Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles
Avignon, France, 27 juin au 1er juillet 2022

Extraction d'informations de messages aéronautiques (NOTAMs) avec des modèles de langue appris de façon auto-supervisée

Alexandre Arnold, Fares Ernez, Catherine Kobus, Marion-Cécile Martin
Airbus AI Research,
prénom.nom@airbus.com

RÉSUMÉ

Avant un vol, les pilotes de ligne doivent lire une longue liste de messages appelés NOTAM (pour NOtice To AirMen) donnant des informations sur des aléas potentiels le long du vol. Ces messages suivent une grammaire particulière, contiennent beaucoup d'acronymes et un vocabulaire spécifique aéronautique. Dans cet article, un modèle de langue de type BERT est pré-entraîné sur un grand nombre de ces messages ; il est ensuite affiné sur trois tâches : l'estimation de criticité, la reconnaissance d'entités nommées et la traduction vers un langage structuré appelé Airlang. L'apprentissage auto-supervisé, permettant de tirer parti du vaste nombre de données non annotées, est particulièrement intéressant dans le domaine aéronautique, pour lequel les annotations sont très coûteuses car nécessitant une forte expertise. Nous montrons les résultats encourageants sur les trois tâches.

ABSTRACT

Information extraction from aeronautical messages

During their pre-flight briefings, aircraft pilots must read a long list of NoTAMs (NOtice To AirMen) indicating potential hazards along the flight route. NOTAM language has a very special phrasing, with lots of acronyms and domain-specific vocabulary. In this paper, we pretrain language models derived from BERT on lots of unlabeled NOTAMs and reuse the learnt representations on three downstream tasks : criticality prediction, named entity recognition and translation into a structured language called Airlang. This self-supervised approach, that leverages the huge amount of unannotated data, is well suited in the aeronautical context since expert annotations are expensive. We present evaluation scores across the tasks showing a high potential for an operational usability of such models.

MOTS-CLÉS : NOTAM, Apprentissage auto-supervisé, Classification, REN, Traduction.

KEYWORDS: NOTAM, Self Supervised Learning, Classification, NER, Translation.

1 Introduction

Chaque vol nécessite une préparation pendant laquelle les pilotes prennent connaissance de tous les éléments importants concernant le vol, à propos des conditions météorologiques, des réserves de carburant ou encore des notifications portant sur la sécurité. Les pilotes les reçoivent de la part des agences gouvernementales de contrôle de la navigation aérienne sous la forme de messages écrits, appelés NOtice To AirMen (NOTAM - message aux navigants). De nombreux messages sont à lire avant le vol (parfois jusqu'à plus de 100 pages pour les long-courriers). Ils sont écrits majoritairement en anglais avec de nombreux acronymes et mots techniques ainsi qu'une syntaxe particulière.

Volume 1 : conférence principale, pages 335–344.

Des travaux ont déjà été menés sur les NOTAMs, par exemple dans un but de classification (Mi *et al.*, 2022). Dans cet article, nous appliquons les dernières avancées en TAL au domaine aéronautique dans le but de réduire la charge de travail des pilotes. Un modèle de langue de type BERT met à profit la vaste quantité de données non annotées et permet après affinage de réaliser trois tâches : l'estimation de la criticité, la reconnaissance d'entités nommées et la traduction vers un langage structuré appelé Airlang. Cet article est organisé comme suit : la partie 2 décrit les NOTAMs et les différentes tâches. La partie 3 détaille les approches et les différents résultats obtenus.

