Knowledge extraction from aeronautical messages (NOTAMs) with self-supervised language models for aircraft pilots

Alexandre ARNOLD
Fares ERNEZ
Catherine KOBUS
Marion-Cécile MARTIN
Airbus (AI Research)
firstname.lastname@airbus.com

Abstract

During their pre-flight briefings, aircraft pilots must analyze a long list of NOTAMs (NOtice To AirMen) indicating potential hazards along the flight route, sometimes up to 100 pages for long-haul flights. NOTAM free-text fields typically have a very special phrasing, with lots of acronyms and domain-specific vocabulary, which makes it differ significantly from standard English. In this paper, we pretrain language models derived from BERT on circa 1 million unlabeled NOTAMs and reuse the learnt representations on three downstream tasks valuable for pilots: criticality prediction, named entity recognition and translation into a structured language called Airlang. This self-supervised approach, where smaller amounts of labeled data are enough for task-specific fine-tuning, is well suited in the aeronautical context since expert annotations are expensive and time-consuming. We present evaluation scores across the tasks showing a high potential for an operational usability of such models (by pilots, airlines or service providers), which is a first to the best of our knowledge.

1 Introduction

Each upcoming flight requires a preparation phase for the crew. During this phase, the pilots check all the elements concerning the flight, being the meteorological conditions, fuel supply, or safety related notifications. Pilots receive these notifications from the aviation authorities under the form of small text messages, called NOtice To AirMen (NOTAM). It represents a large number of messages to read and process before the flight, sometimes up to 100 pages for long-haul flights, which translates to a long analysis time. The messages are mostly, but not only, written in the English language. However, the phrasing being very special, with a lot of acronyms, technical words and without the usual grammar and syntax rules, the language differs from standard English.

In this paper, we aim to apply the latest advances in Natural Language Processing (NLP) to the aeronautical field leading to more autonomy and less overhead for the pilots. In particular, the use of language models like BERT enables leveraging lots of unlabeled data for pretraining and fine-tuning on downstream tasks with limited amounts of labeled data. Our main contribution is to present a knowledge extraction pipeline adapted to the aeronautical context. We introduce a language model trained from scratch on a large amount of raw NOTAMs, followed by three downstream tasks: criticality prediction, named entity recognition and translation into a structured language called Airlang.

This paper is organised as followed: in Section 2, the NOTAMs are detailed both on the operational and the linguistic side, the problem is defined in Section 3 with a focus on the three downstream tasks. In Section 4, after a brief reminder of the state of the art, we present our approaches and the results of our experiments.

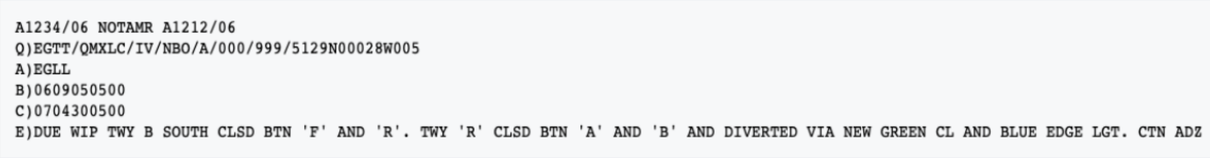
2 NOTAMs in aeronautical context

A NOTAM is a message filled by aviation authorities to alert pilots about potential hazards along a flight route or at a location that could affect the flight. The message can inform about temporary disruptions (from a few hours to one year maximum) on aeronautical infrastructures (for example, closure or limited usage of runway or taxiway in a given airport), about inoperable radio navigational aids, military exercises with resulting airspace restrictions, temporary erections of obstacles near airfields (e.g. cranes), passage of flocks of birds through airspace, etc. An example of a NOTAM message is shown in Figure 1; more details about the fields can be found in Appendix A.

