Analysis of Twitter Accounts Descriptions : Named Entity Recognition

Priya Varadarajan, Anne Kirika, Victor Ernoult

Introduction

Interested in the subject of Named Entity Recognition, our team decided to study the descriptions of Twitter handles and attempt to retrieve useful information from it. This information would in turn be used to identify key characteristics of 2 major news channels'(CNN and Fox News) audiences.

This project will explore two assumptions we made in the beginning. Firstly, we expect to be able to retrieve relevant information from Twitter accounts descriptions. This may prove to be a challenge, as the format varies greatly between accounts and does not follow any pre-defined structure.

Secondly, we assume the 2 news channels will have different audiences, identifiable through accounts' descriptions. We purposefully selected 2 notoriously different news channels: Fox News, supposedly biased in favor of the Republican Party in the United States, and CNN, supposedly biased towards the Democratic Party.

Approach to Named Entity Recognition

Data Extraction

The data was retrieved using rtweet, an R library wrapper for the Twitter API. To make sure the data fits our expectations, we filtered it with a certain criteria. After skimming out irrelevant ones, only English accounts with a description and a follower count over 250 remained, the follower count helped to filter out inactive accounts.

Methodology

To tackle the recognition of named entities, we considered 3 different approaches:

- Existing NLP packages for NER to automatically classify entities
- Dictionary-based method matching known words with their assigned entity name.
- Machine learning Approach using CRF (Conditional Random Fields) and Random forests algorithms

We will explain our approach for all techniques, the advantages, efficiency and the difficulties encountered.

Using Existing NLP packages for Entity Classification

Our first attempt at tackling the problem consisted in applying existing NLP packages on newly seen text data, here the Twitter descriptions. To do so, we explored several NLP packages such as Spacy, MonkeyLearn and OpenNLP. Such packages offer models trained on large sets of labeled text documents, in which they learn to recognize entities within a sentence. They can then be fed entire sentences and output the words recognized as entities.

On Twitter descriptions, the results turned out to be disappointing overall. Most entities easily recognizable by a human eye were not identified, and many random words were misassigned. This is explainable by the lack of structure in the descriptions. These do not follow any format, often use diminutives, slang, and rarely form complete sentences. The models were trained on classic texts and used information from the whole sentence to make their predictions, which is not available here.

	user_id	Description
1	948997266	Iowa State University
2	2762605662	Long live Britain and America and all the true patriots.. The best in the West
4	560692070	#LongLiveBigBen
5	1088956155500490752	IM A SURVIVOR AKA A WALKING MIRACLE AKA A PHENOMENAL HUMAN BEING.
6	1083912778564489216	Official Twitter page of Salvador Rodriguez. - Running for Mayor of South Bend 2019 - Mainta

Since out-of-the-box models cannot be applied in this case, too particular, the automatic labeling of data into entity categories cannot be done through existing NLP packages.We resorted to a dictionary based approach to tackle the issue.

As discussed in class, both methods could eventually be combined into one. Once the entity tagging is done, and if it is accurate enough, it is conceivable to train a model on the labeled data to be able to treat unseen descriptions and potentially conduct a similar study on a different corpus of followers. Model creation is covered in the subsequent tabs.

Dictionary based approach

a) Data Preprocessing

To start with the dictionary based approach we had to do some preprocessing of data and tokenise them in order to tag them to named entities. We used existing text mining packages like 'tm' and 'udpipe' for cleaning the data and annotation. We used tm package to remove whitespaces, change the sentence to lower case, removal of stopwords and remove punctuation as they were not needed for the entity recognition.After this annotation was done to divide the text into token along with POS tagging.

Table 2: Resulting data:

doc_id	sentence_id	sentence	token_id	token	lemma	upos
doc1	1	iowa state university	1	iowa	iowa	NOUN
doc1	1	iowa state university	2	state	state	NOUN
doc1	1	iowa state university	3	university	university	NOUN
doc2	1	long live britain america true patriots best west	1	long	long	ADJ
doc2	1	long live britain america true patriots best west	2	live	live	ADJ
doc2	1	long live britain america true patriots best west	3	britain	britain	NOUN

b) Dictionary Creation and mapping entities

In an effort to extract information from descriptions in form of entities, we defaulted to creating a large set of words dictionary belonging to entities of interest namely profession, organization, hobbies, personality traits and religious affiliation. Description texts and tokens were then matched to these lists of words to map the entities. Two ways of performing the match were considered.

Flexible match approach

Since the descriptions do not follow any format, an information can often be hidden under a diminutive, a misspelled word, or a different way of saying it than the exact word in the dictionary. To partially compensate for this issue, we implemented ways of comparing similarities between words or expressions, to allow a partial match to still lead to a classification. One of these ways is the Levenshtein distance, which measures the difference between two strings. It is implemented in base R with the grepl function.

```
# Example of a grepl utilization
```

```
lookup <- function(word, dictionnary){
  # Checks if the word is there, bounded to avoid finding it within another word
  s = sapply(dictionnary, grepl, pattern=paste0("\\b", word, "\\b"))
  return(names(dictionnary[s])[1])
  }
```

Overall though, in spite of trying several alternatives, an increased flexibility in matching always led to an overflow of false positives and misclassifications which rendered the data invalid. We therefore went to a completely strict match to avoid discrepancies and huge number of false positives.

Strict match approach

In the strict match approach the dictionary words were matched with the tokens and they returned the entity in dictionary if there was a perfect match and this approach seemed to be more prominent in our case for mapping the entities.This was done using the dplyr function map values

```
# Example of the exact match approach

#Mapping the values from the Dictionary into a new column for the entity types based on the tokens

CNN_details$Entity <- plyr::mapvalues(CNN_details$token, from = dictionaryEntity$Value, to = dictionaryE
```

Below is the sample of entity mapped data

Table 3: Named Entity:

doc_id	sentence
doc1	optimistic
doc2	fav youtuber moesargi fav music artist pink fav band sos favourite author sarah j maas im book hoarder enjoy ner
doc2	fav youtuber moesargi fav music artist pink fav band sos favourite author sarah j maas im book hoarder enjoy ner
doc2	fav youtuber moesargi fav music artist pink fav band sos favourite author sarah j maas im book hoarder enjoy ner
doc2	fav youtuber moesargi fav music artist pink fav band sos favourite author sarah j maas im book hoarder enjoy ner
doc2	fav youtuber moesargi fav music artist pink fav band sos favourite author sarah j maas im book hoarder enjoy ner