2 Contexte aéronautique

2.1 Qu'est-ce qu'un NOTAM ?

Un NOTAM est un message émis par les autorités aéronautiques pour alerter les pilotes d'éventuels dangers le long du trajet. Il informe notamment des perturbations temporaires (de quelques heures à un an maximum) sur des infrastructures (comme une fermeture ou usage restreint d'une piste d'atterrissage), d'exercices militaires ayant pour conséquence un espace aérien restreint, de la présence temporaire d'obstacles près des aéroports, etc. Un exemple de NOTAM est montré à la Figure 1.

```
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 – Exemple de NOTAM avec ses différents champs (Q, A, B, C et E). Le champ E explique qu'en raison de travaux ("DUE WIP"), la portion Sud de la *taxiway* 'B', entre les *taxiways* 'F' et 'R' esrt fermée, ainsi que la portion de la *taxiway* 'R' entre les *taxiways* 'A' et 'S'. Le NOTAM précise la route à utiliser en remplacement.

Avant un vol, les pilotes doivent lire les NOTAMs pertinents pour ce vol afin de garantir la sécurité de ce dernier ; cette tâche s'avère longue et ardue. En effet, ces messages sont écrits dans un langage non standard et avec un nombre important d'abréviations. De plus, avec l'augmentation du nombre d'aéroports ces dernières décennies, ainsi que leur propension à alerter du moindre risque pour éviter toute responsabilité, le nombre de NOTAMs émis est de plus en plus importants (en 2018, en moyenne 5500 NOTAMs publiés par jour). Parmi ces NOTAMs, certains sont cruciaux pour le vol mais une grande majorité sont de faible importance.

Les technologies de TAL peuvent être très utiles pour, par exemple, classer les NOTAMs par ordre décroissant de criticité ou pour souligner dans les NOTAMs les informations importantes (comme par exemple les identifiants de *runway* ou *taxiway*). Les NOTAMs décrivant une fermeture de *runway* ou d'un secteur aérien pour raison d'exercices militaires, sont d'importance cruciale par rapport à ceux apparaissant tous les jours publiés par de petits aéroports à propos de vents violents dans la région.

Les NOTAMs sont composés de plusieurs champs ; certains sont structurés et donc faciles à analyser. Dans cette étude, nous nous focalisons sur le champ "E", qui est un champ "libre" très informatif, non structuré, écrit dans un Anglais non standard, comprenant beaucoup d'abréviations et d'acronymes connu du monde aéronautique. Une forte expertise est requise pour décoder ce type de messages.

Le langage NOTAM a été mis au point afin d'être concis et de pouvoir transmettre l'information de façon la plus efficace possible. Afin que ce langage puisse être écrit par tous et compris de tous, des recommandations d'écriture existent et il est fortement recommandé d'utiliser la liste officielle des acronymes. Malgré ces recommandations, les déviations des auteurs sont fréquentes en pratique. Il en résulte un langage non contrôlé présentant les mêmes défis que le langage naturel et qui serait très difficile de couvrir de manière robuste par un système de règles.

2.2 Définition du problème

2.2.1 Prédiction de criticité

Avant un vol, les pilotes doivent prendre connaissance des NOTAMs ; cette tâche s'avère fastidieuse car pour un vol long-courrier, il peut s'agir de plus de 100 pages de NOTAMs d'importance inégale. L'estimation de la criticité d'un NOTAM permettrait de mettre en évidence les messages les plus importants pour le vol.

2.2.2 Reconnaissance d'entités

Mettre en valeur les mots importants dans un NOTAM peut aider le pilote à se focaliser sur les parties les plus cruciales. C'est une tâche classique de TAL appelée Reconnaissance d'Entités Nommées (REN). Une des informations cruciales trouvée dans les NOTAMs est à propos de la fermeture de certaines *runway* ou *taxiway* [1]. La fermeture d'une piste peut être accompagnée d'une raison, de conditions ou d'exceptions (par exemple, une partie de la piste est fermée, ou uniquement certains jours de la semaine, ou uniquement pour le décollage, etc.).

2.2.3 Traduction

Les pilotes sont souvent assistés par des applications numériques accessibles sur l'Electronic Flight Bag (EFB), une tablette remplaçant la sacoche de vol physique qui contenait tous les documents de vol dans le passé. Certaines applications proposent désormais de visualiser des informations de vol contextuelles (par exemple extraites de NOTAMs) dans un format plus digeste pour le pilote, comme des cartes avec des repères visuels. Ces applications s'appuient généralement sur des langages structurés comme Airlang. La traduction des NOTAMs bruts vers Airlang est généralement effectuée manuellement par plusieurs personnes. Nous étudions la possibilité d'automatiser cette traduction en utilisant des modèles de langue "séquence à séquence".

