International Journal of Research in Engineering, Science and Management
Volume 4, Issue 1, January 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

Named Entity Recognition in Tweets

Lakshya Rajoria*
Jaipur, India

***Abstract*: Named Entity Recognition (NER) is part of Natural Language Processing (NLP) and is a form of information extraction that helps locate and classify named entities in unstructured text into categories such as locations, people, organizations etc. While the performance of conventional NLP tools is rigorous for formal pieces of literature such as articles, it is severely degraded in the noisy, informal corpus of 280 character messages that are tweets. That, coupled with the insufficient information in a tweet, named entities being out-of-vocabulary (OOV) and lack of training data makes NER all the more challenging. Recently, several works have been posited to tackle NER including implementing part-of-speech or POS tagging which would identify entities as verb, noun etc. phrases, Conditional Random Fields (CRFs), normalization and other forms of distant supervision or unsupervised learning. In this paper, I propose conducting domain adaptation where the Broad twitter corpus or BTC (Derczynski et al. 2016) is preferred as a means for training, development and test data over the Ritter et al. 2011 dataset. The former is not only significantly bigger than the latter but is also sampled across different regions, time periods, and types of Twitter users. To further delve into the consideration of named entities, we use domain transfer by modifying the corpus from Ritter et al. 2011 to match the 3 named entities specified in the BTC (Person, Location, Organization) and using algorithms put forward in Ritter to evaluate the BTC data. In addition to the BTC data, we will evaluate the results on our own baseline Indian tweets data. Using these new datasets, we hope to test state-of-the-art natural language processing algorithms and machine learning algorithms. We demonstrated that our proposed method of evaluating Ritter algorithms on the BTC and Indian tweets increased the FB1 score by 34.69 (BTC Development DataSet) and 6.65 respectively when compared with the tests run using Ritter train data.**

***Keywords*: Broad Twitter Corpus, Indian tweets, Named entity recognition, Natural Language Processing, Semi-Supervised Learning, Twitter.**

1. Introduction

For humans, the task of being able to identify and name entities from unstructured text is not difficult at all— whether it be recognizing a city, a person or a company by its name or description. But the pace of technology in keeping up with the same level of computational ability humans possess is still lagging. The advent of social media conglomerates like Facebook, Twitter, and Instagram and the creation of new text types such as status messages and user posts have posed challenges for language technology because of the aforementioned informal and noisy nature that is inherent to these text types. Still, because of the easily accessible platform of tweets, they can provide information that is more up-to-date and a faster mode of communication than news articles. The application of named entity recognition for tweets then can become powerful search algorithms used to determine relevant tags for millions of tweets and help enable a smoother discovery of different content, such as news on COVID-19, specific location based information, or even specific people. That is why the rapidly increasing number of tweets in existence warrants data-mining through NER and information extraction.

Currently, there are less than 100k tokens publicly available with the added constraints of high-performance systems, such as those put forward by Liu et al. 2012, not being available for evaluation and thus not reproducible, single-annotators and low-levels of agreements between multiple annotators (Derczynski et al., 2016).

Prior research in this field is dominated by distant supervision and semi-supervised learning algorithms. A commonly used dataset in Twitter NER is the Ritter dataset, which has been used for rebuilding the NLP pipeline (Ritter et al. 2011), KNN algorithm and Conditional random fields (Liu et al. 2011). Refer to more prior research under relevant works.

1	#BoycottFakeStars Unfollow this fake stars who come in industry with the help of his family nor his/her talent. Like sushant singh rajput, rajkumar rao nd many more come in film industry bcz of their talent nor from their back. #BoycottFakeStars
2	When someone wants to make his name have a stardom through tiktok. why don't u let tht happen? It isn't to gain fandom through tiktok. it maybe easy to get one vedio viral but maintaining it. gaining popularity isn't tht easy as u think #TrollingCauseDepression
3	I don't know how many people remember the importance of this day! But two years ago on this day we lost 18 soldiers including my friend Gangadhar Dalui in #UriAttack Paying homage to all d fallen soldiers & taking pledge to support their families within our limits! #UriMartyrsDay

Fig. 1. Example of Noisy (Informal) Text from Tweets sourced from Twitter India. Exemplary of the more challenging task of entity recognition in an environment where slang is the norm and data is not as structured as news and longer articles of text

Machine learning is vital in the case of conducting Twitter NER because of the limited amount of annotated data. If a computer has the ability to learn from training sets without being under-fitted or overfitted and still have a satisfactory F-1 score on different test datasets, then Twitter NER becomes more manageable. The combination of a supervised and unsupervised learning system will allow for the computer to adjust to data it has not annotated before and still produce satisfactory results.