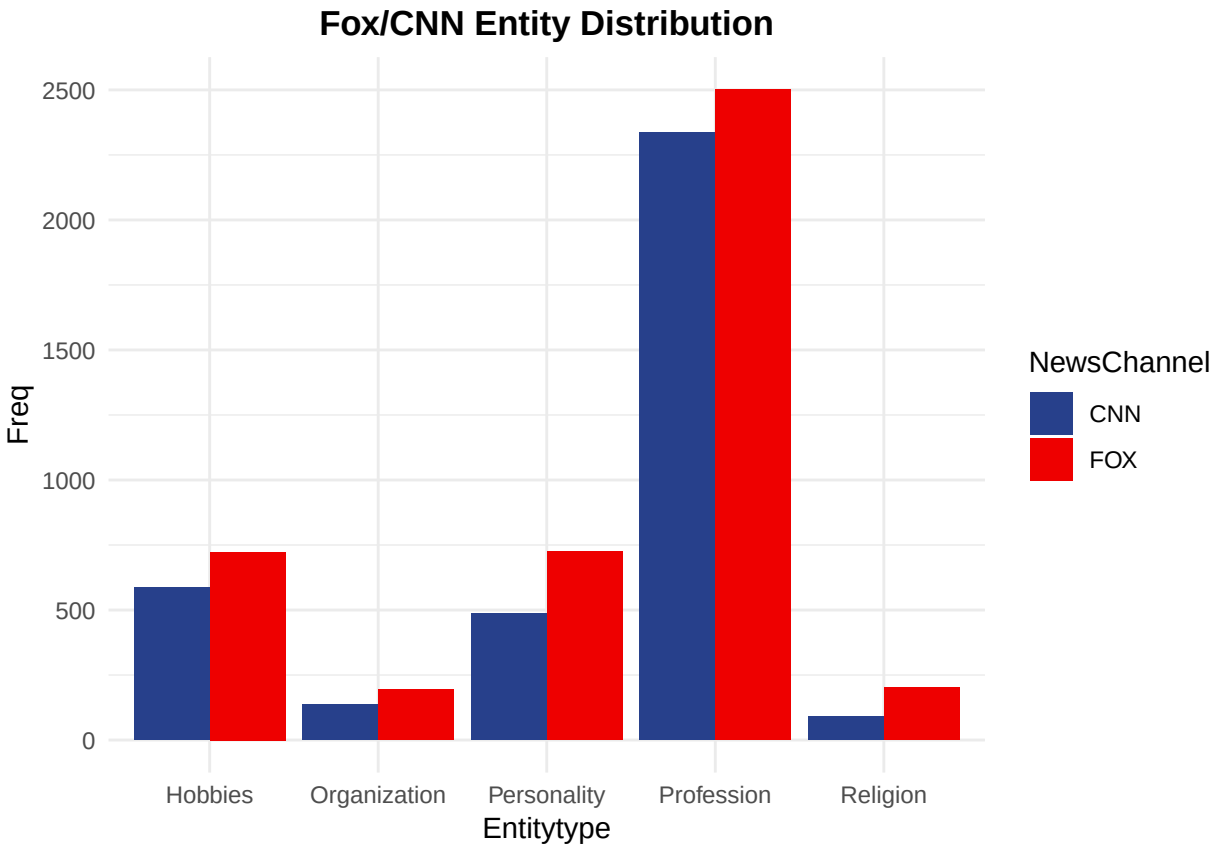
This approach was followed for both CNN and fox and the results were quite efficient.

Results and insights as part of dictionary based approach

Starting with an intial follower count of **3951** and **4620** for CNN and FOX news respectively,the data sets are prepared for NER process by first tokenization and lemmatization of the description variable of the news channels followers description. Lemmatized data is then matched up with a dictionary lookup of defined entity types. The insights infered from the entities are :-

Profession Entity Highest

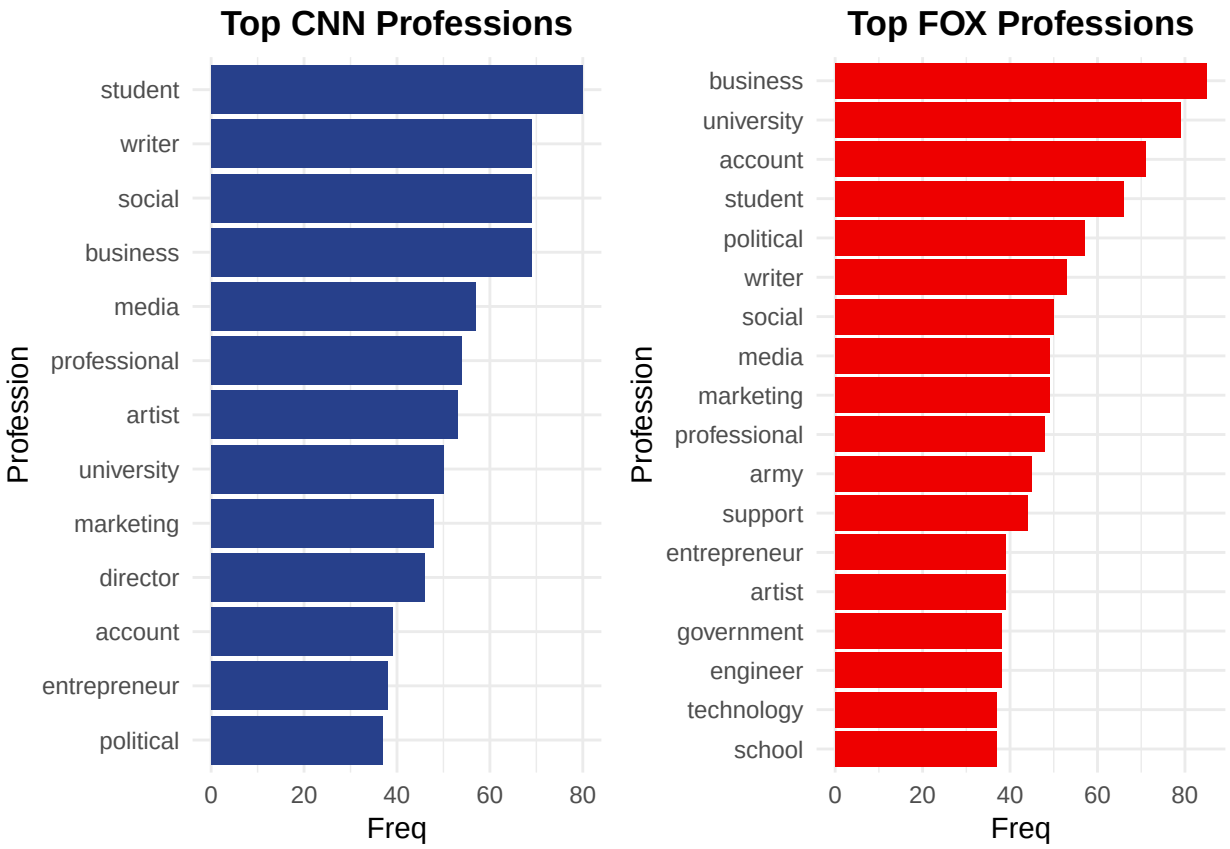
Combined entity types for both Fox News and CNN show that most of the followers revealed professions in their profiles more than the other entity type covered in this paper.Religion was the least.



