPTH_INTERVAL, the FORMATION, the WELL_ID and the AGE.

An example of a well-report is shown in Fig. 1. The text present in the document is very noisy and difficult to interpret, even for a human reader. Documents of this format are written at the end of the drilling process of each well. The document contains critical information to assess a reservoir. When the interpretation of the reservoir is performed during the drilling process, the interpretation of the reported data is handled in real-time by humans. When we need to reassess reservoirs, for example, for evaluating carbon capture storage capabilities, wells were drilled decades ago, and the geologist cannot reread them to assign the information to thousands of wells. That is why we need to create models to perform the task automatically.

The text annotation pipeline uses external resources, matching lexicons in dictionaries, and regular expression patterns. The proposed approach avoids complicated pre/post-processing to provide positive examples for training.

To define the scope of this project, we selected a variant of useful entities to study similar scenarios like the ones proposed by [23]. In the following part of the section, we present each entity and the challenges that we commonly find in its identification process.

Defined Named Entities. An effective analysis must include entities that are: evident from the model, highly noisy, characterized by a limited number of possible instances and finally, entities that could be easily confused between them. Thanks to our methodology we expect to have good accuracy in all of them, but we also aim to detect which are the type of entities that remain challenging to define the future work in this project. The list of entities we are focused on in this presentation are:

(1) **Well Identifier** End of wells reports describe all the studies for one particular well. For instance, `30/2a-8` is a typical well identifier (WELL_ID entity) in the nomenclature of the north sea region. Regular expressions are flexible enough to detect those entities, but we will also detect many noisy labels. For this kind of entity, we want to avoid post-processing operations, improve the quality of the results and generalise the identification (i.e., the USA uses different nomenclature for well identification).
(2) **Period, age & epoch.** The geologic time scale is the "calendar" for events in Earth's history. It subdivides all time into named units of abstract time called eons, eras, periods, epochs, and ages. AGE and PERIOD entities are almost well-defined dictionaries, we expect high-performance detecting them. The EPOCH entity has a specific challenge as it comes from a dictionary containing both unique names and general terms (i.e., `early`, `late`, `lower`, etc.). We aim that in the sentence `the drilling process started late`, the word `late` will not be identified as an EPOCH.
(3) **Formation.** A geological formation consists of a certain amount of rock strata with comparable geological properties. This FORMATION entity is complex, with names ranging from rivers, areas, parks, towns or regions.
(4) **Depth interval and interval.** Depth intervals represent the boundaries of the formations. They usually follow a pattern of `number unit` to/-/and `number unit measure_reference`. The unit could be `feet` or `meters`, with their variations (i.e., `ft`,',", `mt` or `m`). MEASURE_REFERENCE is the reference point or type of the depth (i.e., `True Vertical Depth (TVD)`, `Measure Depth (MD)`, etc.). We also introduced a more relaxed entity, the INTERVAL that follows a similar pattern to the depth interval but without unit and MEASURE_REFERENCE. Since it is a flexible entity, it leads to False Positives, but it helps the model to identify some depth intervals that would be lost otherwise.

4 Methodology

In this section we describe our methodology from the labels generation to the training process of the DNN. Afterwards, we present a more in-depth study for the DNN's training process and finally explain how we use pre-trained language models to accomplish our downstream task.

An overview of the methodology is presented in Fig. 2. It involves multiples stages, starting with the data set creation and finishing with the model training and evaluation.

