EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

A Hybrid Named Entity Recognition System for Aviation Text

Bharathi A, Robin Ramdin, Preeja Babu, Vijay Krishna Menon, Chandrasekhar Jayaramakrishnan, Sudarsan Lakshmikumar

KeepFlying, 10 Kallang Avenue, #0517 Aperia, Singapore 339510

Abstract

Named Entity Recognition (NER) is a crucial task in Natural Language Processing (NLP) that aims to identify and categorize named entities in text. While NER has been well-studied in various domains, it remains a challenging task in new domains where annotated data is limited. In this paper, we propose an NER system for the aviation domain that addresses this challenge. Our system combines rule-based and supervised methods to develop a model with little to no manual annotation work. We evaluate our system on a benchmark dataset and it outperforms baseline scores and achieves competitive results. To the best of our knowledge, this is the first study to develop an NER system that specifically targets aviation entities. Our findings highlight the potential of our proposed system for NER in aviation and pave the way for future research in this area.

Received on 30 June 2023; accepted on 12 October 2023; published on 20 October 2023

Keywords: Named Entity Recognition, Machine Learning, Aviation Herald, Spacy NER, GPE, Rule Augmentation

doi:10.4108/eetsis.4185

1. Introduction

NER is a subfield of NLP that involves identifying and classifying named entities in text data. Named entities are specific pieces of information that have names or labels associated with them, such as people, organizations, locations, dates, and numerical values. NER has been widely applied in various domains, such as healthcare, finance, and social media, and has become an important tool for extracting valuable information from unstructured text data. NER can be performed using several techniques, including rule-based systems, machine learning algorithms, and deep learning models. Rule-based systems involve defining a set of rules and patterns that can be used to identify named entities based on their characteristics, such as their format or context. Machine learning algorithms involve training a model on a labelled dataset of text data, where the model learns to recognize patterns and relationships between named entities and their surrounding text. Deep learning models, such as neural networks, have also been applied to NER tasks, which can learn to recognize complex patterns and relationships between named entities. Deep learning models can achieve high levels of accuracy and can be applied to various types of named entities and contexts.

In this paper, we present an NER system that is developed specifically for aviation data. The aviation industry collects vast amounts of data from a range of sources, including safety reports, aviation maintenance, and air traffic control. This data can be analyzed to gain insights into aircraft operations and the causes of aviation incidents, which can inform maintenance and repair activities. Despite the potential benefits, there has been limited attention given to NLP in aviation. This presents a significant opportunity to leverage advanced NER models to drive substantial improvements in the industry. NER is critical in identifying flight paths, locations, and the causes of incidents and accidents. By identifying and extracting flight paths, aircraft models, and the causes of incidents, we can optimize flight paths, reduce the risk of bird strikes and associated engine issues caused by foreign object damage, and enhance safety. Additionally, NER can be used to connect events and entities to identify areas for improvement in maintenance and repair activities, reducing downtime

*Corresponding author. Email: robin@keepflying.aero

and costs for airlines and aircraft lessors. Our main contributions will include the following

- an aviation text corpus crawled from Aviation Herald articles,
- a set of annotation rules that can be used to auto annotate text corpora for specific aviation domain entities,
- a retrained SpaCy NER model that drastically improve the response on aviation domain entities and geopolitical tags.

2. Background

Named Entity Recognition (NER) is a well established Information Extraction (IE) task in Natural Language Processing (NLP) that has received the attention of several researchers in the last few decades. Early approaches in NER relied on hand-crafted rules and heuristics, but more recent approaches have used the power of Machine Learning (ML) algorithms such as Support Vector Machines (SVM), Conditional Random Fields (CRF) and Neural Networks. Despite the progress made in NER, there are several challenges that still exist such as dealing with noisy data, handling rare and unseen entity types and dealing with language and domain-specific variations in data. So, in more recent times we are seeing an increase in attention on transfer learning based approaches that adapt to various domains and languages with less data. In this section, we provide a brief overview of the existing literature on NER, discuss some applications of NER in various domains, and the importance of domain-specific approaches in improving the effectiveness of models.

2.1. Overview of NER

Various NER approaches have been developed over the years, ranging from rule based systems to more advanced machine learning based techniques. NER was first introduced in the Message Understanding Conference (MUC) in MUC-1, which was held in 1987. MUC was a series of tasks held from 1987 to 1998 focused on IE. They were designed to address the challenges of processing unstructured text from news data and scientific articles, and NER was one of the key tasks in several MUC evaluations. The MUC-6 and MUC-7 were two of the most significant evaluations for NER. MUC-6, which took place in 1995, focused on identifying and classifying named entities in news articles such as persons, organizations, and locations [1]. In MUC-7, held in 1998, the NER task was expanded to include additional types of named entities, such as dates, times and other numerical quantities [2]. During the MUC evaluation campaigns for NER, the participating teams used a combination of hand-crafted patterns and statistical techniques. These approaches played a crucial role in establishing NER as a fundamental task in NLP.