Entity Distribution Across CNN and Fox

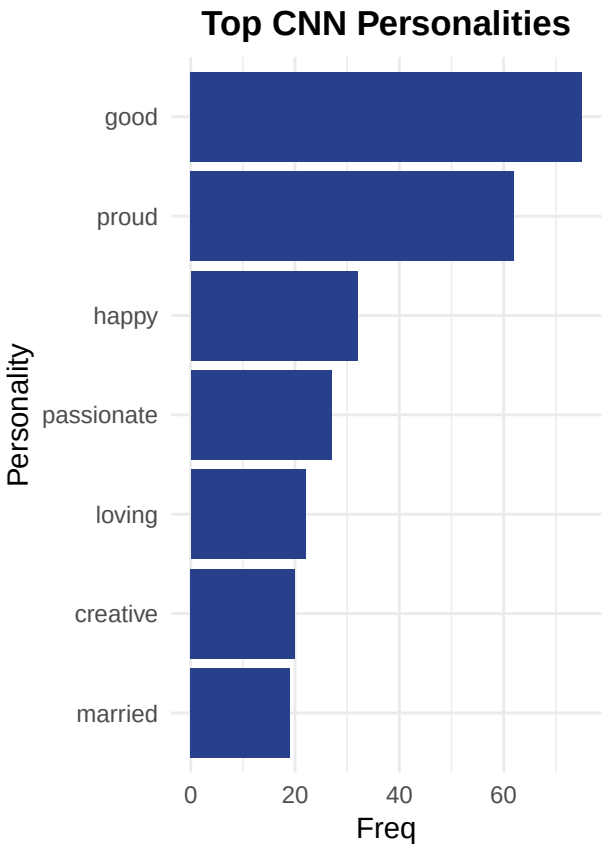
Profession Distribution

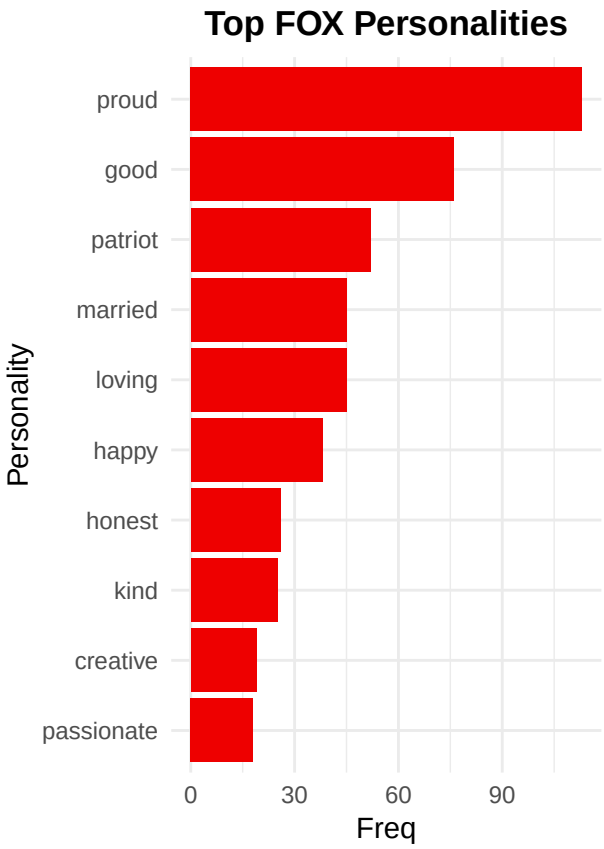
There is about **17%** Professions that are in Fox and not available in CNN followers,whilst **12%** of professions available in CNN and not In Fox News.Although CNN had lesser entities due to intial dataset used,profession entities revealed that CNN were more scholary having profession entities such as student,school,university,writer and teacher higher than Fox. Some of the missing professions from CNN were firefighter,technologist,biomedical,meterologist e.t.c whilst those missing from Fox include painting,embroidery,badminton,dymnastics.



Personality Distribution

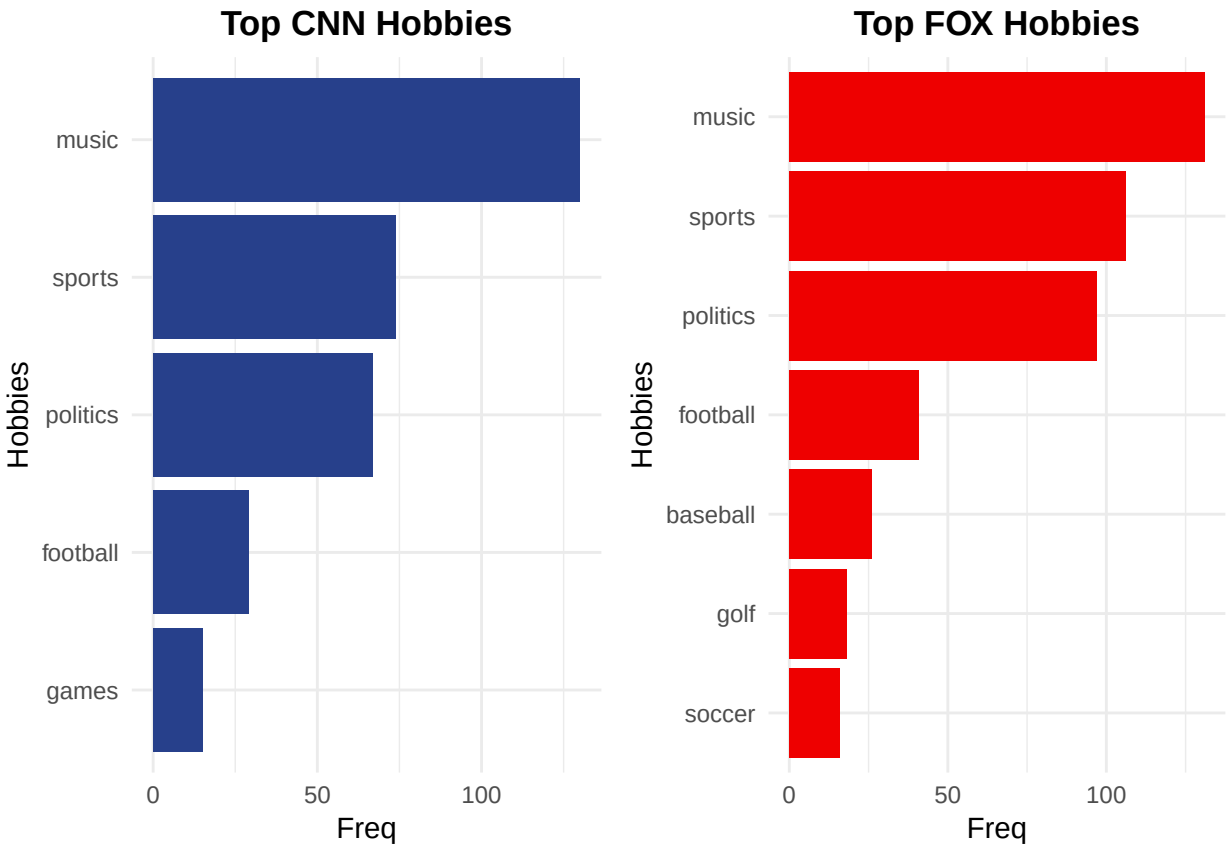
Predominant personality entity in both CNN and Fox News are **good** and **proud** however the number of proud in Fox News is double that in CNN. Some key personality entities missing in CNN and are in Fox include **trustworthy**,**emotional**,**dynamic** and **hypocrite**. Personality entities in CNN and are missing in Fox include **anxious**,**energetic**,**adventurous**,**passive**,**jealous**.





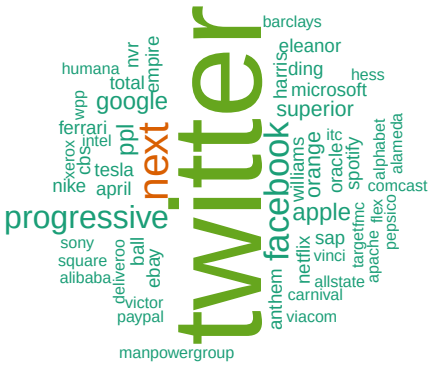
Hobbies Distribution

Music,**sports**,**politics** and **football** emerge as the top interests for both news channels. Some hobbies missing in CNN entities include yoga,bowling,drawing,coloring and canoeing whilst badminton,weights,embroidery,curling,trekkie,birdie are missing in Fox.