(1) The lack of labelled training data has limited the development of NLP tools. We use distant supervision resources (dictionaries & regular expressions) to

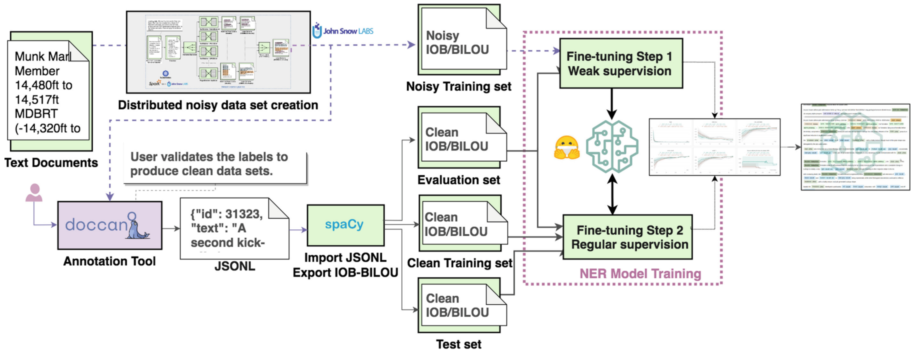


Fig. 2. Implementation pipeline.

create labelled data in a semi-automatic way. This removes the need for intense manual data labelling. The problem is that we get not only True-Positive but also False-Positive examples.

(2) Since we are going to use a noisy data set, we clean part of the labels with an annotation tool to generate a proper evaluation and test set. Notice that we don't annotate from scratch but review the semi-automatic generated labels. We just correct enough examples to control the training process and evaluate the final results.

(3) We used noisy samples to train the model with most of the default parameters, varying the batch size and learning rate. Each batch contains a random number of clean and noisy examples.

(4) According to [23] and [1], using the recommended parameters should be enough. Still, we monitor the training process with a small clean evaluation set to detect in which case the noisy examples start to be learnt by the neural network. In theory, we should see fluctuating loss and performance metrics for the evaluation set.

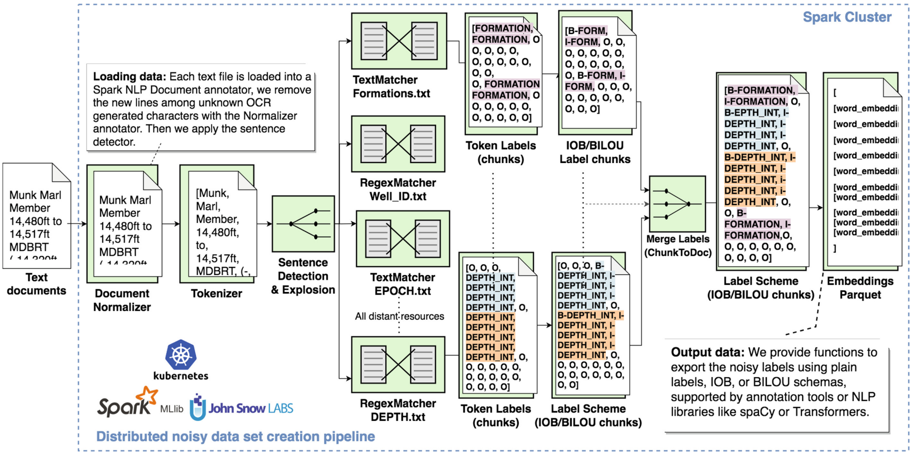
(5) The output model is selected on the basis of the sequence evaluation performance.

4.1 Noisy Data Set Creation

One of the driver elements in our methodology is the data set creation. Without labelled data we follow a weak supervision approach using dictionaries and regular expressions. Our data set creation pipeline is detailed in Fig. 3.

We remove newline characters and normalize the text to avoid rare characters produced by the OCR system. We then tokenize the text and run a sentence detector model.

The matcher component finds the corresponding chunks where the dictionary or regular expressions match the specific sentence. Lexicons were collected from different internal applications where stratigraphic units are used to describe