2.1 Operational point of view

During the pre-flight briefing phase, a pilot has to read all NOTAMs relevant for the flight in order to

Proceedings of NAACL-HLT 2022: Industry Track Papers, pages 188 - 196
July 10-15, 2022 ©2022 Association for Computational Linguistics



```
A1234/06 NOTAMR A1212/06
Q)EGTT/QMXLC/IV/NBO/A/000/999/5129N00028W005
A)EGLL
B)0609050500
C)0704300500
E)DUE WIP TWY B SOUTH CLSD BTN 'F' AND 'R'. TWY 'R' CLSD BTN 'A' AND 'B' AND DIVERTED VIA NEW GREEN CL AND BLUE EDGE LGT. CTN ADZ
```

Figure 1: NOTAM example with its different Q, A, B, C and E fields

guarantee safety.

Reading these NOTAMs is a mandatory task for the pilot but can be long and challenging. First, those short messages are quite cryptic, all written in capitals, with many confusing abbreviations. Second, the number of emitted NOTAMs is growing over time with for example 2 million in 2018 (circa 5500 per day). Some of them are crucial for the flight, but the vast majority are of low importance, which makes the analysis difficult.

NLP techniques can be very helpful in that aeronautical context, typically to rank NOTAMs by criticality and highlight important information in them (like runway and taxiway identifiers). For example, NOTAMs about runway or flight area closure are often more relevant than the ones repeated every day by small airports about strong wind in the area.

```
TWY E(BTN H AND Z)-RESTRICTED DUE TO
CONST RMK/NOT AVBL FOR ACFT WITH MORE
THAN 65M

TWY HOTEL CLSD 283M FROM INTERSECTION
WITH TWY GOLF
```

Figure 2: Example of NOTAMs (E field)

2.2 Linguistic point of view

NOTAMs are composed of multiple fields, some of which contain structured information that is easy to parse with a fixed grammar. In this work we focus on the "E field", which usually contains the most detailed information in an unstructured free-text form. NOTAMs (E field) are quite short text messages and are not written in standard English but rather in a domain-specific language, mainly composed of abbreviations and acronyms from the aeronautical world; an official list of acronyms is maintained by ICAO (International Civil Aircraft Organization).

A strong expertise is required to decode and understand those messages; if, overall, English is the main used language, some authorities use their local language. Two examples of NOTAM content (E field) are shown on figure 2.

The NOTAM language is designed to be concise in order to transmit information in the most efficient way. In order for this language to be understood and written by everyone from the aeronautical world, some guidelines exist and it is strongly encouraged to use the official list of acronyms. Such patterns like ""*RWY XX/YY CLSD*"" (which means that the runway "XX/YY" is closed) appear quite often in the NOTAM corpus but despite the official recommendations, people authoring NOTAMs regularly deviate from them, can make spelling mistakes, etc. The resulting NOTAM language thus presents the same challenges as any natural language and cannot be robustly analyzed with a rule-based system.

3 Problem definition

3.1 Criticality prediction

During the preparation of the flight, the pilot and co-pilot must take note of all these documents introduced above. However, as mentioned before, NOTAMs can be very numerous and may not all be relevant for the flight in question. With the criticality estimation, we aim to highlight the most important messages for the flight, to help the pilot optimize the preparation phase.

3.2 Named entity recognition

As mentioned in Section 2, highlighting the most important and relevant entities can help the pilot digest the NOTAMs and focus on the most insightful parts. This is a typical NLP task called Named Entity Recognition (NER).

One crucial information the pilot needs to know is about the closure of airways (*runway* or *taxiway*)[1]. Sometimes, the closure of an airway is specified with :

- a *geographical* condition : for example, only a given part of the airway is closed

[1] A *runway* is where the aircraft lands/takes-off, whereas a *taxiway* is a road connecting runways to terminals and hangars in an airport

- a *temporal* condition : the airway is closed certain days of the week or at certain time schedules of the day
- an *aircraft* condition : for example, an airway can be closed only to aircraft whose wingspan is larger than a given size
- an *operational* condition : for example, an airway can be closed only for take-off or landing

Similarly, exceptions and reasons can be added to further specify the NOTAM.