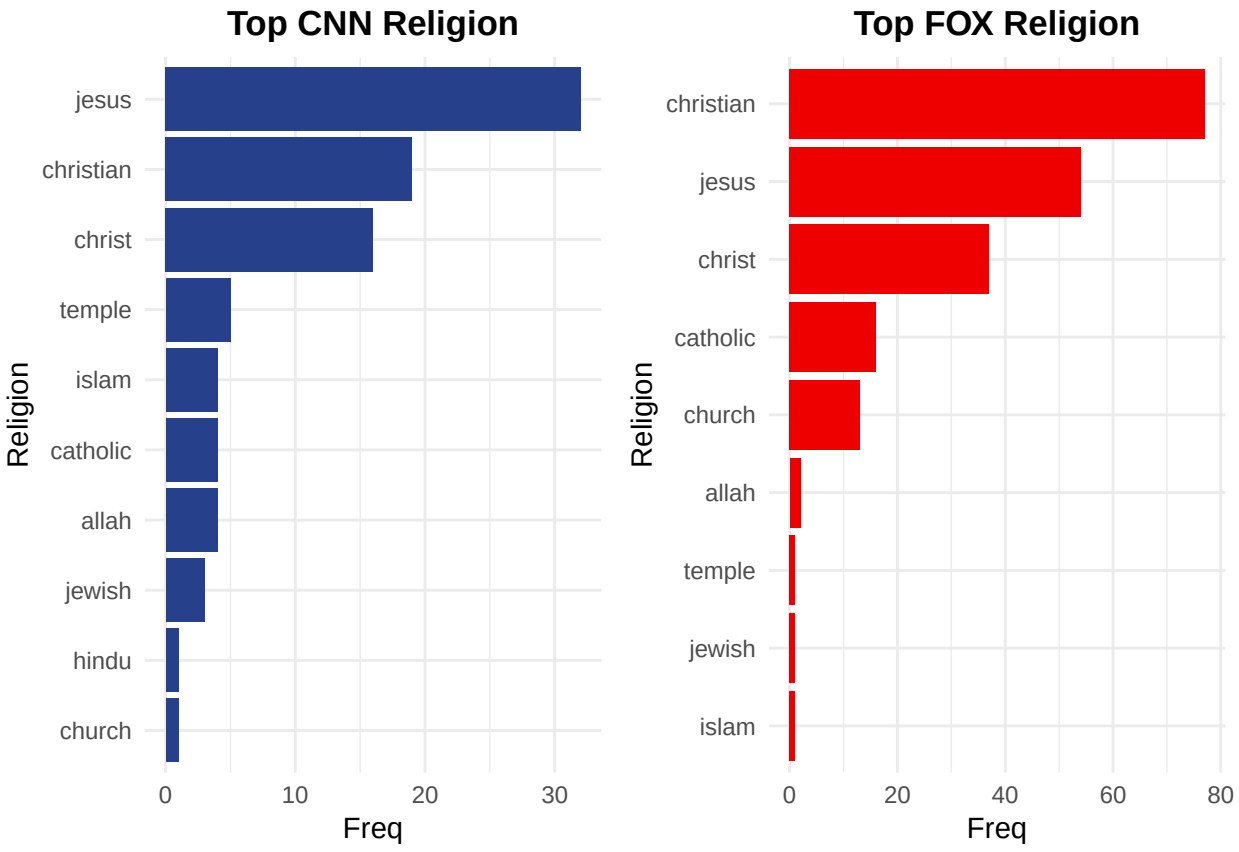
Organization Distribution

Twitter and facebook are the top organization entity. However not alot of followers display their organizations,this is seen by the low count of organization entities.For these reason both entities are combined for the news channels. Some organizations present in Fox and missing in CNN include Total,CBS,Nike,Manpowergroup,Spotify.Ding,SAP,Paypal,Sony,Xerox, Deliveroo are among the organizations missing in Fox News. It is also important to note Twitter identifications may very well be due to the user talking about a Twitter account of anything else related to the company, without being an employee there.



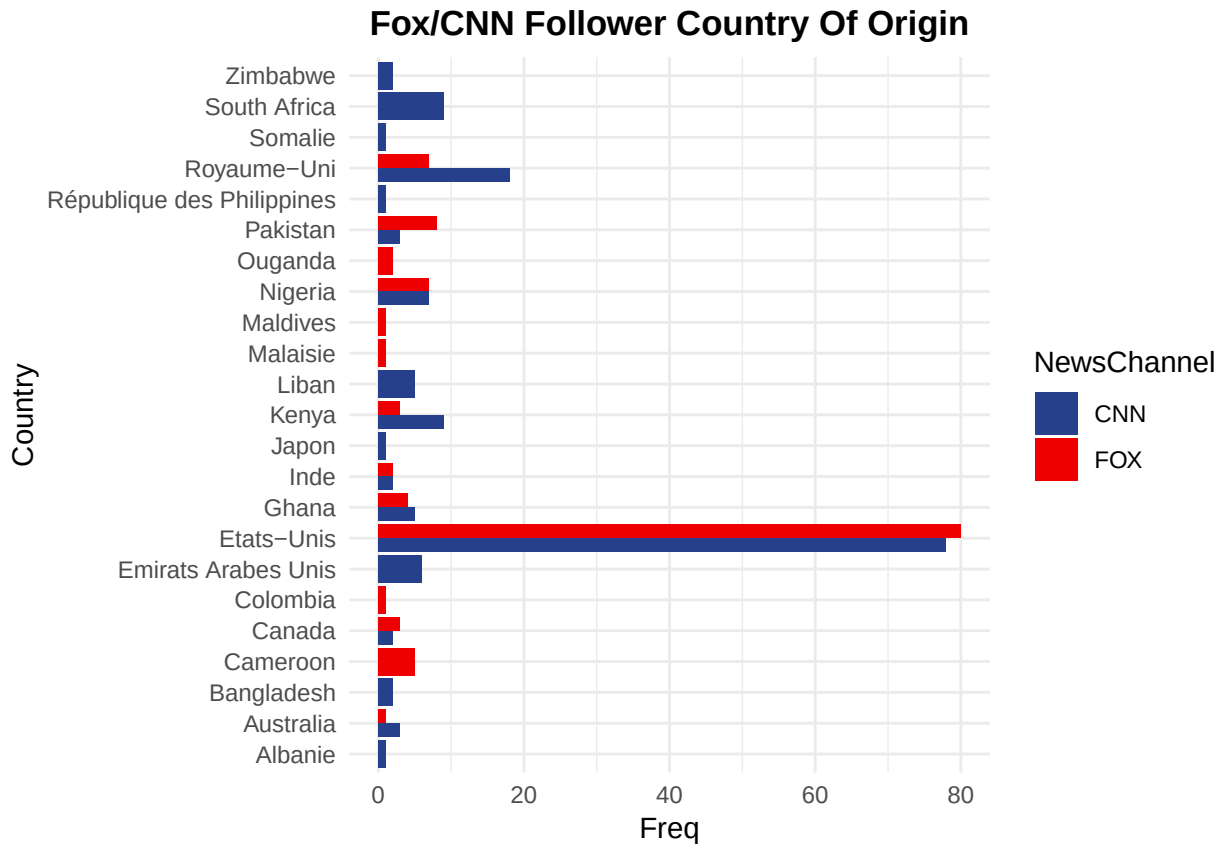
Religion Distribution

This entity type had the least number of distribution among the 5 entities. Entities reveal religious affliated terms.



News Channel Followers Country of Origin

Data revealed that most of the followers for both news channels came from the **USA** however in general **CNN** had more followers revealing their country of origin than FoxNews followers.



c) Limitations in dictionary based approach

Since the dictionary lookup matches were purely done using strict match unigram's a lot of entities were lost in the process.That is missplet tokens,bi-grams or words put within context didn't not apply in the dictionary approach used hence this could inturn lead to potential wrong entity recognition.

Machine Learning approach

As the next level of analysis of entity recognition , we went for the machine learning approach where we used the labelled data as part of Dictionary lookup for training the data. We tried two methods CRF(Conditional Random fields) and Random forests.

CRF(Conditional Random Fields)

Model Building

For CRF we first took the labelled data from the Dictionary look up approach and this labelled entity data was used for Training the model using CRF suite package. We took CNN data as training data with the columns docid, token, UPOS and entity and then added the tag of the preceding and the next term for both Parts of Speech and the tokens and then divided the data into training and validation set (80:20 ratio). The model was trained using L-BFGS with L1/L2 regularization.