3 Expériences et résultats

3.1 Entraînement d'un modèle de langue NOTAM

Le domaine du TAL a connu une révolution grâce à l'architecture de type *Transformer* (Vaswani *et al.*, 2017) et à l'apprentissage auto-supervisé introduit dans (Devlin *et al.*, 2019). Dans cette étude,

1. Une *runway* est la piste de décollage/atterrissage ; la *taxiway* est la piste connectant les *runways* aux terminaux

nous pré-entraînons un modèle de langue sur une grande quantité de NOTAMs et nous affinons ce modèle sur trois tâches de TAL décrites dans la Section 2.2. Comme ce langage n'a presque aucun point commun avec l'anglais, le pré-entraînement se fait de zéro avec un tokeniseur dédié.

Plusieurs variantes de modèle de langue ont été entraînées (RoBERTa (Liu *et al.*, 2019) et DeBERTa v2 (He *et al.*, 2020), chacune avec 6 couches) sur un corpus de 1.2 millions de NOTAMs. Les NOTAMs sont des données publiques et ont pu être collectés via une plateforme en ligne. Les modèles de type RoBERTa et DeBERTa sont entraînés à partir d'un corpus tokenisé respectivement avec BPE (Sennrich *et al.*, 2016) et SentencePiece (Kudo & Richardson, 2018). Chaque tokeniseur a une taille de vocabulaire de 52000. Les modèles de langue ont été appris (sur 3 époques) en utilisant la librairie *transformers* d'Huggingface [2].

3.2 Prédiction de criticité

L'objectif est d'assigner un score au champ E d'un NOTAM, de 1 (priorité la plus basse) à 5 (priorité la plus haute). Nous entraînons une tête de régression dont la donnée d'entrée est la sortie du token de classification [CLS] de la séquence, après son passage à travers le modèle de langue pré-entraîné. Nous avons choisi de traiter cette tâche en tant que régression plutôt que classification pour conserver l'idée de classement entre les scores. En effet, attribuer une criticité de 2 ou 5 à un message, alors que les experts lui ont donné une criticité de 1 ne devrait pas donner la même erreur.

Le jeu de données provient de l'OACI (Organisation de l'Aviation Civile Internationale) ; il est public, utilisable à des fins de recherche, et se compose d'environ 35000 NOTAMs annotés par des experts. Il présente notamment une grande hétérogénéité entre les labels : plus de 10% du jeu de données contient des messages auxquels différents scores ont été attribués, parfois 1 et 5 pour le même NOTAM. Cela atteste de la divergence de points de vue d'un pilote à l'autre. Les trois quarts de ces messages dupliqués le sont avec deux différents scores, probablement venant de deux experts. Pour le quart restant, les messages ont été annotés trois fois ou plus. Par ailleurs, comme montre la Figure 2 , les NOTAMs avec le moins d'importance sont bien plus représentés que les autres. La répartition entre corpus d'entraînement et de test est de 80%/20%.

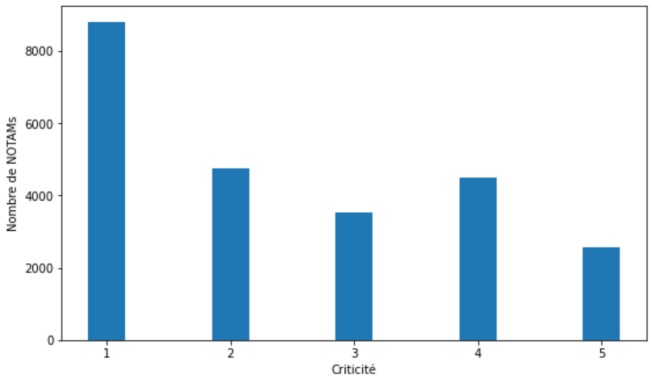

FIGURE 2 – Distribution des scores dans le corpus d'apprentissage

Le meilleur résultat est une erreur absolue moyenne de **1.08** obtenue avec DeBERTa v2 (taille couche cachée de 768). Les scores de prédiction sont arrondis à l'entier le plus proche. Cependant, nous

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

observons un fort biais en faveur des criticités moyennes avec un rappel très faible pour les extrêmes ; pour y pallier, une alternative est d'utiliser la F-mesure multi-classe comme métrique pour sélectionner le meilleur modèle. En outre, afin de résoudre le problème de déséquilibre et d'hétérogénéité du corpus, nous avons gardé, pour chaque NOTAM, le score le plus fréquemment attribué ou son unique score en cas d'annotation unique. Nous procédons ensuite à un suréchantillonnage afin que chaque score soit représenté par le même nombre de messages dans le jeu d'entraînement. Les rappels de la criticité la plus haute et la plus basse sont ainsi améliorés significativement, respectivement par **28%** et **16%** en absolu. Ces résultats sont encourageants bien qu'il semble impossible d'obtenir des prédictions parfaites sur ce jeu de données présentant un fort désaccord entre annotateurs.