3.3 Translation

Pilots and co-pilots are often supported by digital apps provided on the so-called electronic flight bag (EFB), a mobile tablet docked to the aircraft, replacing the physical flight bag that used to contain all flight documents in the past. Beyond giving digital access to the required documents, some of these apps now propose to visualize contextual flight information (e.g. extracted from NOTAMs) in a more digestible format for the pilot, such as maps with visual cues. Such apps typically rely on structured machine-parsable languages like Airlang, synthesizing the most important pieces of information from NOTAMs. Today, the translation from raw NOTAMs to Airlang is generally done manually by multiple humans (in charge of the translation itself or its verification). In this paper, we are investigating the possibility to automate this translation using modern sequence-to-sequence language models. The goal is to accelerate this translation task, potentially reducing the manual effort to the verification part only.

4 Experiments and results

4.1 NOTAM language model pretraining

Significant advances in the NLP field have been made in the recent years thanks to powerful Transformer architectures (Vaswani et al., 2017) and self-supervised pretraining, as introduced by the BERT paper (Devlin et al., 2019) and continued in various derivative work like RoBERTa (Liu et al., 2019) or DeBERTa (He et al., 2020). Due to its characteristics, the NOTAM language can benefit from such state-of-the-art language models.

Following popular practices on BERT models and its variants, the idea is to pretrain a language model on many raw NOTAMs with a dedicated tokenizer (we cannot reuse models available online since there is almost no overlap with standard English), and then fine-tune it for each downstream task introduced in section 3. See Appendix B for architecture details.

We experimented with a few language model variants (RoBERTa and DeBERTa v2, both with 6 layers), each being trained on a dataset of 1.2 million unlabeled NOTAMs, from which the E field (the free text part) was extracted. No other preprocessing was performed on the data.

The RoBERTa models were trained on a corpus tokenized using BPE (Sennrich et al., 2016), whereas the DeBERTa ones were trained using SentencePiece (Kudo and Richardson, 2018) tokenization; both tokenization models has a vocabulary size of 52000.

The language models were trained using the Huggingface *transformers* library [2], using a masked language model objective, during 3 epochs.

4.2 Criticality prediction

The objective is to assign a score to each free-text part of a NOTAM (part E), from 1 = lowest priority to 5 = highest priority. In terms of NLP task, this fine-tuning consists in a sequence-level prediction (classification or regression). We train a regression head that takes as input the output of the classification [CLS] token after passing through the pretrained language model. This pooled embedding reflects the context of the full text as it contains information about all the other tokens of the considered sequence. The classification head is mainly a linear layer.

We choose to cast this task as a regression rather than a classification in order to take into account the ranking of the scores. Indeed, classifying a message to 2 or 5 rather than 1 should not give the same loss value. Therefore, the output of our additional head is of size 1.

The dataset comes from ICAO and is composed of circa 35000 NOTAMs annotated by experts. One of its characteristics is its heterogeneity between the labels : more than 10% of the dataset contains duplicated messages to which different scores have been attributed, sometimes even 1 and 5 for the same NOTAM. This reflects a divergence of views that can come from the pilot's perception or the context of the flight. Moreover, as you can see in Figure 3, NOTAMs with the lowest importance are

[2] https://github.com/huggingface/transformers

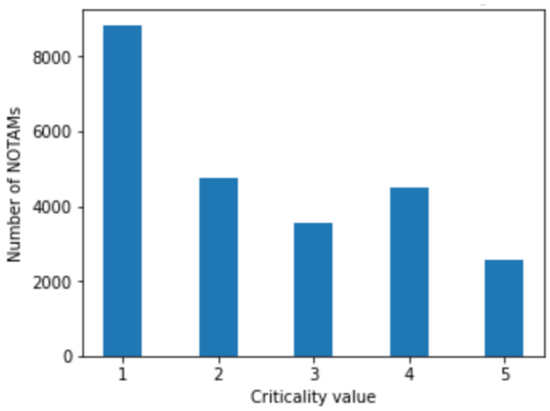


Figure 3: Distribution of the scores in the training set

much more represented than the others.

