

DS_233: For README Algorithm Card Names

Data Science

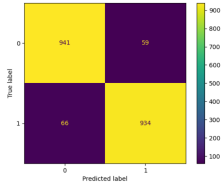
Exported on 04/30/2024

Table of Contents

1	Algorithm Card for: NHS UK Ratings & Reviews Names Identification	3
2	Intended Use	4
3	Metrics.....	5
4	Inputs and Outputs	6
5	Data.....	7
5.1	Evaluation Data	7
6	Methodology	8
6.1	Evaluation	8
6.2	Quantitative Analyses	9
6.3	Ethical Considerations.....	10
6.4	Caveats and Recommendations.....	10
7	Model Deployment.....	12

1 Algorithm Card for: NHS UK Ratings & Reviews Names Identification

At A Glance:

Description :	This algorithm is designed to identify reviews that contain peoples names. A BERT model will search a review and produce a list of strings which it thinks are names. Post processing is then applied to this list to remove common false positive cases (e.g. someone referring to an organisation). The results below and on this page refer to the algorithm as a whole (i.e. result after both the model + post processing have been applied)				
Developed By:	NHS England's Data Science Skilled Team (@Matthew Taylor)	Confusion Matrix:		Precision:	0.941
Launch Date:	📅 06 Feb 2024			Recall:	0.934
Version:	2.0			F1:	0.937
				FP Rate:	5.9%
				FN Rate:	6.6%

This algorithm consists of an off-the-shelf model from hugging-face which is then combined with post-processing steps to form the names rule. Documentation for the hugging face BERT model can be found [here](https://huggingface.co/dslim/bert-base-NER)¹. But in summary, the model endpoint takes a string as input, in our case, a review left by a user/patient of an NHS service (e.g. a GP Practice) and uses NER (named entity recognition) to identify peoples names. The model is able to recognise four types of entities (locations, organisations, person (names) and miscellaneous). The first steps of post-processing filters the list of entities to person only, using the PER entity. More details on post-processing can be found in the algorithm details section under methodology.

¹ <https://huggingface.co/dslim/bert-base-NER>

2 Intended Use

- **Scope:** Reviews are left by users/patients of NHS services (e.g. GPs, hospitals, dentists) on the NHS UK website. On submission of the review, this model/rule then works to identify names within the review title and body. Note: When the BERT model is combined with our post-processing (the names algorithm), it's scope is specific to the NHS UK RAR service.
- **Intended Users:** NHS patients who wish to leave a review about their experience with NHS providers, NHS.UK Rates and Reviews Team
- **Use Cases Out of Scope:** Any application with different rules or guidelines to that of the NHS UK RAR service

3 Metrics

- **Chosen Metrics:** Recall, Precision, F1, FN rate FP rate. Note that the metrics above are calculated from a balanced data sample which is not representative of the real-world balance of reviews containing names and those that do not.
- **Rationale:** Both high precision and high recall were desired features
 - High precision to reduce FP and reduce the number of reviews that would have to be manually checked by human moderators. This is because users have the option to send their review to a human moderator if they think the algorithm has made a mistake and erroneously flagged a word as a name (a FP)
 - High recall to reduce FN which in turn reduces the number of reviews containing names being incorrectly published since FN represent 'missed' names in a review

4 Inputs and Outputs

- **Inputs:** The model endpoint expects data of the following format

Model Input
<pre>{"data": "I went to the hospital and nurses Sarah and Fiona helped me"}</pre>

The names algorithm is called by:

Algorithm Input
<pre>names_rule("Review String","Organisation Name")</pre>

- **Outputs:** The model endpoint returns data in the following format

Model Output
<pre>{ "0": { "entity_group": "PER", "score": "0.9989973", "word": "Sarah", "start": "34", "end": "39" }, "1": { "entity_group": "PER", "score": "0.99879336", "word": "Fiona", "start": "44", "end": "49" } }</pre>

The algorithm output is:

Algorithm Output
<pre>(label (0/1), ["name1","name2","name3"])</pre>

5 Data

- **Data Overview:** The algorithm uses text data primarily sourced from user reviews about healthcare providers submitted on the [NHS.UK website](https://www.nhs.uk/contact-us/find-out-how-leave-review-of-nhs-service/)². These reviews, stored in Dynamics CRM for up to two years, were subject to a rigorous two-level moderation process involving both external and internal teams. The datasets includes a mix of reviews: those that have breached names rule and those published without issues. It provides a comprehensive range of user feedback, crucial for evaluating the model's performance.
- **Data Pre-processing and cleaning:** There was no pre-processing of the datasets. The internal moderation team did not review the datasets to ensure correct classification due to time and availability constraints. It was understood that there would be a degree of noise in the data and this was accepted. Before the review string is passed to the names algorithm, there are no changes made to the case of the string - it remains the same as the original as written by the user. This is contrary to version 1.0 of the algorithm which converted strings to lowercase before passing it to an **uncased** BERT NER model.