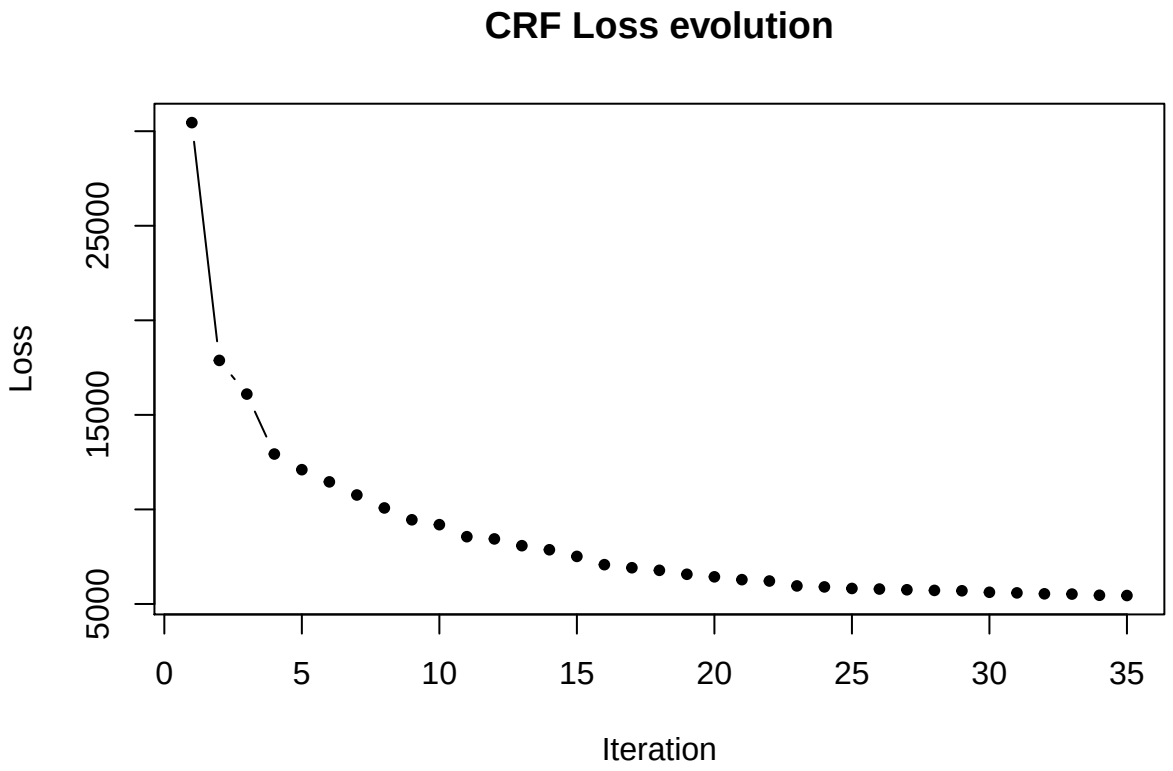
```
# CRF Model

model <- crf(y = CRF_model_train$Entity,
             x = CRF_model_train[, c("upos", "pos_previous", "pos_next",
                                     "token", "token_previous", "token_next")],
```

```
            group = CRF_model_train$doc_id,
            method = "lbfgs",
            options = list(max_iterations = 35))
```

The below loss function graph shows the error loss with iterations,we can see that after 25 iterations the loss becomes stagnant

```
## Summary statistics of last iteration:
## Loss: 5447.809236
## Feature norm: 40.977626
## Error norm: 161.031450
## Active features: 23539
## Line search trials: 2
## Line search step: 0.181836
## Seconds required for this iteration: 0.108
##
## Dumping summary of the model to file C:\Users\akirika\AppData\Local\Temp\RtmpwxfWBQ\crfsuite_25ec45a
```



Random Forests

For this final model, we combine an approach based on memory of previously seen words, word featurization, context information, and classify the resulting observations using a Random Forest classifier. The memory-based part is obtained by learning the most common entity identifications for each word. Once it is trained, if a new word has already been seen, it will be classified as it was in the past.

In addition to this major feature, other information are drawn from the word itself. It includes whether the word contains uppercase, lowercase letters, whether it is in the title format, how many characters it is made of and whether there is digits within the word. The part of speech tag is also retrieved. These features are

also retrieved for the preceding word and the following one, and combined to form a feature space.

Finally, these variables are used to predict the named entity using a random forest classifier.

Model Evaluation for CRF and Random Forests

Finally below are the results achieved as part of both CRF and Random forests

The models applied on the validation set of CNN gives the below confusion matrix and the F1,Precision, recall and accuracy values of the models (CRF and RandomForests)

```
## [1] "CRF Confusion Matrix"
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction     Hobbies Irrelavant Organization Personality Profession
##   Hobbies           65          0            0           0          0
##   Irrelavant        67       5521           21          57        230
##   Organization       0          0            4           0          0
##   Personality        0          0            0          49          0
##   Profession         0          0            0           0        254
##
## Overall Statistics
##
##                Accuracy : 0.9402
##                  95% CI : (0.934, 0.9459)
##     No Information Rate : 0.8808
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6438
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Hobbies Class: Irrelavant Class: Organization
## Precision                   1.00000            0.9364           1.0000000
## Recall                      0.49242            1.0000           0.1600000
## F1                          0.65990            0.9672           0.2758621
## Prevalence                  0.02106            0.8808           0.0039885
## Detection Rate              0.01037            0.8808           0.0006382
## Detection Prevalence        0.01037            0.9407           0.0006382
## Balanced Accuracy           0.74621            0.7490           0.5800000
##                      Class: Personality Class: Profession
## Precision                      1.000000           1.00000
## Recall                         0.462264           0.52479
## F1                             0.632258           0.68835
## Prevalence                     0.016911           0.07722
## Detection Rate                 0.007817           0.04052
## Detection Prevalence           0.007817           0.04052
## Balanced Accuracy              0.731132           0.76240
```

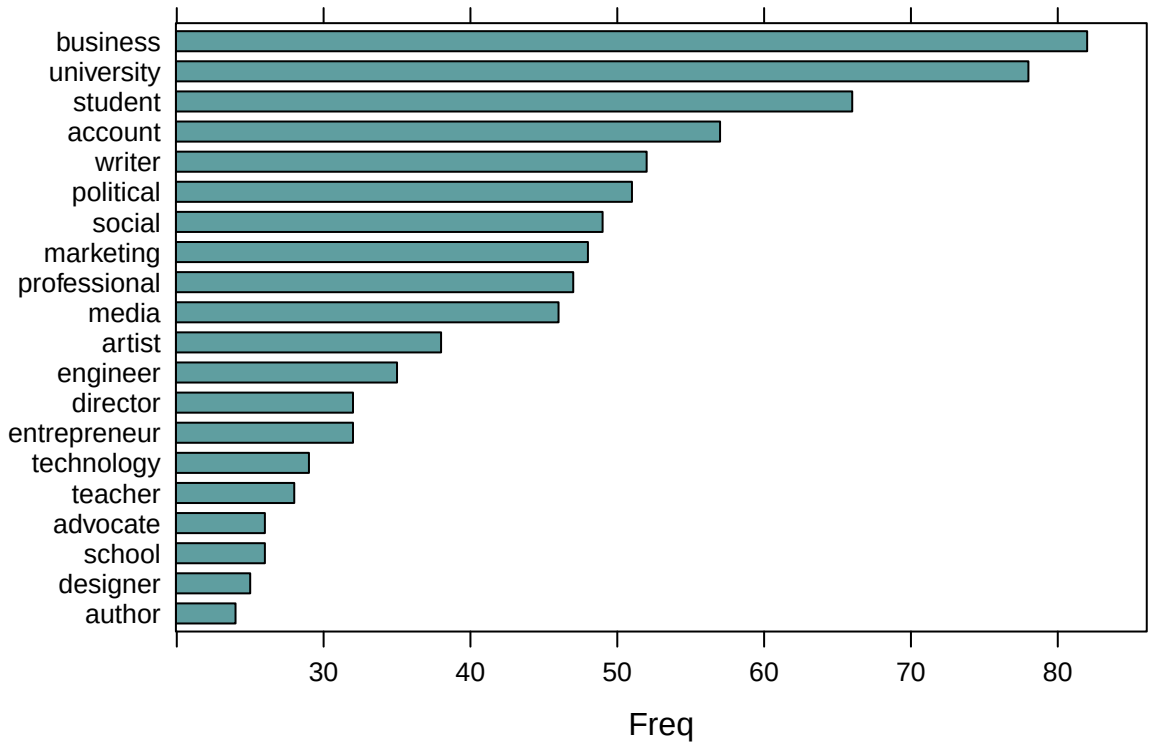
```
## [1] "Random Forest Confusion Matrix"
```