The dataset is split in 80%-20% between training and testing. By training on this dataset and considering the prediction scores rounded to the nearest integer, we reach a Mean Absolute Error (MAE) of 1.08 in the best case: DeBERTa v2 with hidden size of 768. However, we notice a strong bias towards middle criticality scores with low recalls on the extremes.

To mitigate this point, we can alternatively use a multi-class F1-score as evaluation metric for the best model selection. To tackle the imbalance and heterogeneity problem, we pre-process the dataset by keeping for each NOTAM its most frequently attributed score, followed by an oversampling in order to have the same number of messages for each score in the training set. With these changes, the recall of the lowest and highest criticality NOTAMs are significantly improved (by absolute 28% and 16% respectively). This shows the ability to support the pilot in detecting important NOTAMs, even if it is technically impossible to get perfect predictions on this dataset because of the frequent disagreements between the annotators themselves.

A perspective of improvement would be to calibrate the model with inputs coming from the pilots and human factors team. We may expect that the impact of predicting a low priority when it is actually a high priority message would be larger on the flight's safety than the contrary. An asymmetric loss could then be used during training to reflect these specificities.

4.3 Named entity recognition

NER is the second downstream task studied in this work; it is a classical token classification task that can be implemented by adding, on top of each token's embedding, a linear layer and a softmax to derive the most probable entity tag. The pretrained language model is fine-tuned within this architecture on an annotated dataset.

	Train	Dev	Test
#NOTAMs	196	50	62
runway	231	56	71
taxiway	385	82	97
closure	187	42	57
condition	211	51	42
exception	25	7	9
reason	81	21	26

Table 1: NER dataset description

An extension of this approach was explored by adding a Conditional Random Field (CRF) on top of the linear layer as detailed in (Souza et al., 2019). The biLSTM-CRF (Lample et al., 2016) used to be the state-of-the-art approach before the emergence of BERT-based models; in a sequence labeling task, CRF maximizes the probability of the whole sequence of decisions, so it can better take context into account instead of making independent predictions.

The dataset consists in a set of 308 NOTAMs that were annotated with the different entities in the IOB format (Ramshaw and Marcus, 1995). In this study, the following list of entities are considered: *runway*, *taxiway*, *closure*, *condition*, *exception* and *reason*. The dataset is rather small but annotation is quite costly since it requires aeronautical expertise. The dataset was respectively split into training, development and test sets as detailed in Table 1.

Three different kinds of models were trained. The baseline model is a layered biLSTM model (Ju et al., 2018); it was already explored in the context of NOTAMs in previous work (Arnold et al., 2019). The layered aspect is interesting to tackle NOTAM entities, which can be nested as shown in Figure 4. Indeed, inside the *closure* clause, there can be mentions of other entities like *runway*, *taxiway* but also of *condition*, *exception* or *reason*.

The other approaches presented in this paper are based on the RoBERTa language model (trained from scratch on NOTAM data), fine-tuned on the NER dataset in two variants: without and with a CRF layer. As entities are nested, a first simple approach consists in training a separate model for each kind of entity; as *runway*, *taxiway* cannot be nested in each other, they are covered by one model.

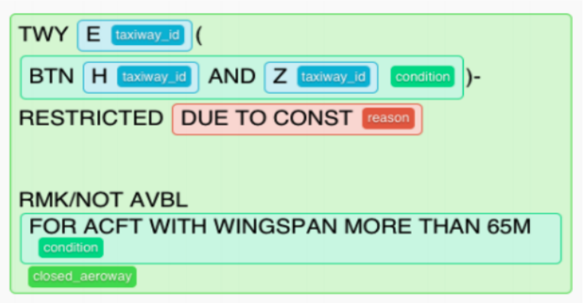

TWY E (
BTN H AND Z)-
RESTRICTED DUE TO CONST
RMK/NOT AVBL
FOR ACFT WITH WINGSPAN MORE THAN 65M

Figure 4: Example of nested entities

Every other entity (*closure*, *condition*, *exception* or *reason*) is covered by its own model.