Une perspective d'amélioration serait d'intégrer les connaissances venant des pilotes et de l'équipe facteurs humains dans la calibration du modèle. Nous pouvons nous attendre à un impact plus fort sur la sécurité du vol en cas de prédiction d'une faible criticité pour un message en réalité très important, qu'en cas de surévaluation de la criticité d'un message. Une fonction de perte asymétrique pourrait alors être utilisée pendant l'entraînement pour tenir compte de ces spécificités.

3.3 Reconnaissance d'entités nommées

La reconnaissance d'entités nommées est implémentée en ajoutant, au plongement de mots de chaque token dérivé du modèle de langue, une couche linéaire et un *softmax* afin de déduire le label de l'entité la plus probable. Le modèle de langue pré-entraîné est affiné sur un corpus annoté en entités. Une couche de type CRF (Conditional Random Field) peut être ajoutée au-dessus de la couche linéaire, comme décrit dans (Souza *et al.*, 2019) ; elle maximise la probabilité de toute la séquence de décisions, ce qui permet de mieux prendre en compte le contexte et les décisions précédentes. Avant l'émergence des modèles dérivés de BERT, l'état de l'art consistait en des modèles de type biLSTM-CRF (Lample *et al.*, 2016).

Le jeu de données utilisé dans cette étude est constitué de 308 NOTAMs. C'est un jeu de données propriétaire, créé en interne. Les entités *runway*, *taxiway*, *fermeture*, *condition*, *exception* et *raison* sont considérées ; un exemple de ces types d'entités est donné à la Figure 3. Le jeu de données est assez petit, l'annotation étant assez coûteuse dans le domaine aéronautique car requérant une expertise métier. Plus de détails sur le corpus utilisé sont donnés dans la Table 1.

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

TABLE 1 – Description du corpus utilisé pour la tâche de reconnaissance d'entités

Trois types de modèles ont été entraînés et testés. Le modèle initial est un modèle biLSTM-CRF à base de couches (Ju *et al.*, 2018) ; cette approche avait été déjà testée pour les NOTAMS dans une précédente étude (Arnold *et al.*, 2019). L'aspect "couches" de ce modèle est intéressant pour traiter les entités imbriquées des NOTAMs ; comme le montre la Figure 3. En effet, à l'intérieur de la clause

fermeture, il peut être fait mention d'autres entités comme *runway*, *taxiway* mais aussi de *condition*, *exception* ou *raison*.

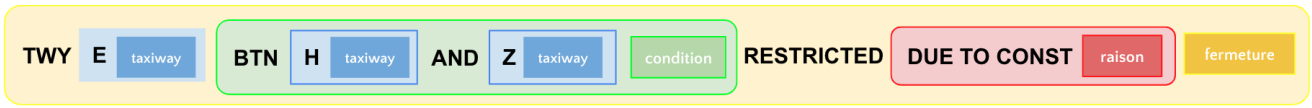

FIGURE 3 – Exemple de NOTAM avec des entités imbriquées de type *taxiway* H et Z, imbriquées dans l'entité *condition*, elle-même imbriquée dans l'entité *fermeture*. Ce NOTAM informe que la portion de *taxiway* E, située entre les *taxiways* H et Z est fermée à cause de travaux/constructions

Les autres approches étudiées dans cette études sont basées sur un modèle de langue de type RoBERTa, qui a été réentraîné de zéro sur le langage "NOTAM" et qui a ensuite été affiné pour une tâche de REN. Deux variantes ont été testées : un modèle sans couche CRF et un autre modèle avec une couche CRF. Comme les entités NOTAMs peuvent être imbriquées (cf. Figure 3), l'approche la plus simple consiste à entraîner un modèle pour chaque type d'entité ; les entités de type *runway* et *taxiway* ne peuvant pas être imbriquées l'une dans l'autre, elles sont couvertes par un seul modèle. Chaque autre type d'entité est couvert par son propre modèle.

Les résultats obtenus avec les différents modèles sont détaillés dans la Table 2. L'utilisation d'un modèle de type RoBERTa (sans couche CRF) dégrade de façon significative les résultats par rapport à la précédente approche à base de biLSTM-CRF sur toutes les entités, sauf pour l'entité de type *runway* pour laquelle la F-mesure est améliorée. La dégradation est encore plus significative pour les entités "longues" comme *fermeture* et *raison*.

	biLSTM CRF à couches			**RoBERTa**			**RoBERTa CRF**			**Modèle multi-tâches**			**Modèle multi-tâches sans *exception/raison***		
Entité	Préc.	Rappel	F1	Préc.	Rappel	F1	Préc.	Rappel	F1	Préc.	Rappel	F1	Préc.	Rappel	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**
fermeture	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			
raison	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 – Résultats pour la tâche de REN en termes de F-mesure, Précision et Rappel (en %)

L'ajout de la couche CRF permet d'améliorer significativement les résultats par rapport à l'approche initiale à base de biLSTM-CRF. La couche CRF améliore, comme attendu, les résultats pour les entités "longues" car elle permet de sortir une séquence d'étiquettes valide.