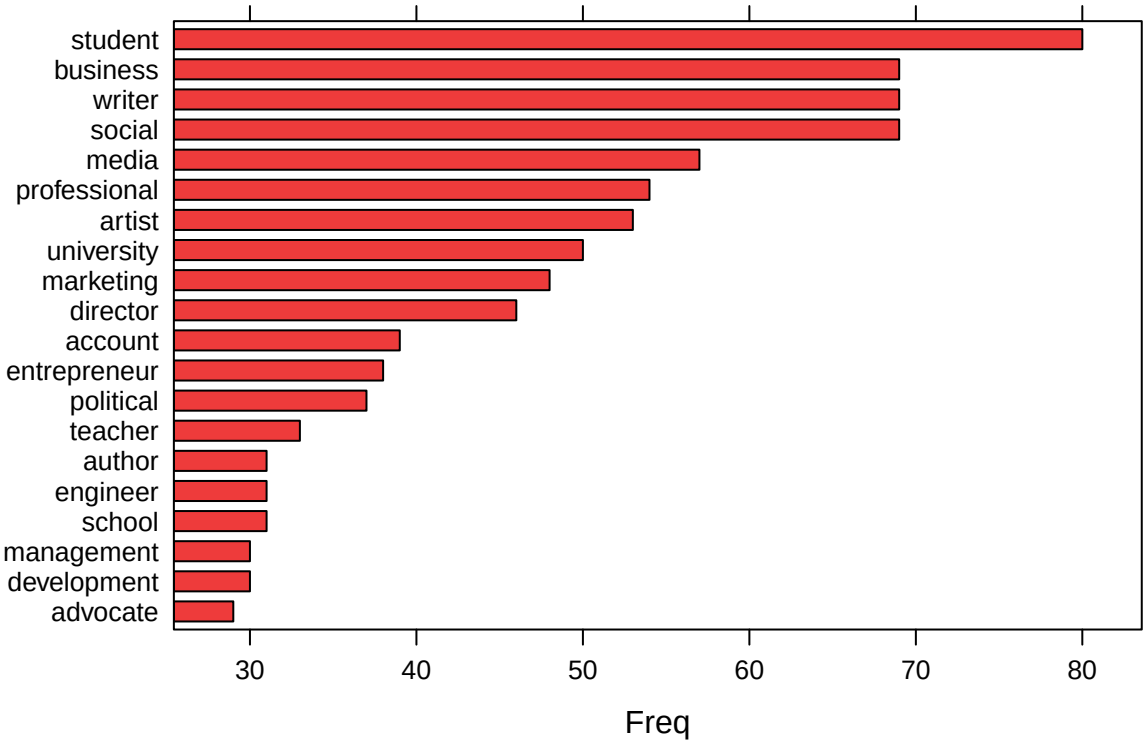
```
## Confusion Matrix and Statistics
##
##               Reference
```

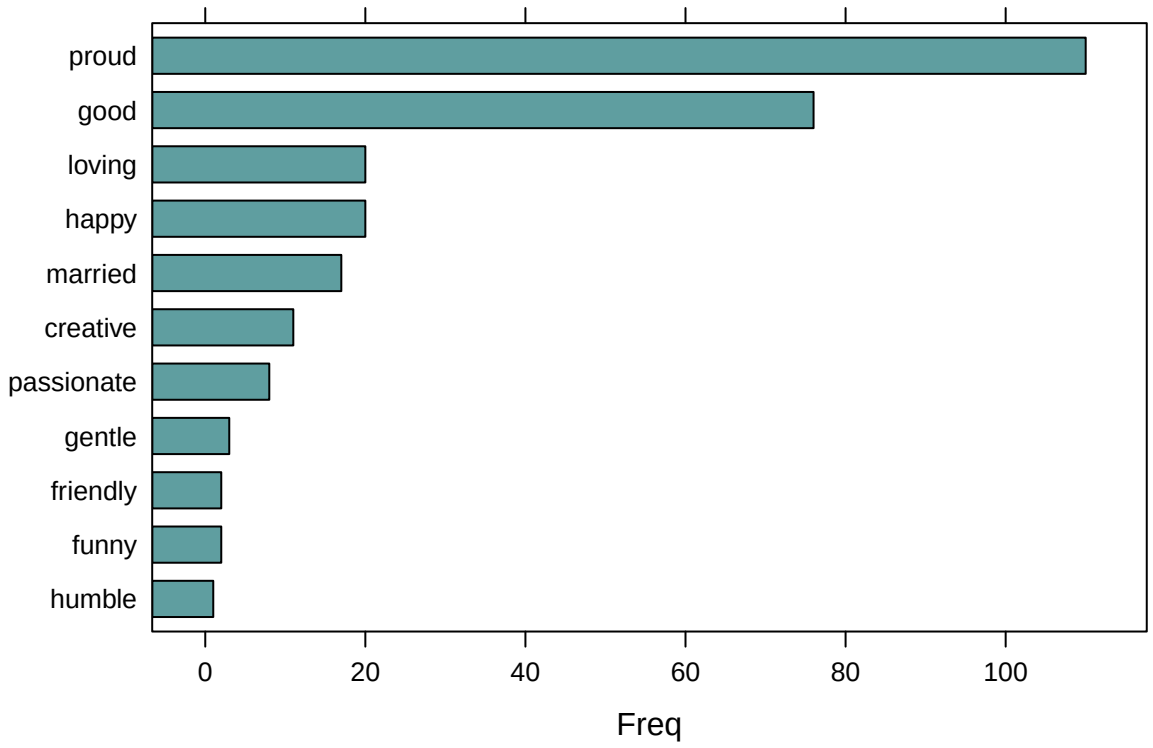
```
## Prediction     Hobbies Organization Personality Profession Religion
##   Hobbies          559            0           1          0        0
##   Organization       0          111           0          0        0
##   Personality        0            0         456          0        0
##   Profession         0            0           0       2267        0
##   Religion           0            0           0          0       21
##   Unassigned        22           24          21         41        1
##               Reference
## Prediction     Unassigned
##   Hobbies               0
##   Organization          0
##   Personality           0
##   Profession            0
##   Religion              0
##   Unassigned        27832
##
## Overall Statistics
##
##                Accuracy : 0.9965
##                  95% CI : (0.9958, 0.9971)
##     No Information Rate : 0.8876
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9827
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Hobbies Class: Organization Class: Personality
## Precision                   0.99821            1.000000            1.00000
## Recall                      0.96213            0.822222            0.95397
## F1                          0.97984            0.902439            0.97645
## Prevalence                  0.01853            0.004305            0.01524
## Detection Rate              0.01783            0.003540            0.01454
## Detection Prevalence        0.01786            0.003540            0.01454
## Balanced Accuracy           0.98105            0.911111            0.97699
##                      Class: Profession Class: Religion Class: Unassigned
## Precision                      1.00000       1.0000000            0.9961
## Recall                         0.98224       0.9545455            1.0000
## F1                             0.99104       0.9767442            0.9980
## Prevalence                     0.07361       0.0007016            0.8876
## Detection Rate                 0.07230       0.0006697            0.8876
## Detection Prevalence           0.07230       0.0006697            0.8911
## Balanced Accuracy              0.99112       0.9772727            0.9845
```

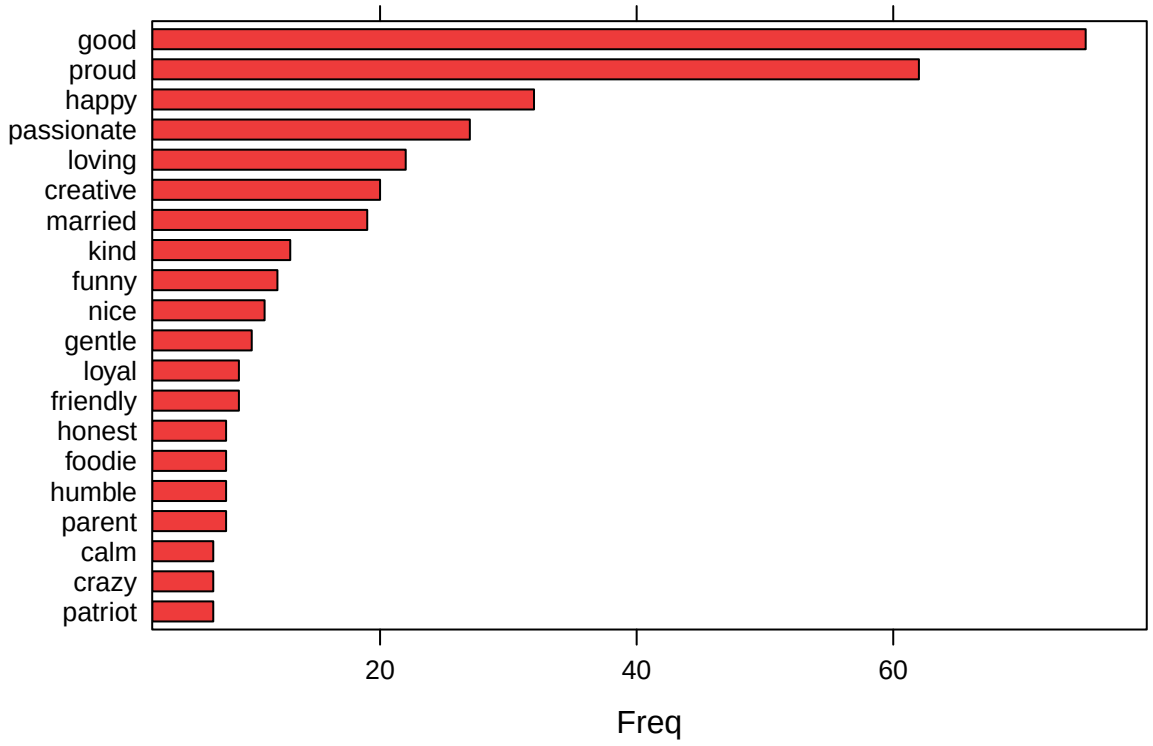
Prediction of the models (CRF and Random Forests) on a new data set - Fox News