The results (obtained with the conlleval script [3] with the different models are summed up in Table 2. Using the RoBERTa fine-tuned model without CRF seems to significantly degrade the results globally compared to the biLSTM-CRF baseline on all entities, except for *runway* where the F1-score is improved. The degradation is even more significant for "long-span" entities like *closure* and *reason*. However, the CRF layer seems to boost the F1-scores for all the entities; results are significantly improved compared to the initial baseline. In this context, where entities can have a long span, the CRF layer seems to play a crucial role in this sequence labeling task. Finally, entities like *runway* and *taxiway* reach very high F1-scores; they are the easiest to catch (because often preceded by keywords like "RWY" and "TWY"). The results on other entities are globally lower; they are more difficult to recognize because they have a longer span and are often less represented in the corpus.

As described previously, the RoBERTa fine-tuned models seem to provide good results globally on all the entities but each entity needs its own model due to the nested aspect. This approach is not very efficient both in terms of memory and computing time. This motivated us to explore multitask learning (Caruana, 1997; Collobert and Weston, 2008); the idea is to start with the pretrained RoBERTa model but this time with one dedicated classification head for each entity type. By simultaneously training on all the entities, each task could hopefully benefit from each other. Results are presented in Table 2. The multitask model that handles all the entities at once keep good F1-scores on entities like *runway*, *taxiway* and *closure*, for which we have more examples in the training set, whereas the results are a bit degraded on entities like *condition* and *reason*. The results for *exception* are to be considered carefully because there are too few examples in the training and test sets. This motivated us to train a multitask model only on the *runway*, *taxiway*, *closure* and *condition* entities for which we had at least 200 occurrences in the training set. F1-scores are further improved on *runway*, *taxiway* and *closure* entities. Multitask learning enables recognizing nested entities with a single model, as long as a minimal amount of data is present for each entity type (otherwise, less represented entities tend to penalize the training overall).

4.4 Translation

The last downstream task of interest is the automatic translation from the raw NOTAM text to the Airlang structured language, the latter being parsable by a fixed grammar (see example in Figure 5). This sequence-to-sequence task requires an encoder-decoder model like the original Transformer architecture (Vaswani et al., 2017). For the encoder, we reuse the pretrained model introduced in Section 4.1. For the decoder, we use a similar model (RoBERTa) but initialized from scratch without pretraining because we do not have access to huge amounts of unlabeled Airlang data, as opposed to raw NOTAMs. We then fine-tune the whole encoder-decoder model end-to-end on a dataset of circa 20000 NOTAM-Airlang pairs (translated by human professionals).

NOTAM: `YMMM E1166/20 17JUN0100-17JUN0300 STIRLING AIRSPACE R192ABC ACT (RA2) DUE MILITARY FLYING SFC / FL300`

Airlang: `TIMEDEF DURATION = 17 Jun 2020 1:00 TO 17 Jun 2020 3:00; AREADEF "YM:192A" FL001 TO FL300 ACTIVE DURATION; AREADEF "YM:192B" FL001 TO FL300 ACTIVE DURATION; AREADEF "YM:192C" FL001 TO FL300 ACTIVE DURATION;`

Figure 5: Example of NOTAM translated to Airlang

Although the output sequence is not constrained by any special mechanism, after training we observe that most generated Airlang translations are valid with regard to the grammar underpinning this structured language. As opposed to classical translation tasks where BLEU or ROUGE scores are often used to allow for some flexibility, here the

[3] https://github.com/sighsmile/conlleval/blob/master/conlleval.py

	Layered biLSTM CRF			RoBERTa			RoBERTa CRF			Multitask model			Multitask model w/o *exception/reason*		
Entity	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
runway	95.8	95.8	95.8	97.3	100.0	98.6	98.6	100	**99.3**	98.6	100.0	99.3	98.6	100.0	**99.3**
taxiway	97.7	87.6	92.4	92.6	89.7	91.1	94.9	94.9	**94.9**	95.8	93.8	94.8	96.9	95.9	**96.4**
closure	70.5	78.2	74.1	59.4	74.6	66.1	87.0	72.7	**79.2**	80.8	76.4	78.5	81.8	81.8	**81.8**
condition	55.9	46.3	50.7	30.7	56.1	39.7	63.2	58.5	**60.8**	67.9	46.3	55.1	61.6	58.5	60.1
exception	100.0	33.3	50.0	100.0	22.2	36.4	100.0	33.3	50.0	100.0	22.2	36.4			
reason	87.0	76.9	81.6	73.9	65.4	69.4	91.7	84.6	**88.0**	91.3	80.8	85.7			