A Hidden Markov Model (HMM) based approach was proposed by Kubala et al. for extracting named entities from the output of a speech transcription system [3]. The proposed scoring system involves first aligning the speech recognizer output to the reference text, followed by applying the MUC-6/MUC-7 NER scorer on the aligned and entity annotated text. In a study on NER that does not rely on a pre-existing list of entities, a combination of rule-based grammars and a statistical maximum entropy model was employed [4]. This method achieved a combined precision-recall score of 93.9% in the MUC-7 competition, demonstrating its effectiveness. Another NER approach based on maximum entropy is presented in the book [5] where the author presents a novel NER system called "MENE" (Maximum Entropy Named Entity) which uses probabilistic modelling with multiple sources of dictionaries resulting in an 88% F-measure on the MUC-7 evaluation. An extension of this approach was demonstrated by Chieu and Ng. using a set of sentence level local features and document level global features that led to improved performance in the MUC-6 and MUC-7 test data [6].

Further to these statistical approaches, one of the earliest works to apply machine learning for NER was proposed in [7] where combinations of word-level features, dictionary lookups and part-of-speech tags were used to train a decision tree classifier to detect names from news articles.

The annual Conference on Computational Natural Language Learning (CoNLL) holds significant relevance to NER, which happens to be one of the most crucial shared tasks featured at the conference. CoNLL plays a vital role in advancing research in NLP and ML. The conference features various shared tasks that challenge participants to develop NLP models that can perform specific tasks, such as NER, semantic parsing, and syntactic analysis. The CoNLL NER task was first introduced in 2002 and has since become a key benchmark for evaluating different approaches to NER. The shared task of CoNLL-2002 [8] focused on four entity types namely persons, locations, organizations and miscellaneous entities in Spanish and Dutch texts, while the task of CoNLL-2003 [9] concentrated on the same entities in English and German texts. Over the years, the CoNLL NER task has evolved to include more complex and diverse datasets, such as multilingual datasets and datasets with more fine-grained entity types. One of the earliest approaches developed for the CoNLL 2003 shared task on NER used Conditional Random Fields (CRF) with automated feature induction

and web-enhanced lexicons, which resulted in state-of-the-art performance on the test set with an F1 score of 84% [10] .

There have also been studies to extract entities with less or no annotated training data. One example of an unsupervised approach that defines a set of domain-independent extraction rules to extract named entities is proposed in [11] which also assesses the probability of the extracted entities using Pointwise Mutual Information (PMI) from search engine hit counts of the entities extracted. In addition, Nadeau presents a semi-supervised entity recognition system called BaLIE which creates a gazetteers of entities and uses rule-based heuristics to capture and classify entities in a document [12].

Over the years, the use of deep neural networks produced state-of-the-art results on several benchmark datasets. The first study to use a Bidirectional LSTM + CRF architecture for NER is described in [13], which conducted several experiments with Long Short Term Memory Networks (LSTMs) and CRFs for sequence tagging tasks and achieved near state-of-the-art result on the CoNLL 2003 dataset. The study presented in [14] pushed that boundary further by introducing a novel BiLSTM + CRF + Convolutional Neural Network (CNN) architecture that utilized both character and word-level representations and achieved state-of-the-art F1-score of 91.21% on CoNLL 2003 dataset [14]. Additionally, another study by Lample et al. introduced two neural architectures, including a BiLSTM and CRF model and a Stack LSTM model, which both demonstrated the best results among all models that did not use any external or language-specific feature on the CoNLL 2003 test set [15].

There are various domain-specific architectures for entity recognition, with a particular emphasis on the biomedical domain. Extensive studies have been conducted in this area in particular. An LSTM-CRF model proposed in [16] which utilizes the domain-specific Wiki-PubMed-PMC embeddings has demonstrated superior performance, achieving the best results on 28 out of 33 datasets. A novel biomedical Named Entity Recognition (NER) model, named D3NER enhanced the NER performance of the BiLSTM-CRF model using finetuned embeddings that consider multiple linguistic information [17].The first work to study the effect of transfer learning for BNER (Biomedical Named Entity Recognition) is presented in the work [18] which finds that transfer learning leads to improved F1 scores when information from a large silver-standard source dataset is transferred to a small gold-standard target dataset. The use of transfer learning for biomedical NLP tasks increased with the introduction of BioBERT, which is a domain-specific Bidirectional Encoder Representations from Transformers (BERT) embedding model that has been pre-trained on large scale biomedical text corpora [19]. It significantly outperforms previous state-of-the-art models on the tasks such as Biomedical NER, Biomedical Relation Extraction and Biomedical Question Answering.