Spark Cluster
Munk Marl Member 14,480ft to 14,517ft MDBRT
Text documents
Munk Marl Member 14,480ft to 14,517ft MDBRT
Document Normalizer
[Munk, Marl, Member, 14,480ft, to, 14,517ft, MDBRT, (-,
Tokenizer
Sentence Detection & Explosion
TextMatcher Formations.txt
RegexMatcher Well_ID.txt
TextMatcher EPOCH.txt
All distant resources
RegexMatcher DEPTH.txt
[FORMATION, FORMATION, O O, O, O, O, O, O, O, O, O, O, O, O, FORMATION FORMATION, O O, O, O, O, O, O, O, O, O, O]
Token Labels (chunks)
[B-FORM, I-FORM, O, O, O, O, O, O, O, O, O, O, O, O, O, B-FORM, I-FORM, O, O, O, O, O, O, O, O, O, O, O]
IOB/BILOU Label chunks
[O, O, O, DEPTH_INT, DEPTH_INT, DEPTH_INT, DEPTH_INT, O, DEPTH_INT, DEPTH_INT, DEPTH_INT, DEPTH_INT, DEPTH_INT, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O]
Token Labels (chunks)
[O, O, O, B-DEPTH_INT, I-DEPTH_INT, i-DEPTH_INT, I-DEPTH_INT, O, B-DEPTH_INT, I-DEPTH_INT, I-DEPTH_INT, I-DEPTH_INT, I-DEPTH_INT, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O]
Label Scheme (IOB/BILOU chunks)
Merge Labels (ChunkToDoc)
[B-FORMATION, I-FORMATION, O, B-EPTH_INT, I-DEPTH_INT, I-DEPTH_INT, I-DEPTH_INT, O, B-DEPTH_INT, I-DEPTH_INT, I-DEPTH_INT, i-DEPTH_INT, i-DEPTH_INT, O, O, B-FORMATION, I-FORMATION,O, O, O, O, O, O, O, O, O, O, O]
Label Scheme (IOB/BILOU chunks)
[[word_embeddi [word_embeddi [word_embeddi [word_embeddi [word_embeddi [word_embeddi [word_embeddi [word_embeddi [word_embeddi [word_embeddi]
Embeddings Parquet
kubernetes
Spark MLlib John Snow LABS
Distributed noisy data set creation pipeline

Loading data: Each text file is loaded into a Spark NLP Document annotator, we remove the new lines among unknown OCR generated characters with the Normalizer annotator. Then we apply the sentence detector.

Output data: We provide functions to export the noisy labels using plain labels, IOB, or BILOU schemas, supported by annotation tools or NLP libraries like spaCy or Transformers.

Fig. 3. Spark NLP implementation pipeline

well logs. However, public information like Wikipedia's taxonomies or specific knowledge bases is commonly used as data sources in such applications.

The matches are then converted into token-level labels. When we have overlapped labels, we have to keep the longest match. Here it is crucial to keep the text alignment with the labels, always keeping one label for each token.

Finally, we built an exporter to save BIO/BILOU files.

4.2 Overcoming Noisy Labels Effect

In this section we present our steps to train the DNN avoiding the noise overfitting.

First of all, to reduce some of the negative effects of label errors, we use language models, which means that not only the entity influences the learning process but also the context in which it appears. Under this scenario, noisy examples are harder to be learnt. Not all of them follow one common usually perfect pattern as clean examples do. As additional bias they occur in similar contexts: the representation is then not as close as the clean examples. Batch size and learning rate are fundamental hyperparameters in our context as already stated by [23,25]. We deeply rely on the straightforward approach explored by [23]. The authors demonstrated that larger batch sizes are better to overcome the effect of noisy data labels. The authors argue that the negative impact of uncorrelated or less correlated noise types is diminishing since updates caused by noisy samples are overwhelmed by gradient updates from clean samples. [25] got similar results, observing that DNN trained on noisy labelled datasets with a high learning rate do not memorise noisy labels.