Table 2: NER results in terms of Precision, Recall and F1-score

model performance is evaluated (on a test set of circa 5000 NOTAM-Airlang pairs) with a much more conservative metric because of the safety-critical context and the fact that the target language is structured: we consider the percentage of "perfect translations", i.e. the ones matching exactly the ground truth in a case-sensitive way. However we noticed a few tiny variations in this ground truth that are parsed equivalently down the line (optional presence of a white space in certain places, some words that are both valid whether they are capitalized or not, equivalent ways of expressing flight levels like "FL001 TO FLxxx" and "FLxxx AND BELOW"...). So we propose to post-process both the model output and ground truth with simple hard-coded rules to normalize their form, leading to adjusted performance scores which better reflect the actual quality of the translation (see Table 3 for results without/with post-processing using different encoder models).

We note that our system (using the best translation model) is able to produce 84.5% correct translations, which can significantly reduce manual efforts from operational teams providing such services to airlines. To further support these teams by giving a confidence score on these translations, we use gradient boosting (Chen and Guestrin, 2016) to train a classifier in charge of detecting good/bad translations based on various seemingly relevant features (length of the NOTAM, number of occurrences for certain elements like days/months, etc.). As seen in Figure 6, this classifier obtains a AUC score (Area Under Curve) of 0.90, showing a strong ability to distinguish good/bad translations (the business can adapt the threshold to select any point on the curve according to their preferred trade-off between probability of detection vs false alarms).

Encoder model	Hidden size	No post-process	With post-process
RoBERTa	768	74.3%	83.6%
RoBERTa	1536	**78.1%**	**84.5%**
DeBERTa v2	768	78.0%	83.1%
DeBERTa v2	1536	77.3%	82.3%

Table 3: Perfect NOTAM to Airlang translation scores

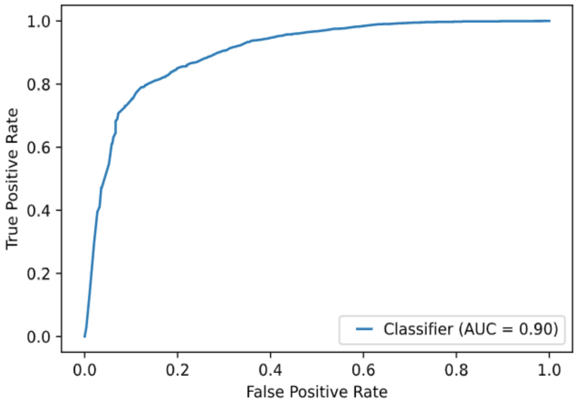


Figure 6: Translation classifier AUC score

5 Conclusion and perspectives

In this work, we presented the use of modern self-supervised language models (derived from BERT) to extract knowledge from NOTAM aeronautical messages. We showed that a single deep learning model pretrained on circa 1 million unlabelled NOTAMs can be efficiently reused on downstream tasks with dedicated fine-tuning. NOTAM criticality prediction can support pilots during their pre-flight briefing by highlighting the most important messages. Furthermore, named entity recognition can be applied to extract relevant parts of NOTAMs (e.g. closed runways/taxiways, specific conditions/exceptions...). Finally, automatic translation to a domain-specific structured language (Airlang) used by pilot apps during flight, can support operational teams providing services to airlines. The evaluation scores on these tasks show a high potential for an operational usability of such models (by pilots, airlines or service providers), which is a first to the best of our knowledge.

