

Interprétation et visualisation contextuelle de NOTAMs (messages aux navigants aériens)

Alexandre Arnold¹ Gérard Dupont¹ Catherine Kobus¹ François Lancelot¹ Pooja Narayan²

(auteurs dans l'ordre alphabétique)

¹ Airbus AI Research, Toulouse, France

² Airbus AI Research, Ottobrun, Allemagne

Résumé

Dans cet article, nous présentons une démonstration de visualisation de l'information extraite automatiquement de la partie textuelle des NOTAMs. Dans le domaine aéronautique, les NOTAMs sont des messages publiés par les agences gouvernementales de contrôle de la navigation aérienne. Nous détaillons la construction du jeu de données, les expériences d'extraction d'information par apprentissage profond (approche et résultats), ainsi que le lien avec la visualisation contextuelle sur des cartes d'aéroports.

Mots Clef

NOTAM, extraction d'information, reconnaissance d'entités imbriquées, biLSTM-CRF, création d'un dataset.

Abstract

In this article, we present a demonstration about the visualization of information automatically extracted from the NOTAMs' textual part. In the aeronautical domain, NOTAMs are messages published by governmental aviation authorities for air traffic control. We detail the dataset construction, the experiments to extract information with deep learning (approach and results), as well as the link with the contextual visualization on airport maps.

Keywords

NOTAM, information extraction, nested NER, biLSTM-CRF, dataset creation.

1 Introduction

Un NOTAM (de l'anglais, Notice to Airmen, "messages aux navigants aériens") peut contenir un certain nombre de méta-données mais sa description complète (champ E) est un champ en texte libre, souvent en anglais avec de très nombreuses abréviations. C'est sur l'interprétation et la visualisation de ce champ que se focalise notre démonstration. Certains organismes (comme ICAO ou Eurocontrol) encouragent l'utilisation de vocabulaire [1] et grammaire [2] pour la rédaction des NOTAMs, mais en pratique ces consignes sont peu respectées. Ainsi l'analyse, le filtrage et le requêtage intelligent des NOTAMs est un problème complexe non résolu [3]. Notre approche vise à ex-

ploiter les progrès significatifs récents en compréhension du langage naturel grâce aux réseaux neuronaux profonds pour extraire des informations clés des NOTAMs.

Dans la partie 2, nous décrivons en détails ce qu'est un NOTAM dans le monde aéronautique mais détaillons aussi ses spécificités en terme linguistique. Nous détaillons ensuite (partie 3) les informations que l'on souhaite extraire de ces messages ainsi que l'approche choisie à base de reconnaissance d'entités imbriquées. La partie 4 décrit le dataset créé (ainsi que le protocole d'annotation suivi) dans le contexte de cette étude. La section 5 décrit en détails les expériences faites ainsi que les résultats obtenus. La partie 6 décrit la démonstration; enfin, sont abordées quelques perspectives d'amélioration du système dans la section 7.

2 Qu'est qu'un NOTAM ?

Un NOTAM est un message publié par les agences gouvernementales de contrôle de la navigation aérienne dans le but d'informer les pilotes d'évolutions, à caractère temporaire (allant de quelques heures à un an maximum) sur les infrastructures (fermeture/limitation d'une *runway* ou *taxiway*, présence d'obstacles, ou tout potentiel danger pour la navigation aérienne); un NOTAM peut également définir des zones interdites de survol.

2.1 Point de vue opérationnel

Un pilote doit obligatoirement prendre connaissance des NOTAMs lors de la préparation d'un vol afin d'assurer une sécurité maximale tout au long de son voyage.

```
- TWY E (BTN H AND Z) -RESTRICTED DUE TO  
CONST  
RMK/NOT AVBL FOR ACFT WITH WINGSPAN  
MORE THAN 65M  
- TWY HOTEL CLSD 283M FROM INTERSECTION  
WITH TWY GOLF
```

FIGURE 1 – Exemples de NOTAMs

La lecture de ces NOTAMs peut vite se révéler ardue car avant chaque vol, le pilote se voit remettre des pages de

messages "cryptiques" écrits en lettre capitale, sans forcément être trié par degré de criticité. Ce qui peut être préjudiciable à la sécurité du vol si le pilote néglige un message critique (par exemple, dans le cas d'une piste d'atterrissage ou d'une zone de vol fermée).