Two-Step Fine-Tuning. [23] suggests that learning with big batch size is enough to mitigate the noise effect. We follow this approach using also a clean evaluation set. This enables to monitor the training evolution to ensure the best hyperparameters configuration for removing noise. Finally, we select the batch size that presents the most consistent behaviour during training. To avoid noise overfitting, we might need to use an early stopping of the training process. We consider this as an adaptation step towards all our domain-specific language where we learn simple entities and patterns. However, if the model does not learn noisy labels it might also be having lousy performance in the difficult to learn patterns, or confusing similar classes like INTERVAL and DEPTH_INTERVAL. In such a case, we plan to run a second fine-tuning stage with regular supervision. It is, having a small training set with clean examples and using traditional hyperparameters to refine the details that might be missing during the first fine-tuning stage. Moreover, we want to evaluate if applying this methodology, we can change the behaviour in the polysemy problem. We expect to influence the algorithm and to see some changes in the predictions for words like *lower* and *late* in the EPOCH entity as the primary example. Additionally, since we are in a transfer learning setting, we use some clean and reliable negative examples to execute a second fine-tuning stage. We intend to evaluate if this helps the model improving the details that might be excluded during the first fine-tuning phase. In this second step we must avoid the forgetting problem [22]. We do this with following strategy: regardless of the errors, we won't target any particular entity but randomly select examples to learn the details. We want to keep the clean training set small, with a similar size to the validation set. Such training set has examples from all the previously learned entities. We are not incorporating a new named-entity or a completely different context. This two-step fine-tuning strategy works even better in more complicated scenarios, where the original training data is not available. Hence, we don't expect any drawback from using it in this more convenient environment.

5 Evaluation

The architecture of the system is provided in Fig. 2. For the project implementation, we use PySpark in a Kubernetes cluster deployed on Google Cloud Platform. For the data set creation, we used a cluster with 16 GB in the driver node and 4 workers with 8 GB RAM each. The training process was done in one single node with 64 GB RAM without GPU.

Specifically, we use Spark NLP for weak data labelling and train the models using Transformers (PyTorch version). The number of resources assigned to the project varied according to the cluster state or the executed task. Our normal configuration for the cluster was with 32 GB of memory in the driver node and four executors with 8 GB each.

The output from the lexicons and regular expressions were cleaned and cross-validated by two engineers using Doccano. Complex examples were verified with domain experts. The reports are publicly available on the Oil and Gas Authority website [16]. The pre-processing code and the OCR were performed by Schlumberger and are not publicly available. The training process was done using the

Table 2. Data sets for training.

Entity	Noisy set	Clean set	Eval set	Test set
WELL_ID	15754	125	151	345
FORMATION	18424	159	167	381
INTERVAL	9218	93	83	189
EPOCH	19366	166	156	360
AGE	11243	130	118	280
PERIOD	7416	79	87	166
DEPTH_INT	4258	40	56	92
TOTAL	85679	792	818	1813

public available HuggingFace Transformers training process with the described hyperparameters. We track our experiments using Weights&Biases (W&B).

Data sets. We collected examples from one thousand different geological reports with more than seven million tokens. We executed the automated noisy data labelling pipeline and we got more than 125,000 sentences with approximately 227,000 entities. However, we did not use the entire data set for our proof of concept. We randomly selected sentences to create the training and evaluation sets. For the noisy training, clean training, and test set we selected respectively 50000, 500, and 1000 sentences.

We present the entities and the number of instances in Table 2.

Evaluation Results. Across all the experiments we use seqeval [18], a framework for sequence labelling evaluation following the CoNLL-2000 shared task data guidelines. Instead of evaluating token by token, the sequence is evaluated based on complete detected named entities. The framework also takes into account class imbalance, ignoring, for instance, the tokens that are not entities labelled as O. We focused our experiments in testing several models using different batch sizes and learning rates as described in the methodology section, evaluating its effect in the fine-tuning steps. For other hyper-parameters, we used the recommended values suggested in [9], with sequences of maximum 128 tokens. Note that we use BERT-Base-Cased like models because we have a lot of capitalized names or upper case codes in our documents.

Figure 4 shows the results over three models: BERT [9] and the HuggingFace distilled version of BERT and RoBERTa [24].

We could see that most of the time, the distilled version of RoBERTa is outperformed by the BERT and the distilled BERT model in all metrics. Hence we decided to focus on the BERT and distilled BERT model.