In the future, alternative NLP methods such as summarization for NOTAMs could be explored to continue reducing pilots' workload. While the use of deep learning networks (and pretrained language models) enabled increasing accuracy in a lot of NLP downstream tasks, they are known to be overconfident in their predictions. It is an issue in the aviation context given its safety-critical nature, where trust in systems' predictions is key. In that respect, uncertainty quantification methods - such as conformal predictions (Vovk V. and Shafer, 2005)(Angelopoulos and Bates, 2021) - could give a reliable measure of confidence in model's outputs. The robustness of the model could also be assessed through adversarial attacks, as in (Morris et al., 2020). Finally, formal methods could be used for verification and could pave the way to the certification of such deep learning models, required for any use on board.

Ethical considerations

In any safety-critical context like aeronautics, there is an inherent risk associated with the use of automatic methods supporting human operators. This is why our proposed techniques are limited to a responsible use on ground, at least until the underlying models can be certified for in-flight use thanks to rigorous methods from the *Trusted Artificial Intelligence* research field. In any case, such systems are only meant to support human analysis and decision making by decreasing workload, not to replace them.

The NOTAMs collected worldwide and used in this study are public data (accessible via numerous official platforms online). The datasets used for named entity recognition and translation are proprietary and built internally by expert annotators as part of their work (with the permission to be used in our work). The ICAO dataset used for criticality prediction is public and enables research use.

The Huggingface Transformers framework supporting model training in our study is open sourced under the permissive Apache 2.0 license. Every model training mentioned in this paper (RoBERTa and DeBERTa v2 ones) took less than 12 GPU hours for pretraining and for each of the three downstream tasks. The hyperparameters used by these models in our experiments are the default ones from the Transformers library (faithful to the original papers), except when explicitly mentioned (e.g. varying the hidden size).

References

Anastasios N. Angelopoulos and Stephen Bates. 2021. A gentle introduction to conformal prediction and distribution-free uncertainty quantification.

Alexandre Arnold, Gérard Dupont, Catherine Kobus, François Lancelot, and Pooja Narayan. 2019. Interprétation et visualisation contextuelle de NOTAMs (messages aux navigants aériens) (). In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Volume IV : Démonstrations*, pages 639–643, Toulouse, France. ATALA.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459, New Orleans, Louisiana. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