2.2 Point de vue linguistique

Les NOTAMs sont des messages de texte courts ; ils ne sont pas écrits dans un Anglais standard correct mais dans un langage "cryptique", contenant presque exclusivement des abréviations ou acronymes (une liste officielle est maintenue par l'ICAO, Organisation de l'Aviation Civile Internationale [1]), et spécifiques au domaine de l'aviation. Une connaissance métier forte est nécessaire pour les décoder. En outre, la langue utilisée est l'Anglais mais il peut arriver que certaines agences utilisent la langue locale de leur pays. Deux exemples de NOTAMs sont présentés figure 1. Ce langage NOTAM est cryptique car il doit être concis afin de transmettre le plus efficacement possible l'information. Afin que ce langage soit universel et qu'il puisse être rédigé et compris par tous (dans le domaine aéronautique), il existe des recommandations d'écriture (il est notamment fortement recommandé d'utiliser la liste officielle d'acronymes). Certains motifs (comme "RWY XX/YY CLSD") sont ainsi fréquents mais malgré ces recommandations, les opérateurs en charge de rédiger les NOTAMs peuvent prendre des libertés, commettre des fautes d'orthographe, etc., ce qui rend ce langage difficilement analysable de façon robuste par un système à base de règles. Ce langage peut parfois plus s'apparenter à du langage naturel.

3 Extraction d'informations contenues dans le NOTAM

3.1 Quelles informations ?

Une des informations importantes pour le pilote est de savoir quelles sont les pistes fermées d'un aéroport donné, *runway* ou *taxiway*¹. Dans le cadre de cette étude, nous nous sommes focalisés dans un premier temps sur la détection de fermeture de *runway* ou de *taxiway*, en identifiant la piste affectée par la fermeture.

Parfois, l'information de fermeture d'une piste s'accompagne de conditions :

- condition de type *géographique* (seule une certaine partie de la piste est fermée),
- condition de type *temporel* (certains jours de la semaine ou à une certaine plage horaire),
- condition de type "*opérationnel*" (pour certaines opérations, le décollage ou l'atterrissage par exemple),
- condition sur le *type d'avion* (par ex., la piste peut être fermée uniquement pour les avions dont l'envergure est supérieure à une taille donnée)

1. Une *runway* est une piste d'atterrissage tandis qu'une *taxiway* est une "route" permettant de connecter les pistes d'atterrissage aux terminaux, aux hangars ou à d'autres infrastructures

Parfois, au contraire, il peut y avoir des exceptions à la fermeture d'une piste ; elles peuvent être de même type que les conditions (à savoir géographique, temporel, opérationnel ou sur le type d'avion). Enfin, la raison de la fermeture (par exemple, pour cause de travaux) peut être fournie dans le NOTAM. Il peut être intéressant de capturer cette information ; le pilote peut vouloir la consulter.

Pour résumer, la liste des 6 entités d'intérêts dans les NOTAMs est la suivante : *closed_aeroway*, *runway_id*, *taxiway_id*, *condition*, *exception* et *reason*.

3.2 Notre approche : extraction d'entités imbriquées

Détecter les entités textuelles précédemment listées (*closed_aeroway*, *runway_id*, etc.) s'apparente à une tâche de reconnaissance d'entités (NER). Par rapport aux jeux de données classiques de reconnaissance d'entités nommées, pour lesquels les entités sont de type *flat*, les entités à détecter dans notre étude ont la particularité de pouvoir être imbriquées, comme le montre la figure 2 ; ainsi, dans la clause *closed_aeroway*, il est souvent fait mention de la *taxiway/runway* affectée ainsi qu'une éventuelle exception (ou condition) et une raison de fermeture.