5.1 Evaluation Data

We focused on evaluating our model's performance with added post-processing steps, rather than assessing the general performance of the Hugging Face BERT model.

- **Datasets Used:** Names Rule Evaluation Datasets. Two datasets were used: 1000 reviews containing names and 1000 reviews that are publishable (no names included).
- **Why These Datasets:** The data are all examples of real names/published reviews, so will give us an accurate representation of how the model will perform in live deployment.
- **Representativeness:** To accurately reflect typical reviews, the evaluation was completed on a random sample of 1000 from each dataset. However this was not representative of the real-world balance of submissions, so the resulting TP, TN, FP, FN were scaled accordingly. There was no further breakdown of other characteristics like organisation type, sentiment, etc.

² <https://www.nhs.uk/contact-us/find-out-how-leave-review-of-nhs-service/>

6 Methodology

- Natural language processing (NLP), using BERT (Bidirectional Encoder Representation from Transformers) - a type of neural network
 - **Models Used:** [BERT-base-NER](#)³
 - **Alternative models considered:** [BERT-base-NER-uncased](#)⁴, [BERT-large-uncased-finetuned-ner](#)⁵, [BERT-large-NER](#)⁶
- **Algorithm Details:**

Our process begins by sending the review text to a frozen version of [BERT-base-NER](#)⁷ which has been deployed to an endpoint. There, the model breaks down the text into tokens and assigns various entity labels to them. The initial post-processing step involves filtering these entities to focus only on the "PER" (person) entity. This helps us compile a list of names mentioned in the review.

1. The next steps in post-processing include:
 - a. Removing Non-names: We discard names that are found in a pre-defined allow-list.
 - b. Adding definite names: We include names from a block-list into the results. (Note: This block-list is not yet in use, so it currently doesn't affect the outcome.)
 - c. Allowing Organisation names with fuzzy matching: We use the organization's name known from the review and perform fuzzy matching with the names identified by the BERT model. If the similarity score exceeds a certain threshold, we infer that the name refers to the organization rather than an individual, and thus we allow it through.
 - d. Allowing Name Sign-offs: Reviews can be signed off with the user's name. Therefore, any names in the last few words of the review are not filtered out.

6.1 Evaluation

- **Algorithm Validation:** The BERT model itself was not evaluated by us but we did evaluate the names algorithm as whole, including the pre and post processing steps applied to review submissions and lists of names. The rule was tested with a balanced data set of reviews containing names and reviews not containing names. There was a preference to reduce false positive cases as in the real-world we expect far more more reviews without names in them than with names, so FP cases will scale up more under real traffic than FN. However, we still wanted a low rate of FN so we took a balanced approach to reduce FP and FN.
- **Evaluation Focus:** When choosing a BERT model and post-processing steps we focused on obtaining a low number of FN and we wanted to ensure the model (which was trained on the conll2003 dataset) was producing good results for our use case (NHSUK reviews). Once the model was chosen, our evaluation focused on the effectiveness of our post processing steps applied to the results returned

³ <https://huggingface.co/dslim/bert-base-NER>

⁴ <https://huggingface.co/dslim/bert-base-NER-uncased>

⁵ <https://huggingface.co/Jorgeutd/bert-large-uncased-finetuned-ner>

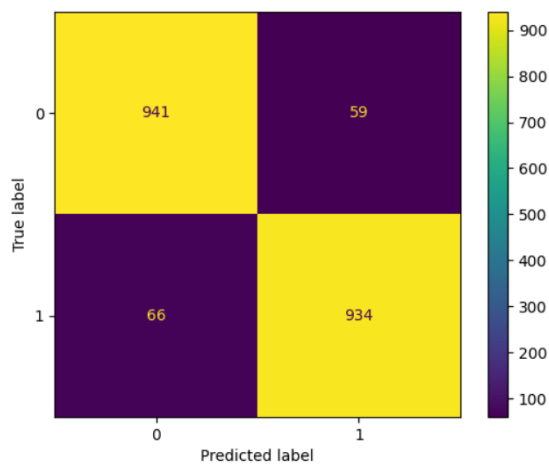
⁶ <https://huggingface.co/dslim/bert-large-NER>

⁷ <https://huggingface.co/dslim/bert-base-NER>

from the model. We used examples of FP to improve the post processing steps (e.g. we added some common FP names to an allow list such as 'god', 'dad').

6.2 Quantitative Analyses

- **Performance Breakdown:** The [BERT-base-NER](https://huggingface.co/dslim/bert-base-NER)⁸ model performed best when passed a review with case unchanged, with high accuracy in identifying reviews containing names. The results of this analysis in line with the evaluation methods described in **Algorithm Validation** are below:



Metrics:

Precision:	0.941
Recall:	0.934
F1:	0.937
FP Rate:	5.9%
FN Rate:	6.6%

- **Bias and fairness Analysis:** Bias analysis for the names algorithm is pending.