There has also been research in the food science domain, where NER can be used to extract relevant information such as the names of food products, ingredients, and dietary recommendations from textual data. A dietary recommendations extraction model, drNER, is demonstrated in [20] which implemented a two-phase approach, utilizing dictionaries to identify entity mentions in the first phase and a syntactic parser to select and extract those entities in the second phase. Another closely related food based Named Entity Recognition system called FoodIE is proposed in [21] which uses several rules based on Part-of-Speech (POS) and semantic tags to identify food entities from text. Apart from the biomedical and food science domains, applications of NER are found in a number of other areas as well. These works can be found in the references [22–25]

The use of NLP techniques in the aviation domain has significantly increased in recent years. Specifically, NER has emerged as a crucial task to extract useful information from unstructured text data such as maintenance logs, incident reports, and customer feedback, which can be used to improve safety, efficiency, and customer satisfaction. A system for entity recognition in the Air Defence domain has been proposed by Guo et al. It involves constructing a domain knowledge graph and using a set of pattern recognition rules and verification methods to identify flying targets from simulation data, as described in reference [26]. A Chinese NER model utilizing BiLSTM and CRF has been proposed in [27] for the Aircraft Design field. The model uses attention mechanism and a ranger optimizer, resulting in a high F1-score of 88.91%. In reference [28], a Chinese NER system was proposed for civil aviation passenger reviews. The system leverages a one-dimensional CNN to extract local contextual features and LSTMs to extract long-range features, which are then combined and passed as inputs to a CRF model for entity labelling. The system outperforms other baseline models for this task, including BERT. An NER approach introduced for Chinese aviation security incidents data merges character and sentence embeddings and passes them through BiLSTM and CRF layers to predict entity labels belonging to one among 12 entity classes from Chinese safety incidents data [29].

There are other tasks in aviation NLP that are closely related to NER or depend on the outputs of an NER system. For example, an important task in civil aviation is understanding passenger intention from service requirements. The Dual Intent and Entity

Transformer (DIET) architecture was used by He et al. to train an intent recognition model for a generated service requirement corpus [30]. Recognizing call-signs from noisy Air Traffic Control transcripts can be a challenging task, as it falls under the umbrella of NER. By utilizing surveillance call-signs as contextual information during prediction, an Encoder-Decoder model developed by Blatt et al. improves the accuracy of call-sign recognition by a factor of 4 according to their experiments reported in [31]. The use of contextual information, in this case, surveillance call-signs, was found to be highly beneficial in recognizing call-signs from noisy Air Traffic Control transcripts.

3. Dataset: Aviation Herald

The Aviation Herald is a reliable and well-curated website that publishes reports of aviation incidents, accidents, and news related to commercial flights or commercial operators. In this study, we use web scraping to extract our dataset from The Aviation Herald website, and we use this to develop and evaluate an NER system that is specifically tailored to the aviation domain.

Our dataset consists of 26,531 records with 5 columns as described below:

1. Date - indicates the date of the aviation event reported in the news article.

2. Description - contains a short summary or headline of the article.

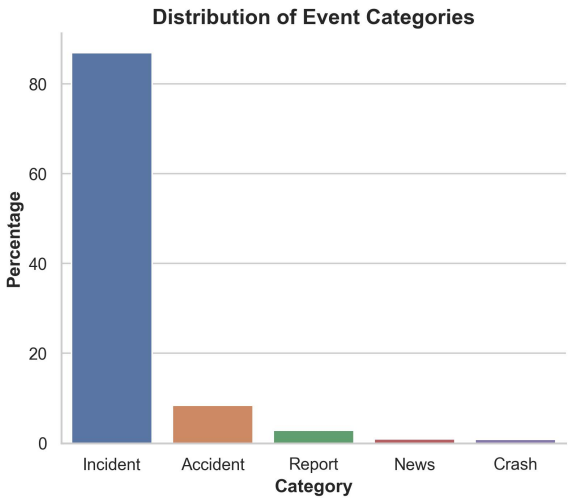
3. Filter - categorizes the aviation events into the five categories: Incidents, News, Accidents, Reports, and Crashes

4. News - contains the actual textual content of the news article

5. Link - provides the link to the original news article.

Figure 1.a shows the distribution of different categories of events or news articles present in the dataset. As evident from the graph below, Incidents make up nearly 85% of the events, followed by accidents and reports that account for 8% and 2% of the reported articles. Crashes are rare in general and make up less than 1% of the total events in aviation.

The articles in our dataset cover a time-line ranging from the year 2000 to 2022, providing a comprehensive view of aviation events over the past two decades. This long timespan allows us to analyse the trends and patterns in aviation events and develop a robust NER system that can accurately identify entities across various time periods.



(a) Distribution of Categories of events

Figure 1. Event Categories

After scraping the dataset, we performed several pre-processing steps on the dataset to ensure the text data was standardized and of high quality.

1. *Dropping null and duplicate records*: Null values or missing data points can cause issues with data analysis and modelling, so we removed them as a standard pre-processing step. Additionally, we also removed duplicate records to ensure that our analysis is unbiased

2. *Standardizing the text*: We standardize the text in the dataset by adding a space after all commas, fullstops and apostrophes to ensure that the text is consistent and that the tokenizer or parser does not treat certain phrases or words differently. We also remove extra white-space characters to just a single space to avoid issues with tokenizations and white-space errors in SpaCy.