Les méthodes à l'état de l'art pour les problèmes de reconnaissance d'entités sont maintenant basées sur des bi-LSTMs/CRFs depuis les travaux de [5] ; le pouvoir représentatif du contexte des LSTMs bidirectionnels a surpassé celui des features discrètes classiques (qui nécessitent en outre une connaissance métier et du *feature engineering*). Peu de travaux récents adressent pourtant le problème des entités imbriquées tout en utilisant des méthodes état de l'art, basées sur des bi-LSTMs/CRFs. Les travaux présentés dans l'article [4] répondent totalement à ce manque.

L'architecture de modèle utilisée pour notre étude est une architecture à base d'itérations, où chacune s'attèle à reconnaître les entités imbriquées du plus bas niveau au plus haut niveau d'imbrication. La reconnaissance d'entité se fait dynamiquement ; si à une itération donnée, une entité est détectée, alors l'itération suivante va essayer de détecter un autre niveau d'entité. Sinon, la détection de nouvelle entité s'arrête à ce niveau-là.

La représentation des mots fournie à l'entrée du réseau est une concaténation d'un *word embedding* classique ainsi que la concaténation des sorties d'un LSTM bidirectionnel au niveau du caractère. Les *word embeddings* peuvent être initialisés avec des *word embeddings* pré-entraînés de façon non supervisée sur une plus grande quantité de corpus non annoté.

4 Création d'un jeu de données

Etant donné l'absence d'ensemble de NOTAMs annotés, nous avons créé notre propre jeu de données annotés afin de pouvoir entraîner des modèles de *machine learning* pour la reconnaissance d'entités.

Créer un jeu de données annotés est coûteux ; le coût est encore plus élevé quand l'annotation requiert une connais-

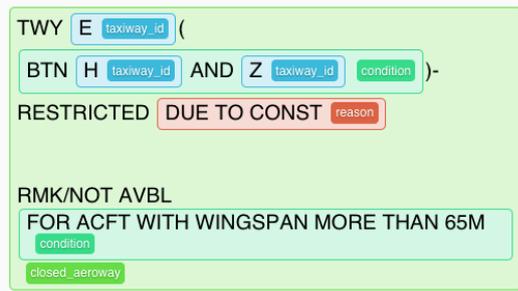


FIGURE 2 – Exemple d'entités imbriquées

sance métier comme c'est le cas pour les NOTAMs. Après une phase de rédaction de consignes d'annotation, 308 NOTAMs ont été distribués à un ensemble de 8 annotateurs. Chaque annotateur devait annoter environ 80 NOTAMs, en utilisant un outil d'annotation développé en interne, qui permet d'annoter des entités imbriquées. Chaque NOTAM est annoté par deux annotateurs différents.

Lors de la phase finale de validation des annotations, chaque NOTAM, pour lesquels les annotateurs n'étaient pas d'accord, était revu par un trio d'annotateurs, afin de corriger les éventuelles erreurs d'annotation et de valider une annotation finale.

Le corpus annoté a été divisé en un corpus d'apprentissage, de développement et de test ; les caractéristiques des différents corpus sont détaillés dans le tableau 1.

	Train	Dev.	Test
#NOTAMs	196	50	62
Vocabulary size	743	262	340
#tokens	4331	971	1161
Avg. length	22.1	19.4	18.7
closed_aeroway	187	42	57
runway_id	231	56	71
taxiway_id	385	82	97
condition	211	51	42
exception	25	7	9
reason	81	21	26

TABLE 1 – Description des corpus d'apprentissage, de développement et de test utilisés (taille des corpus et répartition des entités)

5 Expériences et évaluations

Pour notre étude, nous avons utilisé le *framework* Nested-NER², implémentant l'approche des entités imbriquées, décrite dans [4].