Les entités de type *runway* et *taxiway* obtiennent de très hautes F-mesures ; ce sont en effet les entités les plus faciles à détecter car souvent précédées de mots-clé spécifiques comme "RWY" et "TWY". Les résultats sur les autres entités sont globalement moins bons ; ce sont des entités plus longues donc plus difficiles à reconnaître entièrement et elles sont aussi moins représentées dans notre corpus.

Les entités pouvant être imbriquées, chaque entité nécessite son propre modèle, ce qui n'est pas efficace ni en terme de mémoire ni en temps de calcul. Cela nous a motivés à explorer l'apprentisage multi-tâches (Caruana, 1997; Collobert & Weston, 2008). L'idée est d'ajouter, en parallèle, une tête de classification pour chaque entité ; un seul modèle permettrait de couvrir toutes les entités et l'entraînement joint pourrait être bénéfique pour chacune d'entre elles. Les résultats sont présentés

dans la Table 2. Le modèle multi-tâches obtient de très bons scores F1 pour *runway*, *taxiway* et *fermeture*, pour lesquelles plus de données annotées sont disponibles, tandis que les scores sont dégradés pour *condition* et *raison*. Ces résultats nous ont motivés à entraîner un nouveau modèle multi-tâches, mais en ne considérant que les entités ayant au moins 200 occurrences dans le corpus, i.e. *runway*, *taxiway*, *fermeture* et *condition*. Les scores F1 sont encore améliorés pour ces entités. L'apprentissage multi-tâches permet donc de détecter efficacement les entités imbriquées, à condition d'avoir suffisamment d'annotations. Les résultats pour *exception* et leurs fortes fluctuations sont à considérer avec précaution étant donné ses faibles occurrences. L'apprentissage multi-tâches permet donc de reconnaître un ensemble d'entités imbriquées dans un seul modèle. Une quantité d'annotations suffisante semble cependant nécessaire ; les expériences menées tendent à montrer que les entités faiblement "dotées" (en termes d'annotations) pénalisent globalement l'apprentissage global sur les autres entités.

3.4 Traduction

La dernière tâche est la traduction du texte NOTAM vers le langage structuré Airlang (voir Figure 4), sur la base d'un jeu de données propriétaire créé en interne par des experts dans le cadre de leur travail, avec la permission de les utiliser dans notre étude de recherche. Cette tâche "séquence à séquence" nécessite un modèle d'encodeur-décodeur comme l'architecture originale du *Transformer* (Vaswani *et al.*, 2017). L'encodeur est le modèle pré-entraîné introduit dans la Section 3.1. N'ayant pas eu accès à d'importantes quantités de données Airlang, nous utilisons, pour le décodeur, un modèle similaire (architecture RoBERTa) mais initialisé à partir de zéro sans entraînement préalable. L'ensemble du modèle encodeur-décodeur est affiné sur environ 20000 paires NOTAM-Airlang (traduites par des professionnels humains).

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 4 – Exemple de NOTAM traduit vers Airlang

Bien que la séquence de sortie ne soit contrainte par aucun mécanisme particulier, la plupart des traductions Airlang générées respectent la grammaire qui sous-tend ce langage structuré. Les systèmes de traduction sont souvent évalués en termes de scores BLEU ; étant donné le contexte de sécurité critique et l'aspect structuré du langage cible, le système est ici évalué, en étant sensible à la casse, en termes de pourcentage de "traductions parfaites". Nous avons cependant remarqué quelques infimes variations dans cette vérité terrain qui n'ont aucun impact sur la lecture machine qui en découle (présence facultative d'un espace à certains endroits, des mots analysés de la même manière qu'ils soient en majuscules ou non, des façons équivalentes d'exprimer des niveaux de vol comme "FL001 TO FLxxx" et "FLxxx AND BELOW"...). La sortie du modèle et la vérité terrain ont été post-traitées et normalisées à l'aide de règles simples ; il en résulte des scores reflétant mieux la qualité réelle du système (cf. Table 3). Le meilleur système est capable de produire 84.5% de traductions correctes, ce qui peut réduire considérablement l'effort humain des équipes opérationnelles effectuant ces traductions.

Pour accompagner davantage ces équipes en donnant un score de confiance sur ces traductions, nous

utilisons le gradient boosting (Chen & Guestrin, 2016) pour entraîner un classifieur chargé de détecter les bonnes/mauvaises traductions en fonction de diverses caractéristiques se montrant pertinentes (longueur du NOTAM, nombre d'occurrences pour certains éléments comme les jours/mois, etc.). Ce classificateur obtient un score AUC (aire sous la courbe) de $0,90$, démontrant une forte capacité à distinguer les bonnes/mauvaises traductions (le seuil peut être ajusté en opération pour sélectionner n'importe quel point sur la courbe selon le compromis préféré entre probabilité de détection et fausses alarmes).