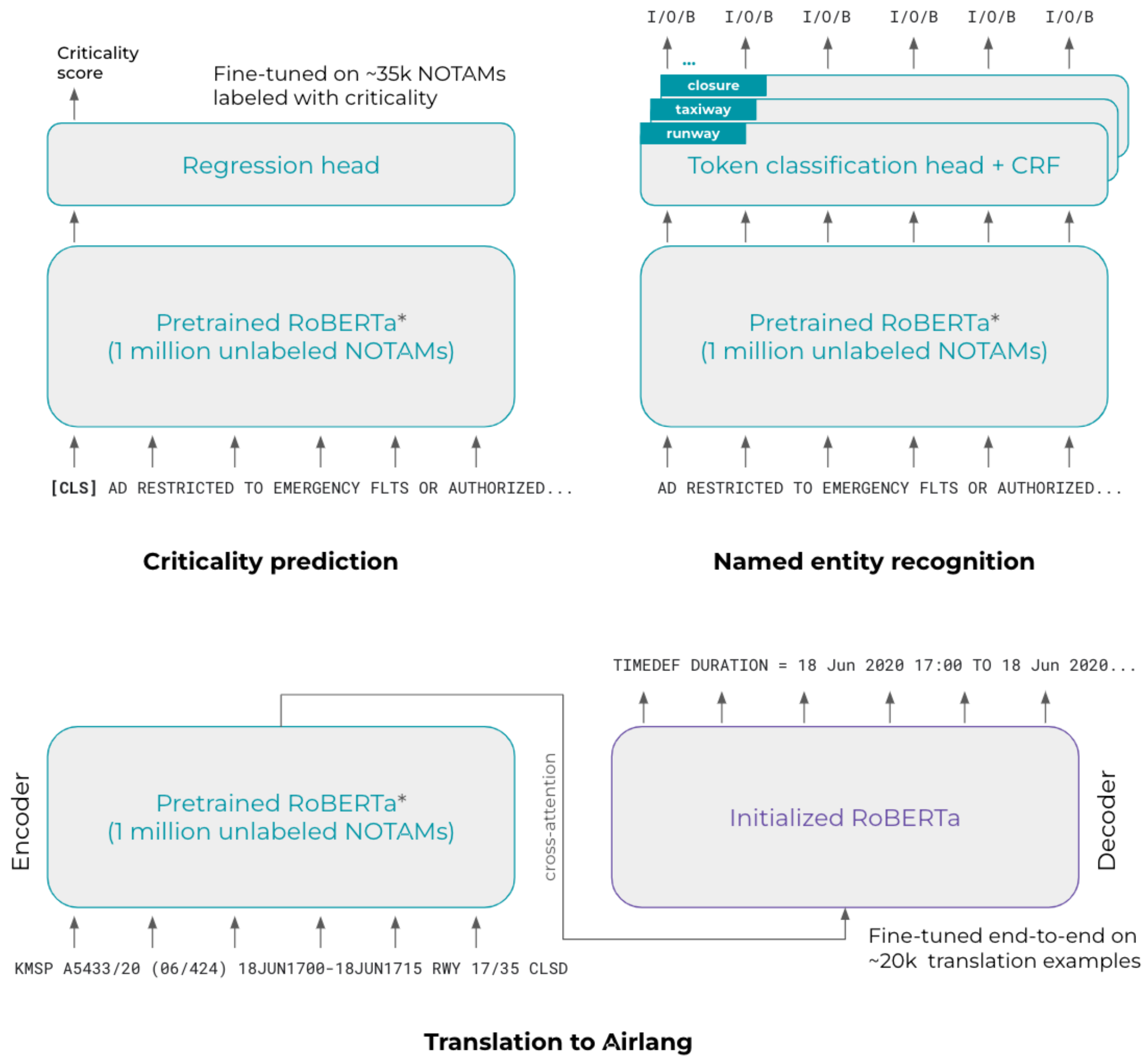
Gammerman A. Vovk V. and G. Shafer. 2005. *Algorithmic Learning in a Random World*. Springer.

A Appendix: NOTAM details (ICAO format)

A NOTAM message is structured into 5 or 6 different fields, namely:

- Q field is the Qualifier line; it contains a series of classification tags that the operator is supposed to fill while authoring the NOTAM
- A field is the ICAO indicator of the aerodrome or the FIR (Flight Information Region)
- B field corresponds to the date/time when this NOTAM becomes effective
- C field corresponds to the date/time when the NOTAM ceases to be effective
- D field (optional) can specify a miscellaneous diurnal time for the NOTAM if the hours of effect are less than 24 hours a day
- E field contains the NOTAM text message (the free text part), the part of interest for our study

B Appendix: Model architectures for the three downstream tasks


Criticality score
Fine-tuned on ~35k NOTAMs labeled with criticality
Regression head
Pretrained RoBERTa*
(1 million unlabeled NOTAMs)
[CLS] AD RESTRICTED TO EMERGENCY FLTS OR AUTHORIZED...
I/O/B I/O/B I/O/B I/O/B I/O/B I/O/B
...
closure
taxiway
runway
Token classification head + CRF
Pretrained RoBERTa*
(1 million unlabeled NOTAMs)
AD RESTRICTED TO EMERGENCY FLTS OR AUTHORIZED...
TIMEDEF DURATION = 18 Jun 2020 17:00 TO 18 Jun 2020...
Encoder
Pretrained RoBERTa*
(1 million unlabeled NOTAMs)
cross-attention
Initialized RoBERTa
Decoder
Fine-tuned end-to-end on ~20k translation examples
KMSP A5433/20 (06/424) 18JUN1700-18JUN1715 RWY 17/35 CLSD

Criticality prediction

Named entity recognition

Translation to Airlang

* The same pretrained model can be shared for the three tasks