⁸ <https://huggingface.co/dslim/bert-base-NER>

6.3 Ethical Considerations

- **Sensitive Data Use:** All the data which users submit gets held in a Microsoft Dynamics 365 database. The reviews themselves can contain PID (e.g. people's names), and information about the reviewer (email, IP address) is also held as part of the record. By submitting their review for publication, the user consents for their record to be used in analysis to improve the service. These records are automatically deleted or redacted after a 2 year period. Note that in order to train our models, we have to export this data and import it into Azure Machine Learning Studio (AMLS). This does **not** have the same automatic deletion policy in place. In principle, it would be simple enough to create our own retention policy enforcer inside AMLS. However, this has not been done at time of writing, and indeed we have not thoroughly explored the extent to which the policy applies in the adapted context. Coming to a clear answer on this and implementing a solution is one of the outstanding tasks to accomplish.
- **Implications for human safety:** There is a risk that the model does not catch every name left in the review and therefore identifies someone (a patient or staff member), however it is unlikely that this will result in a risk to safety for the named individual. It is also possible that someone, intent on naming an individual, could find a way around the model (e.g. by including a number in the name).
- **Potential Harms:** Erroneously publishing a review with a name in could identify an individual. Erroneously flagging a review as containing a name could be frustrating for the user. However, organisations can still manually flag reviews that they think break RAR policy (i.e. contain names) and users will have the ability to send the review to a human if they think the model has erroneously flagged a name.

6.4 Caveats and Recommendations

- **Caveats and Limitations:**
 - The BERT model is restricted to 512 tokens whereas the maximum character count for a review is currently 3000. This means that it is possible for longer reviews to be truncated by BERT and for parts of a review to remain unchecked by the model. To counter this, we can split longer reviews into chunks and pass each chunk the model separately, aggregating the results in the Flask App.
- **Future Directions:**
 - Possible improvement to `allow_name_signoff()` function to only remove a name if it appears once in the submission
 - Identify diseases - sometime names of diseases are flagged as names (seems to apply more to cases where diseases are misspelt)
 - Identify typos - typos and bad grammar can cause the model difficulty in determining names from non-names and vice versa. Spell checking before submission to the model was attempted but the results were not good enough - it is worth exploring further.
 - Add further logic to the Flask App to ensure longer reviews are not truncated by BERT's token limit.

- **Additional Notes:**
 - **Influencing Factors:**
 - The position of the any flagged names with the review
 - The organisation of which the review is about
 - Submissions with misspellings or poor grammar can reduce model accuracy (see bias analysis when completed)

7 Model Deployment

- **Versioning:** The version of the model is v1.0, the algorithm version is v2.0
- **Links:** The BERT-base-NER model is stored in the Azure Model Registry which is accessible only to authorized members of the NHS England's Data Science Team. However, the model is open source and accessible from [hugging face](https://huggingface.co/dslim/bert-base-NER)⁹. Links to public Github repo (which contains the .py file for model deployment) and the repo containing the Flask App (where the algorithm code is stored) are pending, dependent on public code release.
- **Inference Process:** The inference process for the names algorithm involves receiving and processing user reviews in real-time through a Flask app, which queries the Azure-hosted model endpoint. The model first tokenises the text, and then performs the token classification task to pick out entities. The entities are returned to the Flask app which are then filtered and processed by the algorithm as outlined in the methodology.
- **Deployment Strategy:** The model, developed on Azure Machine Learning, leverages Azure's managed services. This means the compute resources are managed by Azure, simplifying scaling and management processes. The models are deployed using the Standard_DS3_v2 SKU, a general-purpose endpoint type that provides a balance of CPU, memory, and I/O, suitable for various workloads. This SKU features 4 cores and 14 GiB of RAM. The deployment allows for both manual and automatic scaling, with autoscaling as a built-in feature. The current setup, with an instance count of 1, is optimized for the anticipated workload but can be adjusted based on demand. Monitoring performance and response times dashboards were set up to determine the need for scaling, either by increasing the number of instances or upgrading to a more powerful SKU. The endpoint is secured with key-based authentication to ensure only authorized access.
- **Model Performance in Deployment:** There are no results post-deployment at the time of release, since this model was not dark-launched before going live. Going forward, ongoing monitoring and evaluation are in place to ensure the model's performance remains consistent and reliable, with periodic updates as needed.
- **Timing Analysis:** Some general timing analysis was performed to see how quickly the Flask app is able to query, receive and handle the results from each endpoint. This can be used as an indirect measure of the endpoint latency for practical purposes. The names algorithm is one of our slower algorithms, with the Flask app taking 0.73 seconds (mean) to send and receive the names classification per review. Since the various moderation endpoints are queried in parallel, this means that the performance of the names algorithm is not a limiting factor to the speed of the automoderation processing. Further to this, user testing of the review submission experience was completed and found the speed of the automoderation processing to be acceptable.

⁹ <https://huggingface.co/dslim/bert-base-NER>