*Corresponding author: lakshyarajoria@gmail.com

2. Related Work

Named entity recognition has been vastly researched and its solutions can be categorized into rule based (Krupka and Hausman 1998), machine learning based (Finkel and Manning, 2009; Ritter et al. 2010; Liu et al. 2011) and hybrid methods (Jansche and Abney, 2002). Still, today data-driven methods have increasingly become the norm.

A. NER on non-twitter corpora

Heuristic-based approaches dominated the field of NER in the 20th century, but Bikel et al. was crucial in revolutionizing conventional approaches to become flexible with each new source of text by employing learning algorithms. Handcrafting finite state patterns such as <proper-noun>+ <corporate designator> ==> <corporation> for recognizing names, locations etc. did not take into account typical naming conventions such as how organizations choose to have names representative of the industry they are in or the type of service/good they are offering. For this reason, Bikel et al. reasoned that to prevent the use of excessive resources being allocated for fitting rules to different data and to minimize the significant tweaking that was required with the introduction of each new text, a hidden Markov model needed to be implemented. With this method they were able to construct a bigram language model that would compute the likelihood of a sequence of words by associating a probability with each transition to the next word from the current word.

Currently, research into NER has mostly focused on formal texts such as news articles (Mccallum and Li, 2003) but has also diversified into the biomedical NER systems with Yoshida and Tsujii's 2007 publication utilizing shallow parsing and POS tagging and orthographic features. In another case, because of elements of supervision on Twitter-based NER approaches requiring the availability of labeled data, which was lacking, Finin et al. 2010 proposed an alternative. He used a crowd-sourcing way utilizing Amazon Mechanical Turk Services and CloudFlower to prepare labeled data and trained a CRF model for testing the effectiveness of human done labeling. Still, NER has gained renewed interest from the challenging task posed by tweets.

B. Rebuilding the pipeline

Ritter et al. 2011 presents a novel way of "rebuilding the NLP pipeline" through POS tagging, shallow-parsing or chunking and NER. This was significant as it outperformed the conventional Stanford NER system, which, because of its unreliable capitalization and misclassification of nouns and proper nouns, posed large losses in performance metrics. His model would first make use of Conditional Random Fields for named entity segmentation as a sequence labeling task and then a distantly supervised approach applying LabeledLDA so as to add constraints from Freebase, an open-domain database, on the dataset as a form of supervision and to classify named entities.

The outputs derived from these two NLP tasks can be then used for the feature generation for named entity recognition.

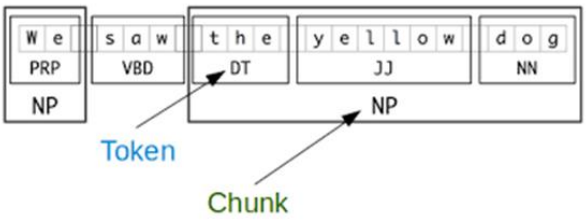

Fig. 2. Shallow Parsing used to identify non-recursive phrases such as noun phrases, verb phrases and prepositional phrases

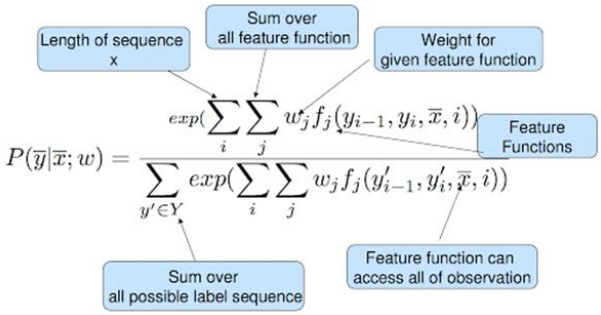

Fig. 3. The POS tagging task utilizes CRFs for inference and learning

Named Entity Classification Algorithms:

for each type: $t = 1 \dots T$ **do**
  Generate β_t according to symmetric Dirichlet distribution Dir(η).
end for
for each entity string $e = 1 \dots |E|$ **do**
  Generate θ_e over $FB[e]$ according to Dirichlet distribution Dir($\alpha_{FB[e]}$).
  for each word position $i = 1 \dots N_e$ **do**
    Generate $z_{e,i}$ from Mult(θ_e).
    Generate the word $w_{e,i}$ from Mult($\beta_{z_{e,i}}$).
  end for
end for

Fig. 4. Generative process for Named Entity Classification

Algorithm 1: Collapsed Gibbs sampling for LDA.