Avant l'étape d'apprentissage, une étape de pré-traitement est appliquée sur le corpus : tokenisation basique, suffisante pour les NOTAMs (tokenisation qui permet de regrouper

2. <https://github.com/meizhiju/layered-bilstm-crf>

les lettres, les chiffres et les symboles) puis une normalisation des chiffres (tous convertis en 0, afin de réduire la taille du vocabulaire).

Pour initialiser les poids des *word embeddings*, les premières couches du réseau ont été pré-entraînées de façon non-supervisée sur un ensemble de 22000 NOTAMs environ, en utilisant Word2Vec [6], en prenant une fenêtre de contexte de 5, un minimum d'occurrence des mots égal à 5, et avec différentes tailles (jusqu'à 300).

Durant la phase d'apprentissage, l'utilisation du *early stopping*, de la régularisation *L2* et du taux de *dropout* permet d'éviter le surapprentissage. La sélection des hyperparamètres incluant la taille du batch, le taux de *dropout*, l'utilisation d'*embedding* pré-entraîné et la dimension de l'*embedding* est réalisée par optimisation bayésienne.

Le meilleur modèle final sur le corpus de développement est obtenu en 71 epoch avec les hyper-paramètres suivants : *word embedding* pré-entraîné de taille 300, *batch size* de 24, taux de *dropout* de 0.48.

Le tableau 2 présente les résultats de notre modèle sur le corpus de test.

Entity type	Préc. (%)	Rec. (%)	F1 score (%)
closed_aeroway	69.5	71.9	70.7
runway_id	95.8	97.2	96.5
taxiway_id	97.8	89.7	93.5
condition	55.3	50.0	52.5
exception	100.0	33.3	50.0
reason	73.9	65.4	69.4
Overall	82.9	77.3	80.0

TABLE 2 – Résultats de l'extraction d'entités pour chaque type d'entité (en termes de Précision, Recall et F1 score)

Les performances d'extractions sont variables selon les types d'entités. La performance d'extraction des identifiants de pistes et de taxiways est plus importante que pour le reste des autres entités ; ces entités sont en effet, par nature, plus faciles à détecter et sont également plus nombreuses dans le corpus d'apprentissage. Les performances sur des entités plus longues comme *closed_aeroway*, *condition*, *reason*, etc. sont moins élevées car la détection est plus difficile sur des entités longues. Il faut relativiser ces résultats par rapport à la petite taille de nos données d'apprentissage ; le modèle devrait certainement bénéficier d'un accroissement de la taille de notre jeu de données annotées.

6 De l'extraction d'information à la visualisation contextuelle (carte)

Pour faciliter la compréhension des NOTAMs en opération, l'extraction des entités imbriquées n'est qu'une première étape. La finalité est d'être capable de les représenter dans leur contexte, notamment géographique (et temporel à terme). Ainsi nous proposons un prototype sous forme