3. *Removing non-alphanumeric characters*: Non-alphanumeric characters can also cause issues and therefore we remove them. However, we retain some characters such as hyphens "-" and Open and Closed parenthesis "(", ")" to maintain the integrity of certain entities. Hyphens are a necessary part of aviation entities such as Flight IDs (Eg: 6E-6646), Aircraft Model (A320-200) etc and we use parenthesis to help tag mentions of countries in the dataset while creating the training data.

4. *Removing NOTAM and METAR codes*: NOTAM and METAR codes are not relevant to our proposed NER system, so removing them helps to streamline the text data and remove any

unnecessary noise that could interfere with the entity recognition process.

5. *Removing stop-words*: Stop-words are commonly occurring words such as "the", "a", "they", "was", "at", etc., that are typically removed from text data because they do not carry much meaning or relevance. However, in our use case, certain stop-words are necessary to identify entities especially since we use a rule-based system along with SpaCy NER that considers stop-words to make predictions on the dataset. We only retrained a handful of stop-words that are necessary for SpaCy to identify entities. This helps to reduce the size of the vocabulary while still ensuring that the NER system can accurately identify entities.

The final dataset consisted of a total of 25,833 records. By standardizing the text and removing irrelevant information, we have created a clean and focused dataset that is suitable for our NER system.

4. Methodology

In this section, we describe our methodology for building an aviation NER system using SpaCy [32]. We chose to work with the SpaCy library for several reasons. Firstly, SpaCy has been shown to be highly accurate for named entity recognition tasks, consistently outperforming other popular libraries in benchmarks. Additionally, SpaCy is designed with deployment in mind, making it easy to integrate our NER model into real-world applications. Moreover, SpaCy has a user-friendly API, making it easy to customize the model to our specific needs. This allowed us to create a custom tokenizer as well as integrate our own heuristics and rules to improve the accuracy of our NER system.

The preliminary goal of our research is to capture and classify seven entities of interest: Airline, Aircraft Model, Aircraft Manufacturer, Aircraft Registration, Flight Number, Departure Location, and Destination Location. Since there are no publicly available entity-annotated training data for aviation, we adopted a three-step approach for training the NER system. In the first step, we used Regular Expressions (RegEx) and pre-trained SpaCy model to generate silver labels for the entities of interest. In the second step, we used these silver labels as training data for our SpaCy NER model. Finally, we extracted the country and city data from predicted Departure and Destination data with the help of a geographical entity library. Within this section, we will provide a detailed explanation of these steps. Our proposed methodology is depicted in Figure 2, which highlights the key steps and processes.

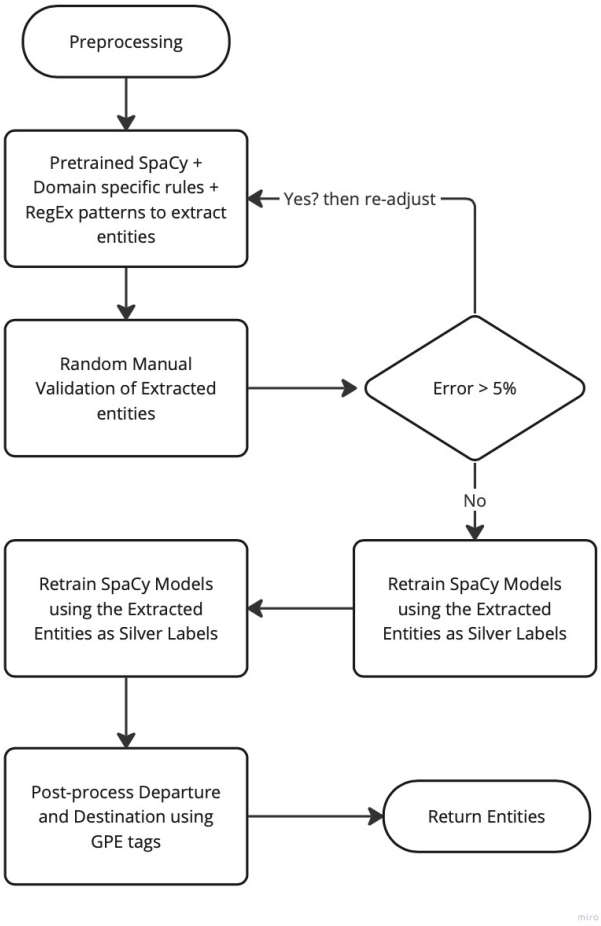


Figure 2. Proposed Methodology

4.1. Pre-trained SpaCy + RegEx

The first step in our NER approach for extracting entities from the news articles involves using a combination of RegEx (Regular Expressions) and pre-trained SpaCy NER. In addition to using this approach to extract entities, we also utilize the outputs of this model as silver labels to train a SpaCy NER model on our entities of interest. In situations where no annotated training data is available, generating silver labels using simple heuristics can be an effective approach to train an NER system for specific entities.