We present the performance of the two selected models in Table 3.

As explained in Sect. 3 the high performance in entities like DEPTH_INTERVAL and PERIOD were expected, since these entities are consistent with dictionaries

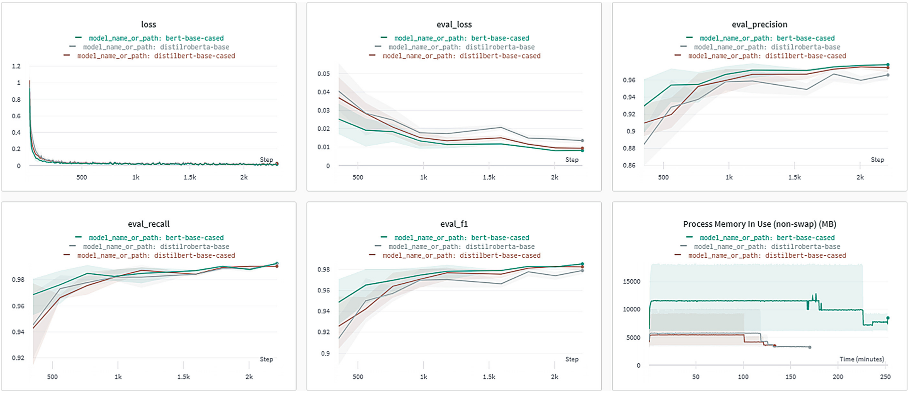


Fig. 4. Benchmark of three pre-trained models.

Table 3. Results for test set. DistilBERT and BERT with a BatchSize of 64. First and second fine-tuning results

Named entity	d-BERT-64 St 1			d-BERT-64 St 2			BERT-64 St 1			BERT-64 St 2			Supp
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
DEPTH_INT	0.99	0.99	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.95	0.98	0.96	92
FORMATION	0.90	0.86	0.88	0.84	0.90	0.87	0.92	0.87	0.89	0.85	0.91	0.88	381
WELL_ID	0.46	0.48	0.47	0.90	0.96	0.93	0.46	0.48	0.47	0.91	0.96	0.94	345
AGE	0.97	0.97	0.97	0.96	0.97	0.97	0.96	0.97	0.97	0.96	0.97	0.97	280
PERIOD	0.98	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	166
INTERVAL	0.92	0.97	0.95	0.93	0.97	0.95	0.93	0.95	0.94	0.92	0.96	0.94	189
EPOCH	0.89	0.98	0.93	0.89	0.97	0.93	0.91	0.99	0.94	0.90	0.98	0.94	360

or well-defined patterns. It helps us to evaluate that we are not degrading the performance in the well-known consistent cases. Furthermore, with them, we evaluate the performance in other entities like INTERVAL or FORMATION, where the former is a pattern similar to other non-entities tokens present in the text, and the latter comes from incomplete dictionaries. The WELL_ID is the hardest entity to learn since they have an inconsistent pattern that matches other tokens (i.e., section numbers and coordinates, which can also appear without context).

To validate the advantage of using the two-step fine-tuning approach, we learned a single-step fine-tuning BERT model and the equivalent distilled BERT model using the combination of the noisy and the clean training set as a unique training data set. We present the performances of these models in Table 4.

Result Discussion. The two steps training method presents a slightly better precision (2% points improvement) than the single-step fine-tuning BERT model. Furthermore, the two steps model training has, as expected, a better recall (up

Table 4. Results for test set. BERT with a batch size of 64. BERT Stage 1 and Stage 2 are the two fine-tuned results, whereas stage 2 is the final result. BERT Single-Step is the single-step fine-tuned BERT model