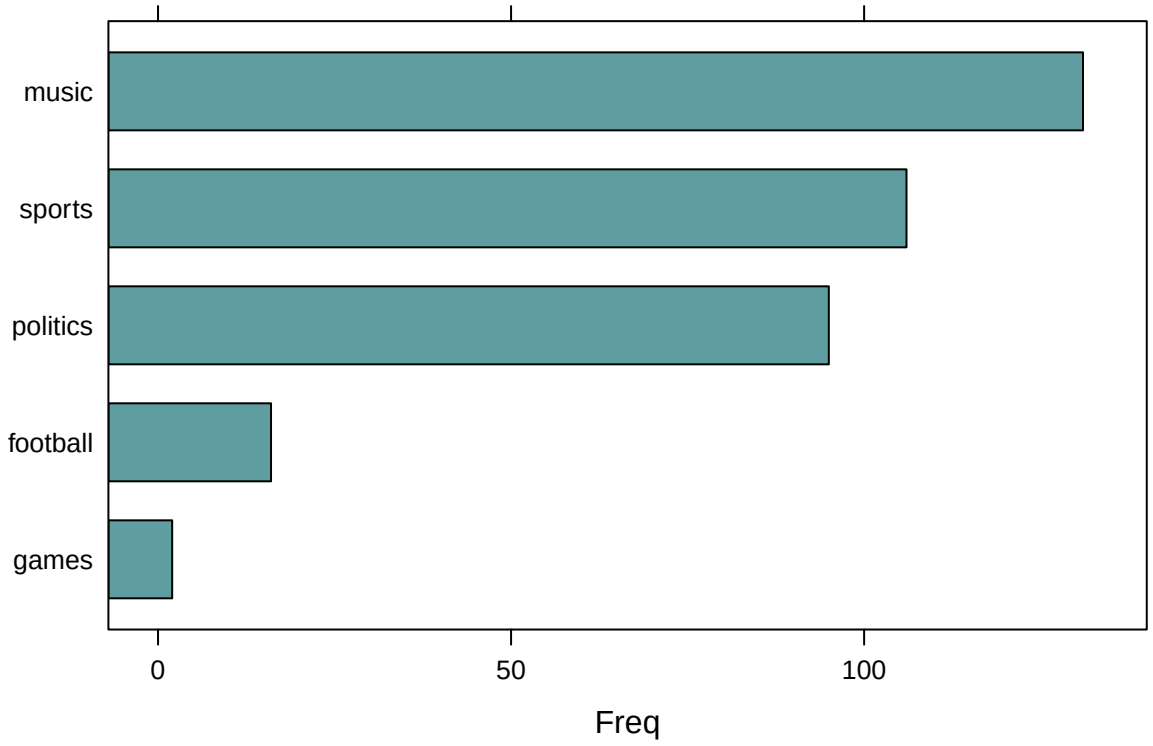
After evaluating the models on the validation set, the models were applied on Fox news data and the results were pretty good.Below graphs show the predicted Entities for Fox news after applying the models CRF and Random forests

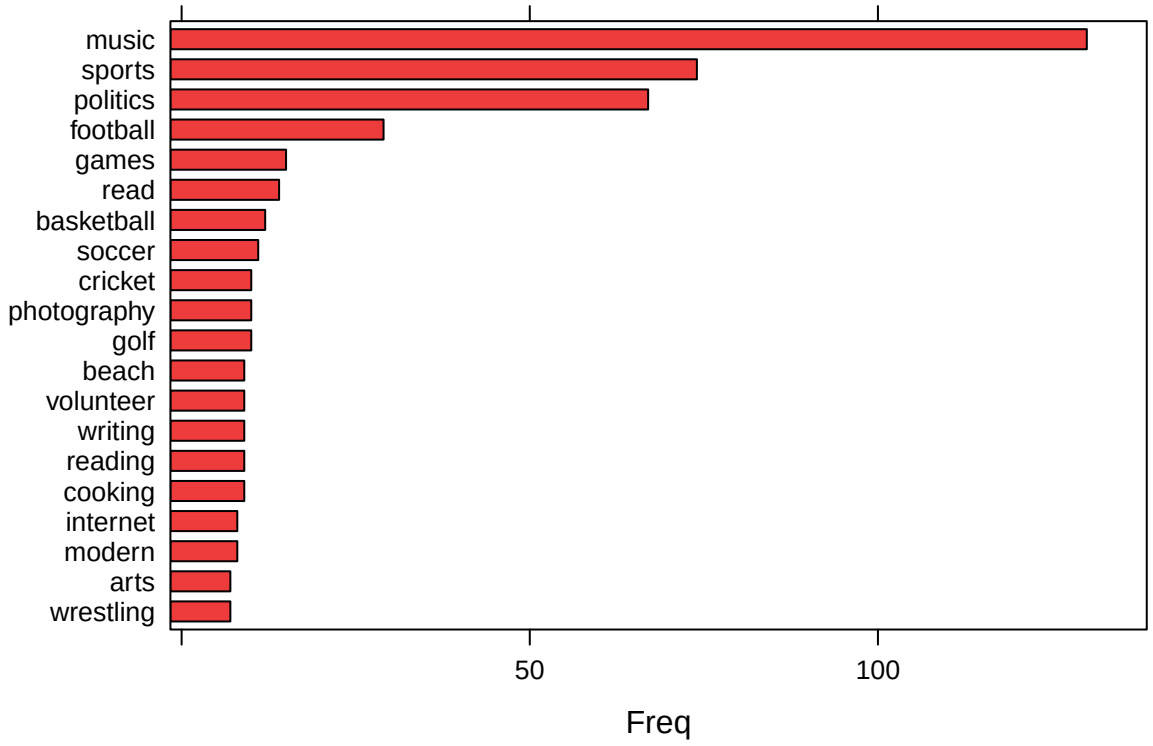

Predicted Profession in Fox news (CRF)
business
university
student
account
writer
political
social
marketing
professional
media
artist
engineer
director
entrepreneur
technology
teacher
advocate
school
designer
author
30
40
50
60
70
80
Freq