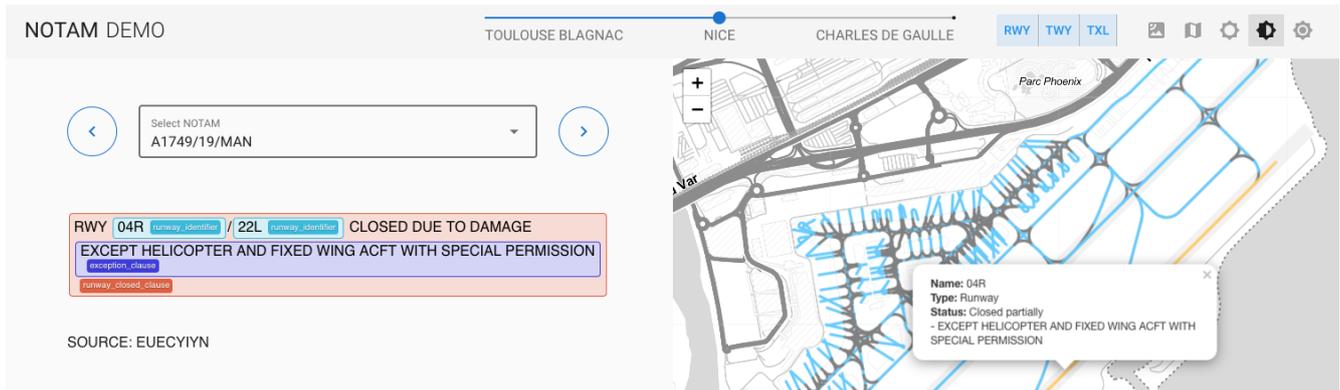


FIGURE 3 – Capture de la démonstration NOTAM

d'application web où chaque NOTAM, annoté automatiquement, est accompagné d'une visualisation sur une carte d'aéroport. Cette démonstration se concentre à ce stade sur la détection et l'affichage des *runways*, *taxiways* et *taxilanes* fermées (complètement ou partiellement, selon des conditions/exceptions extraites également). Les pistes ouvertes s'affichent automatiquement en bleu et celles fermées en rouge. Lorsqu'une piste est détectée partiellement fermée, elle s'affiche en jaune et les conditions/exceptions extraites sont listées dans une bulle contextuelle en cliquant dessus sur la carte.

La transformation des entités imbriquées telles qu'annotées par le système d'apprentissage machine vers la représentation finale pour l'affichage sur carte interactive s'appuie sur un ensemble de règles simples prédéfinies. A titre d'exemple, une entité "*runway_id*" détectée dans une "*closed_aeroway*" clause sera affichée fermée sur la carte (rouge), à moins que cette clause contienne aussi des conditions et/ou exceptions comme entités imbriquées, auquel cas la piste sera indiquée partiellement fermée (jaune).

7 Conclusion et perspectives

Dans cet article, nous avons décrit une démonstration qui permet d'afficher directement sur les cartes d'aéroport, les informations importantes contenues dans les NOTAMs, comme la fermeture de *runway* ou de *taxiway*.

Cette démonstration montre la valeur ajoutée d'un tel outil de visualisation de l'information pertinente couplé à une extraction d'information performante; il permettrait au pilote de se focaliser sur la tâche de préparation de vol en ayant toutes les informations disponibles, en lui évitant la tâche laborieuse de parcourir la liste exhaustive des NOTAMs.

Plusieurs pistes d'améliorations de la démonstration sont envisagées; la première piste concerne l'amélioration la brique même d'extraction d'information en augmentant la taille du corpus d'apprentissage annoté, en explorant notamment des méthodes d'*active learning* afin d'optimiser l'effort d'annotation en le concentrant sur les exemples les

plus pertinents.

L'autre piste d'amélioration est d'incorporer une mesure de confiance sur les entités détectées; étant donné la criticité de l'utilisation finale de l'information extraite, il est important d'afficher le degré de confiance du système automatique afin que le pilote en ait connaissance et puisse lui-même vérifier le message original si besoin.

Nous voudrions remercier Estelle Delpech, Denys Bernard, Monika Litvova et Eric Romey pour leur contribution au guide d'annotation ainsi qu'à l'effort collégial d'annotation.

Références

- [1] ICAO Abbreviations and Code, http://www1.atmb.net.cn/CD_web/UploadFile/2013052810480238.pdf
- [2] EUROCONTROL Guidelines for Pre-digital NOTAM Templates. <https://www.eurocontrol.int/sites/default/files/publication/files/eurocontrol-guidelines-pre-digital-notam-templates-ed-1.3.pdf>
- [3] Bobrow, Robert. Intelligent Semantic Query of Notices to Airmen (NOTAMs). BBN TECHNOLOGIES CAMBRIDGE MA, 2006
- [4] Meizhi Ju1, Makoto Miwa and Sophia Ananiadou, A Neural Layered Model for Nested Named Entity Recognition, Proceedings of NAACL-HLT 2018, pages 1446–1459
- [5] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer, Neural architectures for named entity recognition, Proceedings of NAACL-HLT 2016, pages 260–270
- [6] Mikolov, Tomas, et al. Distributed representations of words and phrases and their compositionality. Proceedings of NIPS 2013