Bert version	Named entity	BERT Stage 1			BERT stage 2			BERT single-Step			Supp
		P	R	F1	P	R	F1	P	R	F1	
Distilled Bert	DEPTH_INT	0.99	0.99	0.99	0.98	0.98	0.98	0.96	0.98	0.97	92
	FORMATION	0.9	0.86	0.88	0.84	0.9	0.87	0.90	0.86	0.88	381
	WELL_ID	0.46	0.48	0.47	0.9	0.96	0.93	0.62	0.64	0.63	345
	AGE	0.97	0.97	0.97	0.96	0.97	0.97	0.97	0.98	0.97	280
	PERIOD	0.98	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	166
	INTERVAL	0.92	0.97	0.95	0.93	0.97	0.95	0.93	0.96	0.94	189
	EPOCH	0.89	0.98	0.93	0.89	0.97	0.93	0.91	0.99	0.94	360
	Micro avg	**0.84**	**0.86**	**0.85**	**0.91**	**0.96**	**0.93**	**0.87**	**0.89**	**0.88**	**1813**
Bert	DEPTH_INT	0.98	0.98	0.98	0.95	0.98	0.96	0.92	0.98	0.95	92
	FORMATION	0.92	0.87	0.89	0.85	0.91	0.88	0.90	0.87	0.89	381
	WELL_ID	0.46	0.48	0.47	0.91	0.96	0.94	0.72	0.74	0.73	345
	AGE	0.96	0.97	0.97	0.96	0.97	0.97	0.97	0.97	0.97	280
	PERIOD	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	166
	INTERVAL	0.93	0.95	0.94	0.92	0.96	0.94	0.95	0.96	0.96	189
	EPOCH	0.91	0.99	0.94	0.9	0.98	0.94	0.91	0.99	0.94	360
	Micro avg	**0.85**	**0.86**	**0.85**	**0.91**	**0.96**	**0.94**	**0.89**	**0.91**	**0.90**	**1813**

to 5% points) given the fact that the longer training time for the single-step fine-tuning BERT model reduces its flexibility to identify new entities.

We see that the second fine-tuning step improves the accuracy and precision of the models. First, as shown in Table 3 the high performance was maintained for the consistent entities, which was expected since the second training set contains clean examples for all the entities. In other words, we introduced examples of all entities avoiding the catastrophic forgetting problem.

Furthermore, the training set in the second step was focused on providing cleaner examples for the WELL_ID and AGE. Therefore, it makes sense that it helped the model predict multi-token WELL_ID. The second fine-tuning step catches the full WELL_ID with proper boundaries, as shown in Table 5.

In a second round of analysis, we also evaluated the generalization capacity of the models by testing non-existing ages such as `Sylvanian` or `Renotian`. In the sentence "`The late Sylvanian is...`", the token `late` was identified as an EPOCH with a probability of 99% and `Sylvanian` as an AGE with a probability of 70%. Notice that the model never saw `Sylvanian` as an example before, but it appears with a similar structure to other AGE names, and in the same sentence (context) there is an EPOCH (the word `late`), hence the model classified it as an AGE. Nevertheless, it is only 70% confident about the prediction (since it has never been seen before). In the sentence "`I was late for class`", the token `late` was NOT identified as an EPOCH by the BERT_ST2 model: it is the same token as a valid EPOCH, but the context is not valid; therefore it has another

Table 5. Example of multi-token WELL_ID. The model fails to catch the full multi-token WELL_ID with only the first fine-tuned step but succeeds with the second fine-tuned step.

Token	BERT ST1	BERT ST2
Well	B-WELL_ID	B-WELL_ID
13_22a	B-WELL_ID	I-WELL_ID
–	B-WELL_ID	I-WELL_ID
C29X	O	I-WELL_ID
wellsite	O	O
Geological	O	O

meaning. It shows that the second training step has a great potential to improve the capacity to remove the False Positives introduced by words with multiple meanings.

6 Related Work

In this section, we illustrate related work starting from introducing the works that generally studied Named-entity recognition. In the second part of this section, we will analyze the approaches that treat NER using pre-trained word representations. Finally, we will analyse the approaches used in Oil & Gas Industry and other domains.