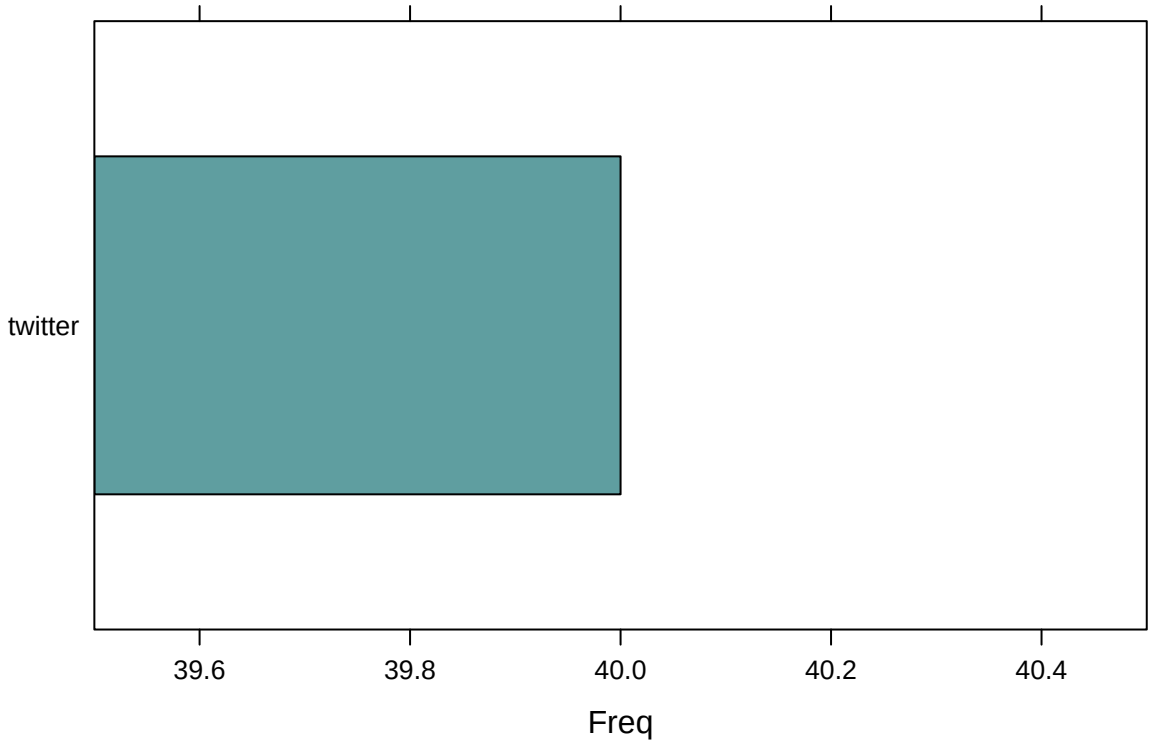

Predicted Profession in Fox news (Random Forest)
student
business
writer
social
media
professional
artist
university
marketing
director
account
entrepreneur
political
teacher
author
engineer
school
management
development
advocate
30
40
50
60
70
80
Freq

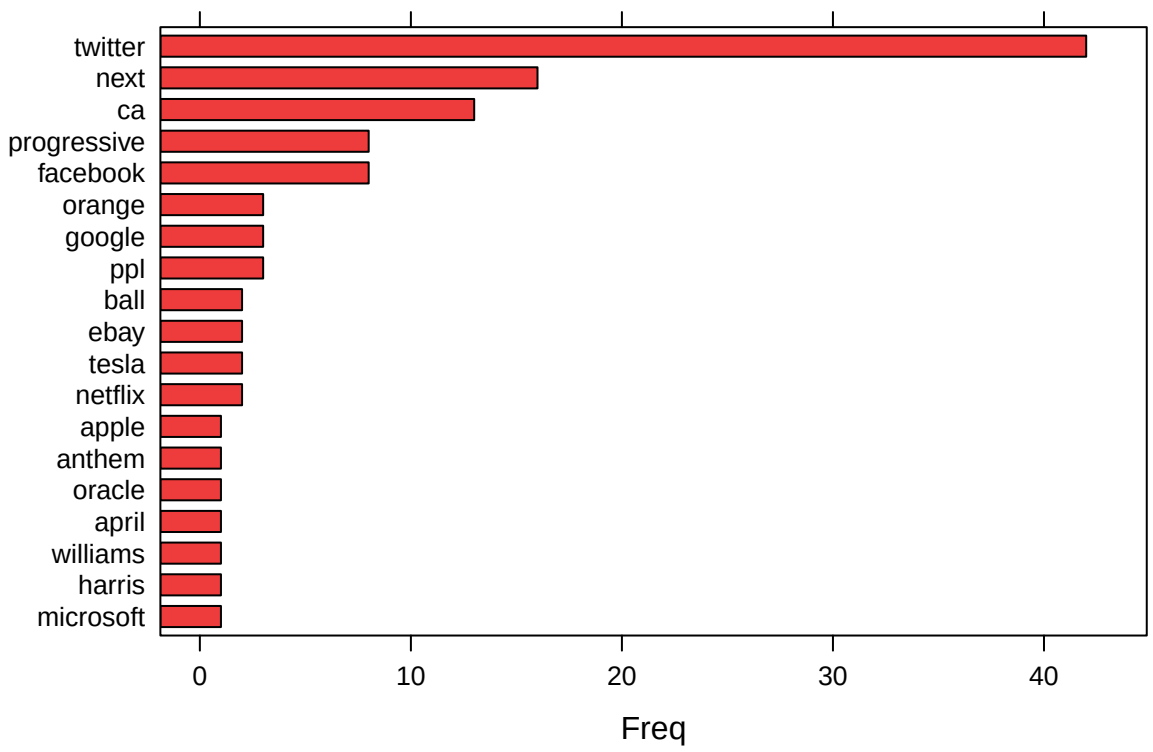

Predicted Personality in Fox news (CRF)
proud
good
loving
happy
married
creative
passionate
gentle
friendly
funny
humble
0
20
40
60
80
100
Freq


Predicted Personality in Fox news (Random Forest)
good
proud
happy
passionate
loving
creative
married
kind
funny
nice
gentle
loyal
friendly
honest
foodie
humble
parent
calm
crazy
patriot
20
40
60
Freq


Predicted Hobbies in Fox news (CRF)
music
sports
politics
football
games
0
50
100
Freq


Predicted Hobbies in Fox news (Random Forest)
music
sports
politics
football
games
read
basketball
soccer
cricket
photography
golf
beach
volunteer
writing
reading
cooking
internet
modern
arts
wrestling
50
100
Freq


Predicted Organisation in Fox news (CRF)
twitter
39.6
39.8
40.0
40.2
40.4
Freq



Inference

CRF - The results from CRF shows us that they are more efficient and give almost accurate predictions if the data is labelled properly. Hence Machine learning approach with a dictionary lookup labelled data can give us more precise results for named entity recognition in NLP.

Random Forests - The results from the three approaches developped in the Python Notebook, around memory-based classification and random forests, give interesting insights into which method should be prioritized. The simplistic memory-based approache gives impressively good results on an unseen dataset, and allows classification of most of the entities. It is possible that these results are due to the similarity of the datasets between CNN & Fox, both having a similar structure and both coming from Twitter descriptions. However, a simple memory recognition may not be sufficient to generalize to uncommon data, hence the need to extract features from the words themselves to draw extra information, not dependent on whether the word has already been seen or not.

Using a random forest classifier on this data only showed extremely poor results.

Finally, a combination of memory-recognition and featurization on both the word itself and its surrounding words proved to be very effective. Moreover, this approach may very well offer more generalization potency for applications outside the 2 datasets studied. Overfit should still be watched out for, as the model relies heavily on memory-recognition.

In our analysis CRF and Random forests almost have the same accuracy, but other evaluation metrics such as precision, recall, F1 are higher for random forests classifier when combined with the approach of surrounding words similar to CRF. Hence we can say that a combination of random forest classifier with the logic of featurisation of surrounding words would be one of the good methodologies for Entity recognition.

Conclusion

Throughout our approach, we worked on several ways of performing Named Entity Recognition on big datasets, which does not allow for human labeling.Named Entity Recognition is very dependent on the context and models can hardly generalize to different cases. This is probably why custom models trained on case-specific data performed very well compared to the approaches tried in the first instance (out-of-the-box NER models) in our analysis. To tackle any new problem statement with regards to entity recognition, it may therefore be relevant to adopt a combination of dictionary-based tagging followed with building customised models using machine learning with existing algorithms. Also, the model developed in this context of entity recognition applied to different analysis like analysis of job descriptions might perform the same or bad. Hence with named recognition more research might be required on developing models using machine learning or any other approaches that would suffice all scenarios.