RegEx, is a sequence of characters that define a search pattern. By defining appropriate patterns, we can identify the entities of interest from the text data. For instance, to identify flight numbers, we used regular expressions to match the pattern of two letters followed by three to four digits. The RegEx approach is utilized to capture entities such as Airline, Aircraft Manufacturer, Aircraft Model, Aircraft Registration, and Flight Number. We apply a set of dataset-specific heuristics and rules to the text to identify these entities accurately. One of the key benefits of using RegEx to

generate silver labels is that it is a relatively quick and straightforward process. It does not require manual annotation of data, which can be time-consuming and resource-intensive. Also, RegEx models are scalable, and we can apply the same patterns to large amounts of text data, which would be difficult to annotate manually.

Next, we create a SpaCy document from the news text and use the re-tokenize functionality of SpaCy to combine multi-token entities into a single token. We then iterate through each token in the document and use a set of context rules and surrounding words to identify Geopolitical Entities (GPE). We extract GPE entities in two ways. First, we use the pre-trained SpaCy NER model to predict GPE entities. Second, we use a set of context heuristics to classify a token as GPE. We also use a set of rules to prevent SpaCy from classifying non-GPE tokens as GPEs, such as an Airline or an Aircraft Manufacturer. However, we found that the default SpaCy tokenizer cannot handle hyphenated words, such as "ATR-72-200", which results in the splitting of these words into three different tokens. To address this, we created a custom SpaCy tokenizer that allows us to retain the hyphenated words as a single token. Additionally, we removed suffix rules that split letters following a number, such as in the case of "DC-8T", which SpaCy would split into two separate tokens. This pre-processing step improves the accuracy of our NER system by ensuring that we do not miss any entities that contain hyphenated words or suffixes.

After developing our RegEx + pre-trained SpaCy NER approach, we applied this pipeline to the entire processed dataset from The Aviation Herald. By doing so, we were able to extract and classify entities in a wide range of news articles. This allowed us to generate a large dataset of silver labels that were then used to train the SpaCy NER model.

While using RegEx to generate silver labels can be a useful approach, it is important to acknowledge its limitations. RegEx models are not always accurate, and patterns may not always capture all possible variations of the entities of interest. For example, variations in the format of flight numbers or aircraft models can cause RegEx models to miss identifying the entities of interest. Therefore, it is crucial to evaluate the quality of the silver labels generated by the RegEx model before using them as training data for the NER system. To address this limitation, we took several steps to validate the quality of the silver labels generated by our RegEx model. One approach was to randomly sample a subset of the data and manually review the results to ensure that the model was accurately capturing the desired entities. We also continually adjusted the RegEx patterns based on the results of our validation process to improve the accuracy of the generated labels.

Through this iterative process, we were able to ensure that the silver labels generated by our RegEx model were of high quality and suitable for use as training data for the NER system.

4.2. Training SpaCy

In order to train our custom SpaCy NER model, we used the training data with silver labels generated by the RegEx combined with pre-trained SpaCy model. The training data was already in the required format with entity annotations and token indices. However, we needed to convert it into the SpaCy format in order to train the model.

To accomplish this, we created a '.spacy' file containing the training data in the required format. This was done by using the SpaCy library's DocBin class, which allowed us to efficiently create a binary file of SpaCy Docs. We iterated through each document in the training data, and for each entity in the document, we created a tuple containing the start and end indices of the entity, as well as its label. These tuples were then added to the Doc object using the *set_ents* method. Finally, the Doc object was serialized into the '.spacy' file using the DocBin class's *to_disk* method. Once the training data was formatted into the .spacy file, we created training and validation data sets. The training set consisted of 80% of the data, while the remaining 20% was used as the validation set.

Next, we trained the custom NER model using the train command provided by SpaCy. We used the 'en_core_web_lg' pre-trained model as the base model for our custom NER pipeline, and used the config.cfg file to specify the hyper-parameters for training the model. We used a batch size of 100, 10 epochs, learning rate of 0.001 and an evaluation frequency of 50. We also defined the drop-out rate, L2 regularization penalty, and other training settings in the config.cfg file. We present the results of our model in section 5

4.3. GPE Extraction

Extracting fine-grained entities such as cities and countries can be a challenging task. The "GPE"tags on which our model has been trained only provide a generic label for location names without specifying the exact city or country. Therefore, after training our custom NER model, we created an extraction pipeline to get entities such as countries and cities from our Departure and Destination prediction. To extract the departure and destination locations from the text, we first used RegEx to locate a span or subset of the SpaCy document that contained relevant information. This involved identifying patterns such as "departing from"or "to"followed by a location entity. Once we had identified this subset, we then extract the specific GPE entities that corresponded to departure and destination

Table 1. Results of pre-trained SpaCy vs Our model

SpaCy Model	Precision	Recall	F1 Score
Pre-trained NER	24.43	40.94	30.60
Custom-built NER	93.79	93.67	93.73

locations. For each GPE entity in the subset document, we used the Geonamescache library to map the entity to a specific city or country. Geonamescache is a Python library that provides access to a comprehensive database of geographical place names, including cities, regions, and countries. By using this library, we were able to accurately map GPE terms to specific locations, which greatly improved the accuracy and reliability of our entity extraction process.