Named-entity recognition systems have been studied and developed for decades. Nevertheless, the methods using deep neural networks (DNN) have only been introduced in the last decade [14], with recent special improvements given the new capabilities with pre-trained models and transfer learning [14]. **Models with pre-trained word representations** The widely used approach based on DNNs for NER was proposed in [6]. This model applies a Convolutional Network Architecture to the token sequence. Posterior works typically change the encoding part, which ranges from char-based, word-based, and encoding additional features. Examples include predefined word representation like word2vec, GloVe, or BERT or the explicit inclusion of suffixes and prefixes. In this context [12] work focused on changing the CNN with a bidirectional LSTM encoder. They do not perform any pre-processing; they do not take into account morphological information from characters or words. Instead, all features are learned by a CNN, achieving SOTA results. Other approaches [17] take advantage of the usage of a large semantic database and implement distant supervision: the relation classifier is trained using textual features.

Some models are based on general word embeddings, that are fine-tuned for NER. The original work, illustrated in [19], presented an F1 score of 92.2 over the CoNLL 2003 test set. [5] improves this result to 92.6 by using Cross-View Training (CVT). The semi-supervised learning algorithm improves the representations of a Bi-LSTM sentence encoder utilising a mix of labelled and unlabelled data. Zalando Research has also made a great effort in providing SOTA models, getting an F1 score of 92.86 over the same data set [2]. Using the BERT base model(i.e. using the pre-trained embeddings) gives an F1 score of 91. Fine-tuning the same model for NER, however, improves this score to 96.4. In 2019 the pooled version of the approach improved this score to 93.18 [2]. **Energy Industry** a NER for geosciences trained for the Chinese language has been proposed by [20]. They use a generative model, building a data set from seed terms without labelled data with good results. Another system from geoscience is the Portuguese NER [7]. It defines the target entities for the Brazilian sedimentary basins. They used a conventional approach with three different embeddings configurations tested using a BiLSTM-CRF architecture. Some other approaches are focused on unsupervised clustering-based technique to match attributes of a large number of heterogeneous sources as also proposed in [3] to identify entities.

NER in Other Domains. NER is well studied in specific domains like medical data, neuroscience, or scientific data. Bio-NER for the biomedicine field has named entities related to RNA, protein, cell type, cell line, and DNA with different shared tasks. Similarly to the general field, up to 2018 BiLSTM-CRF [13]

Noisy Labels. Label noise has always been an existing problem in machine learning, due to the potential negative impact it has over classification as also stated in [10].

Since weakly supervised learning is gaining a huge attraction, dealing with noise in Deep Neural Networks has become a highly active research field for representation learning [8]. Most works focused on generating and aggregating synthetic noise to well-known data sets [23]. [1] identifies three different approaches to mitigate the effect of noisy labels as widely described in [4,11,15].

7 Conclusion

Named Entity Recognition is the first fundamental step for Information Extraction and Knowledge Base creation. The main objective of our research was to build a NER System for the Oil & Gas industry. However, instead of creating one model for some specific entities in this domain, We aimed to explore a methodology/framework that facilitates the creation of a Named Entity Recognition system based on noisy data labels. The methodology is flexible enough to incorporate new target entities without labour-intense human annotation and sufficiently robust to enhance generalization. We create labels using distant supervision resources like dictionaries and regular expressions. Distant supervision introduces noisy labels, translating mainly into False Positives in the training set. To mitigate the effect of noisy labels, we followed a method with three key elements: (1) Distributed processing - to enable the labelling of bigger data sets than the ones we could have obtained with manual annotation. (2) Transfer learning with pre-trained language models - to learn bidirectional context representations in our domain-specific corpus (3) SOTA training techniques -

to avoid over-fitting the noisy examples. Furthermore, we proposed a two-step fine-tuning approach that showed to be effective in improving the prediction capacity in hard-to-learn named entities. We apply this model to many domain documents from the north-sea and create a knowledge graph that would be used to feed a model.