Encodeur	Taille cachée	Sans post-traitement	Avec post-traitement
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 – Pourcentages de traduction parfaite du texte NOTAM vers le langage structuré Airlang

4 Conclusion et perspectives

Cet article décrit l'utilisation de modèles de langue auto-supervisés (dérivés de BERT) dans le but d'extraire des connaissances des messages aéronautiques appelés NOTAMs. Un modèle d'apprentissage profond pré-entraîné sur environ 1 million de NOTAMs peut être réutilisé pour accomplir différentes tâches, en l'affinant de manière adéquate. La prédiction de la criticité peut aider les pilotes pendant leur phase de préparation en mettant en évidence les messages les plus importants. La reconnaissance d'entités nommées permet d'extraire les éléments les plus pertinents d'un NOTAM (par exemple, la fermeture de pistes, des conditions ou contraintes spécifiques, ...). Enfin, la traduction automatique vers un langage structuré propre au domaine (Airlang) peut aider les équipes opérationnelles fournissant des services aux compagnies aériennes. Les résultats obtenus sur ces tâches montrent un haut potentiel pour un usage opérationnel.

Les techniques basées sur de l'apprentissage profond et des modèles de langue de type BERT ont permis d'améliorer significativement les performances de système de TAL mais elles sont souvent trop confiantes sur leurs prédictions. C'est un problème dans le contexte aéronautique où il est essentiel d'avoir confiance dans les sorties des différents systèmes. C'est pourquoi des techniques de quantification d'incertitude - comme les prédictions conformes (Vovk V. & Shafer, 2005; Angelopoulos & Bates, 2021) - doivent être explorées car elles pourraient donner une mesure de confiance fiable sur les sorties des modèles. Des méthodes formelles pourraient être également utilisées pour la vérification, ce qui serait une première étape vers la certification des modèles d'apprentissage profond, obligatoire pour une utilisation embarquée dans le cockpit.

Références

ANGELOPOULOS A. N. & BATES S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. DOI : 10.48550/ARXIV.2107.07511.

ARNOLD A., DUPONT G., KOBUS C., LANCELOT F. & NARAYAN P. (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, p. 639–643, Toulouse, France : ATALA.

CARUANA R. (1997). Multitask learning. *Machine Learning*, **28**(1), 41–75.

CHEN T. & GUESTRIN C. (2016). Xgboost : A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, p. 785–794.

COLLOBERT R. & WESTON J. (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, p. 160–167, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/1390156.1390177.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (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)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

HE P., LIU X., GAO J. & CHEN W. (2020). Deberta : Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv :2006.03654*.

JU M., MIWA M. & ANANIADOU S. (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)*, p. 1446–1459, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1131.

KUDO T. & RICHARDSON J. (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*, p. 66–71, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/D18-2012.

LAMPLE G., BALLESTEROS M., SUBRAMANIAN S., KAWAKAMI K. & DYER C. (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*, p. 260–270, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1030.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized bert pretraining approach. DOI : 10.48550/ARXIV.1907.11692.

MI B., FAN Y. & SUN Y. (2022). NOTAM text analysis and classification based on attention mechanism. *Journal of Physics : Conference Series*, **2171**(1), 012042. DOI : 10.1088/1742-6596/2171/1/012042.

SENNRICH R., HADDOW B. & BIRCH A. (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)*, p. 1715–1725, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/P16-1162.

SOUZA F., NOGUEIRA R. & LOTUFO R. (2019). Portuguese named entity recognition using bert-crf. DOI : 10.48550/ARXIV.1909.10649.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in neural information processing systems*, p. 5998–6008.

VOVK V. G. A. & SHAFER G. (2005). *Algorithmic Learning in a Random World*. Springer.