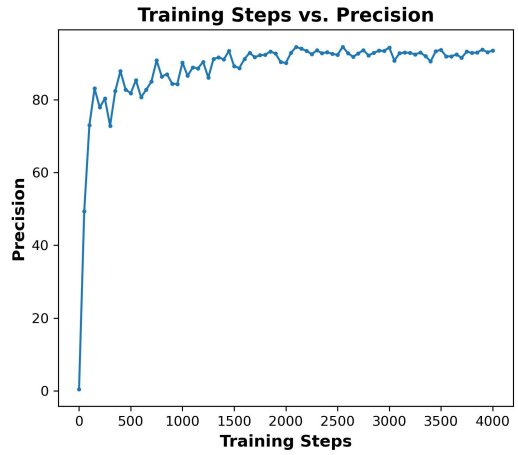
By layering the capabilities of Regular Expressions, SpaCy and Geonamescache library, we were able to develop a custom NER model that can accurately capture our entities of interest in aviation text. Thus, our approach demonstrates the development of a powerful domain-specific NER model without the need for manual annotation work

5. Results

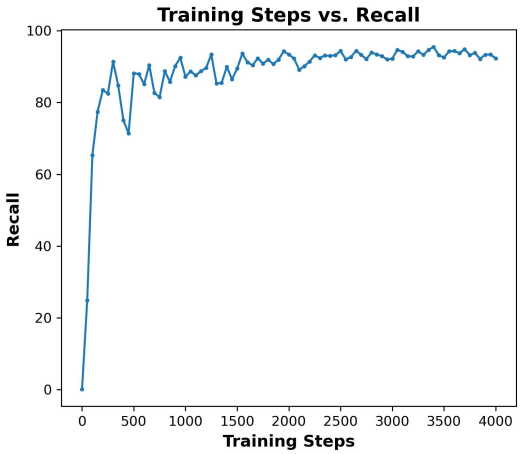
After validating the outputs of the RegEx model, we used 3000 data-points to train our SpaCy model. The data was split into 2500 training data points and 500 validation data points. We trained the model on all 7 entities of interest as described previously. We experimented with different hyper-parameters and features in SpaCy, and our final model achieved an F1-score of 93% on both the validation and test sets. which included all entities. Figures 3 illustrate the performance on the validation set with increasing training steps in terms of Precision, Recall and F1 score.

Table 1 presents a comparison of the performance of our proposed model with a baseline pre-trained SpaCy model, using precision, recall, and F1 score as evaluation metrics. As shown in the table, our proposed model achieves significantly higher scores across all metrics compared to the baseline model, indicating its superior performance in identifying entities and relations in civil aviation text.

A comparison of sample outputs given by our model against a pre-trained SpaCy model is presented in Figure 4. Figure 4.a shows the plain text of an article before any annotations and Figures 4.b and 4.c show the same text with entities annotated by our model and the pre-trained SpaCy model, respectively. The entities are marked with coloured boxes and the bold text to the right of an entity term represents the type of entity (such as an Aircraft Manufacturer or an Aircraft Model). It is clear from Figure 4.c that the pre-trained SpaCy is unable to capture highly technical and aviation specific entities such as an