A similar approach could be applied in other domains where many documents are available. In such scenarios, distant supervision enables extracting thousands of sentences with entities. Even with noise, bigger data sets and the proposed training process will help the model to capture the regular context where entities occur, helping to remove false positives even in domains with polysemy challenges. As future work, we would like to explore the effect of the size of the clean data set on the model performance following our approach. This will allow us to provide clear recommendations on how much data has to be cleaned for the second fine-tuning step. Moreover, as an extension of our work, we can consider replacing the regular expressions and dictionary approach with labelling and transformation functions like in Snorkel [21].

Acknowledgements. We are grateful to the Oil & Gas Authority that provided the access to wells reports used in our research (under the Oil and Gas Authority Licence [16]).

References

1. Abid, A., Zou, J.Y.: Improving training on noisy stuctured labels. CoRR (2020)
2. Akbik, A., Bergmann, T., Vollgraf, R.: Pooled contextualized embeddings for named entity recognition. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 724–728 (2019)
3. Arman, M., Wlodarczyk, S., Bennacer Seghouani, N., Bugiotti, F.: PROCLAIM: an unsupervised approach to discover domain-specific attribute matchings from heterogeneous sources. In: Herbaut, N., La Rosa, M. (eds.) CAiSE 2020. LNBIP, vol. 386, pp. 14–28. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58135-0_2
4. Bahri, D., Jiang, H., Gupta, M.R.: Deep k-nn for noisy labels. CoRR (2020)
5. Clark, K., Luong, M.-T., Manning, C.D., Le, Q.V:. Semi-supervised sequence modeling with cross-view training. CoRR (2018)
6. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. CoRR (2011)
7. Consoli, B., Santos, J., Gomes, D., Cordeiro, F., Vieira, R., Moreira,V.: Embeddings for named entity recognition in geoscience Portuguese literature. In: Proceedings of The 12th Language Resources and Evaluation Conference, pp. 4625–4630, Marseille, France, 2020. European Language Resources Association
8. Deng, Z., Dong, Y., Pang, T., Su, H., Zhu, J.: Adversarial distributional training for robust deep learning. CoRR (2020)
9. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805 (2018)
10. Frenay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE Trans. Neural Netw. Learn. Syst. **25**(5), 845–869 (2014)

11. Ghosh, A., Kumar, H., Sastry, P.S.: Robust loss functions under label noise for deep neural networks. AAAI'17, pp. 1919–1925. AAAI Press (2017)
12. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR (2015)
13. Khan, M.R., Ziyadi, M., Abdelhady, M.: Mt-bioner: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers. CoRR (2020)
14. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. CoRR (2018)
15. Li, J., Wong, Y., Zhao, Q., Kankanhalli, M.S.: Learning to learn from noisy labeled data. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5046–5054 (2019)
16. Licence. Oil and Gas Authority Licence (2022) Accessed Jan 2022. https://www.ogauthority.co.uk/media/5850/oga-open-user-licence_210619v2.pdf/
17. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2, ACL '09, pp. 1003–1011, USA, 2009. Association for Computational Linguistics
18. Nakayama, H.: seqeval: A python framework for sequence labeling evaluation (2018). https://github.com/chakki-works/seqeval
19. Peters, M.E.,et al.: Deep contextualized word representations, CoRR (2018)
20. Qiu, Q., Xie, Z., Liang, W., Tao, L.: Gner: a generative model for geological named entity recognition without labeled data using deep learning. Earth Space Sci. **6**, 931–946 (2019)
21. Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Sen, W., Ré, C.: Snorkel. Proc. VLDB Endowment **11**(3), 269–282 (2017)
22. Robins, A.V.: Catastrophic forgetting, rehearsal and pseudorehearsal. Connect. Sci. **7**, 123–146 (1995)
23. Rolnick, D., Veit, A., Belongie, S.J., Shavit, N:. Deep learning is robust to massive label noise. CoRR (2017)
24. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. ArXiv, abs/1910.01108 (2019)
25. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. CoRR (2018)