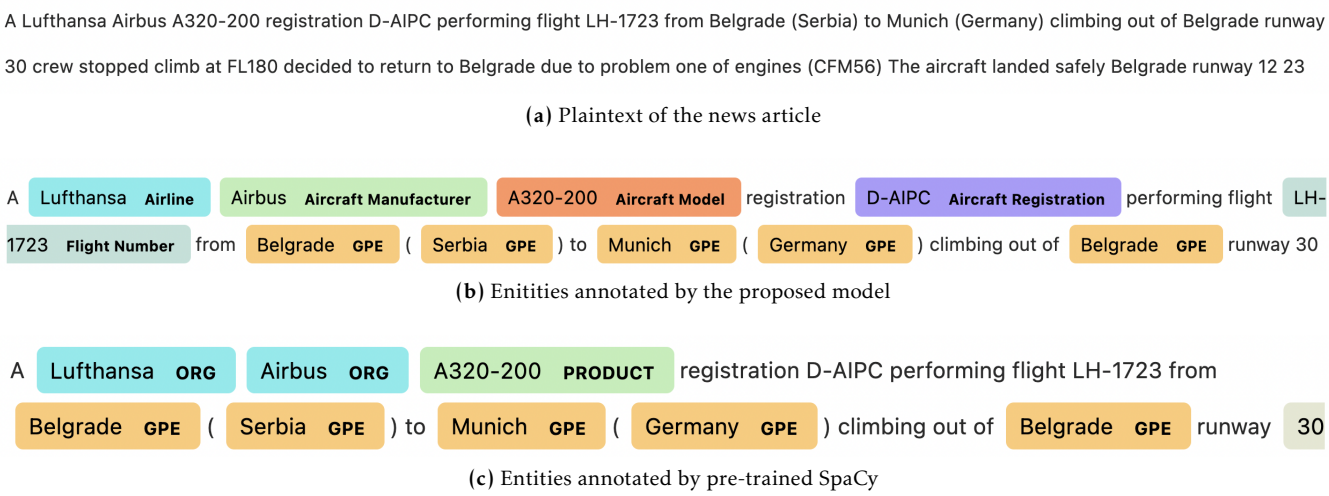
(a) Training Steps vs Precision



(b) Training Steps vs Recall

(c) Training Steps vs F1-score

Figure 3. Training step vs Validation score curve



(a) Plaintext of the news article

(b) Enitities annotated by the proposed model

(c) Entities annotated by pre-trained SpaCy

Figure 4. NER Outputs of the models

Aircraft Registration or a Flight number. The model only outputs generic labels such as a PRODUCT or an ORG which are insufficient for aviation specific applications. Therefore, these results underscore the need for a more specialized model to accurately classify aviation-specific entities. Our proposed model satisfies this need and demonstrate its effectiveness and novelty in capturing fine-grained entities in aviation texts. The results of our model, as can be seen from Figure 4.b, highlight its potential for practical application in this field.

6. Conclusion

In this research, we proposed a hybrid NER model specifically designed for the aviation domain, and presented comparative results. The model achieved high accuracy in identifying and classifying entities relevant to civil aviation. To the best of our knowledge, this is the first NER model proposed for civil aviation text with such a comprehensive set of entities. Our model outperformed existing pre-trained models for NER, demonstrating its efficacy in identifying aviation-specific entities. We adopted a combination of rule-based and supervised learning approach to train the custom SpaCy NER model, using silver labels generated by a RegEx model, which were manually validated and adjusted. This approach allowed us to overcome the challenge of limited annotated data, which is a common issue in the aviation domain. Our approach enabled the efficient use of unlabelled data, making it possible to leverage the vast amounts of aviation text data available.

While our results were promising, there is room for further improvements. Future research could explore the use of better features and more advanced model architectures to improve the performance of our proposed NER model. Additionally, we believe the development of an aviation-specific language model could further enhance the model's performance by improving its understanding of aviation jargon, entities and contexts.

Overall, our proposed custom SpaCy NER model for aviation is a novel approach that showed promising results, providing a foundation for future research in the domain of aviation NLP.

References

[1] Grishman, R. and Sundheim, B. (1996) Message understanding conference-6. In *Proceedings of the 16th conference on Computational linguistics* - (Association for Computational Linguistics). doi:10.3115/992628.992709, URL https://doi.org/10.3115/992628.992709.

[2] Chinchor, N. and Robinson, P. (1997) Muc-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, **29**: 1–21.

[3] Kubala, F., Schwartz, R., Stone, R. and Weischedel, R. (1998) Named entity extraction from speech. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop* (Citeseer): 287–292.

[4] Mikheev, A., Moens, M. and Grover, C. (1999) Named entity recognition without gazetteers. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics* - (Association for Computational Linguistics). doi:10.3115/977035.977037, URL https://doi.org/10.3115/977035.977037.

[5] Borthwick, A.E. (1999) *A maximum entropy approach to named entity recognition* (New York University).

[6] Chieu, H.L. and Ng, H.T. (2003) Named entity recognition with a maximum entropy approach. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* - (Association for Computational Linguistics).

doi:10.3115/1119176.1119199, URL https://doi.org/10.3115/1119176.1119199.

[7] Baluja, S., Mittal, V.O. and Sukthankar, R. (2000) Applying machine learning for high-performance named-entity extraction. *Computational Intelligence* **16**(4): 586–595. doi:10.1111/0824-7935.00129, URL https://doi.org/10.1111/0824-7935.00129.

[8] Sang, E.F.T.K. (2002) Introduction to the CoNLL-2002 shared task. In *proceeding of the 6th conference on Natural language learning - COLING-02* (Association for Computational Linguistics). doi:10.3115/1118853.1118877, URL https://doi.org/10.3115/1118853.1118877.

[9] Sang, E.F.T.K. and Meulder, F.D. (2003) Introduction to the CoNLL-2003 shared task. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* - (Association for Computational Linguistics). doi:10.3115/1119176.1119195, URL https://doi.org/10.3115/1119176.1119195.

[10] McCallum, A. and Li, W. (2003) Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* - (Association for Computational Linguistics). doi:10.3115/1119176.1119206, URL https://doi.org/10.3115/1119176.1119206.

[11] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S. *et al.* (2005) Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence* **165**(1): 91–134. doi:10.1016/j.artint.2005.03.001, URL https://doi.org/10.1016/j.artint.2005.03.001.

[12] Nadeau, D. (2007) Semi-supervised named entity recognition: Learning to recognize 100 entity types with little supervision doi:10.20381/RUOR-19854, URL http://ruor.uottawa.ca/handle/10393/29684.

[13] Huang, Z., Xu, W. and Yu, K. (2015) Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991* .

[14] Ma, X. and Hovy, E. (2016) End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics). doi:10.18653/v1/p16-1101, URL https://doi.org/10.18653/v1/p16-1101.

[15] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. and 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* (Association for Computational Linguistics). doi:10.18653/v1/n16-1030, URL https://doi.org/10.18653/v1/n16-1030.

[16] Habibi, M., Weber, L., Neves, M., Wiegandt, D.L. and Leser, U. (2017) Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics* **33**(14): i37–i48. doi:10.1093/bioinformatics/btx228, URL https://doi.org/10.1093/bioinformatics/btx228.

[17] Dang, T.H., Le, H.Q., Nguyen, T.M. and Vu, S.T. (2018) D3ner: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information. *Bioinformatics* **34**(20): 3539–3546. doi:10.1093/bioinformatics/bty356, URL https://doi.org/10.1093/bioinformatics/bty356.

[18] Giorgi, J.M. and Bader, G.D. (2018) Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics* **34**(23): 4087–4094. doi:10.1093/bioinformatics/bty449, URL https://doi.org/10.1093/bioinformatics/bty449.

[19] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H. and Kang, J. (2019) BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **36**(4): 1234–1240. doi:10.1093/bioinformatics/btz682, URL https://doi.org/10.1093/bioinformatics/btz682.

[20] Eftimov, T., Seljak, B.K. and Korošec, P. (2017) A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PLOS ONE* **12**(6): e0179488. doi:10.1371/journal.pone.0179488, URL https://doi.org/10.1371/journal.pone.0179488.

[21] Popovski, G., Kochev, S., Seljak, B. and Eftimov, T. (2019) FoodIE: A rule-based named-entity recognition method for food information extraction. In *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods* (SCITEPRESS - Science and Technology Publications). doi:10.5220/0007686309150922, URL https://doi.org/10.5220/0007686309150922.

[22] Jafari, O., Nagarkar, P., Thatte, B. and Ingram, C. (2020) SatelliteNER: An effective named entity recognition model for the satellite domain. In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (SCITEPRESS - Science and Technology Publications). doi:10.5220/0010147401000107, URL https://doi.org/10.5220/0010147401000107.

[23] Biswas, P., Sharan, A. and Kumar, A. (2015) Agner: Entity tagger in agriculture domain. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (IEEE): 1134–1138.

[24] Kumar, A. and Starly, B. (2021) "FabNER": information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing* **33**(8): 2393–2407. doi:10.1007/s10845-021-01807-x, URL https://doi.org/10.1007/s10845-021-01807-x.

[25] Leitner, E., Rehm, G. and Moreno-Schneider, J. (2019) Fine-grained named entity recognition in legal documents. In *Lecture Notes in Computer Science* (Springer International Publishing), 272–287. doi:10.1007/978-3-030-33220-4_20, URL https://doi.org/10.1007/978-3-030-33220-4_20.

[26] Guo, Z., Yu, L., Chen, G., Zhang, X., Wei, H. and Tang, Y. (2020) Entity recognition based on knowledge graph in air defense domain. *Journal of Physics: Conference Series* **1693**: 012168. doi:10.1088/1742-6596/1693/1/012168, URL https:

//doi.org/10.1088/1742-6596/1693/1/012168.

[27] Bao, Y., An, Y., Cheng, Z., Jiao, R., Zhu, C., Leng, F., Wang, S. *et al.* (2020) Named entity recognition in aircraft design field based on deep learning. In *Web Information Systems and Applications* (Springer International Publishing), 333–340. doi:10.1007/978-3-030-60029-7_31, URL https://doi.org/10.1007/978-3-030-60029-7_31.

[28] Xing, Z., Dai, Z., Luo, Q., Liu, Y., Chen, Z. and Wen, T. (2020) Research on name entity recognition method in civil aviation text. In *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT* (IEEE). doi:10.1109/iccasit50869.2020.9368691, URL https://doi.org/10.1109/iccasit50869.2020.9368691.

[29] Zhao, Y., Liu, H. and Chen, Z. (2021) Named entity recognition for chinese aviation security incident based on BiLSTM and CRF. In *2021 2nd Asia Conference on Computers and Communications (ACCC)* (IEEE). doi:10.1109/accc54619.2021.00021, URL https://doi.org/10.1109/accc54619.2021.00021.

[30] He, N., Ye, W. and Zhu, P. (2021) An approach to natural language intention understanding of civil aviation passengers based on DIET architecture. In *The 5th International Conference on Computer Science and Application Engineering* (ACM). doi:10.1145/3487075.3487101, URL https://doi.org/10.1145/3487075.3487101.

[31] Blatt, A., Kocour, M., Vesely, K., Szoke, I. and Klakow, D. (2022) Call-sign recognition and understanding for noisy air-traffic transcripts using surveillance information. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* : 8357–8361.

[32] Honnibal, M. and Montani, I. (2017–2021), spaCy: Industrial-strength natural language processing in Python, https://spacy.io.