initialize $\mathbf{z}$ at random, e.g., $z_{di} \sim \text{Multinomial}(1/T)$;
$\mathbf{a}_{dt} = |\{i : z_{di} = t\}| + \alpha$;
$\mathbf{b}_{wt} = |\{(d,i) : x_{di} = w, z_{di} = t\}| + \beta$;
$\mathbf{c}_t = |\{(d,i) : z_{di} = t\}| + |W|\beta$;
repeat
  forall $d \in D$, $i \in \{1 \dots N_d\}$ **do**
    $t \leftarrow z_{di}$; $w \leftarrow x_{di}$; $\mathbf{a}_{dt}$--; $\mathbf{b}_{wt}$--; $\mathbf{c}_t$--;
    $t \sim \text{Multinomial}(\mathbf{a}_d \mathbf{b}_w / \mathbf{c})$;
    $z_{di} \leftarrow t$; $\mathbf{a}_{dt}$++; $\mathbf{b}_{wt}$++; $\mathbf{c}_t$++;
until *convergence* ;

Fig. 5. Collapsed Gibbs Sampling

To infer values for hidden variables, Ritter et al. makes use of Collapsed Gibbs Sampling (see figure 5), which estimates the posterior distribution over types slightly better than the application of the Bayes Rule. Predictions are then made using the Dirichlet distribution, as indicated in Figure 4, and 100 iterations of the Gibbs Sampling, which as specified before, holds the hidden topic variables in the training data.

Semi Supervised Learning:

Liu et al. 2011 propose another multifaceted approach regarding the implementation of normalization of tweets. In this method, the model would correct "ill-formed words" using a global linear model, combination of K-Nearest Neighbours or

KNN algorithm with a linear conditional random fields (CRFs) model and a semi supervised learning framework that makes up for the lack of training data. The K-Nearest-Neighbors algorithm is used for pre-labeling the over 12k tweet corpus, which is then used as the input for the CRF model in performing sequential labeling. With the added introduction of 30 gazetteers-representing general knowledge across a host of different domains-into the mix, the method Liu proposes aims to combine global information from KNN and gazetteers with contextual information from the tweets to subsidize the lack of training data.

Algorithm 1 NER for Tweets.

Require: Tweet stream i; output stream o.
Require: Training tweets ts; gazetteers ga.

Initialize l_s, the CRF labeler: $l_s = train_s(ts)$.
Initialize l_k, the KNN classifier: $l_k = train_k(ts)$.
Initialize n, the # of new training tweets: $n = 0$.
while Pop a tweet t from i and $t \neq null$ **do**
  for Each word $w \in t$ **do**
    Get the feature vector $\vec{w}$: $\vec{w} = repr_w(w, t)$.
    Classify $\vec{w}$ with knn: $(c, cf) = knn(l_k, \vec{w})$.
    if $cf > \tau$ **then**
      Pre-label: $t = update(t, w, c)$.
    end if
  end for
  Get the feature vector $\vec{t}$: $\vec{t} = repr_t(t, ga)$.
  Label $\vec{t}$ with crf: $(t, cf) = crf(l_s, \vec{t})$.
  Put labeled result (t, cf) into o.
  if $cf > \gamma$ **then**
    Add labeled result t to ts , $n = n + 1$.
  end if
  if $n > N$ **then**
    Retrain l_s: $l_s = train_s(ts)$.
    Retrain l_k: $l_k = train_k(ts)$.
    $n = 0$.
  end if
end while
return o.

Algorithm 2 KNN Training.

Require: Training tweets ts.

Initialize the classifier l_k:$l_k = \varnothing$.
for Each tweet $t \in ts$ **do**
  for Each word,label pair $(w, c) \in t$ **do**
    Get the feature vector $\vec{w}$: $\vec{w} = repr_w(w, t)$.
    Add the $\vec{w}$ and c pair to the classifier: $l_k = l_k \cup \{(\vec{w}, c)\}$.
  end for
end for
return KNN classifier l_k.

Algorithm 3 KNN predication.

Require: KNN classifier l_k ;word vector $\vec{w}$.

Initialize nb, the neighbors of $\vec{w}$: $nb = neigbors(l_k, \vec{w})$.
Calculate the predicted class c^*: $c^* = argmax_c \sum_{(\vec{w}', c') \in nb} \delta(c, c') \cdot cos(\vec{w}, \vec{w}')$.
Calculate the labeling confidence cf: $cf = \frac{\sum_{(\vec{w}', c') \in nb} \delta(c, c') \cdot cos(\vec{w}, \vec{w}')}{\sum_{(\vec{w}', c') \in nb} cos(\vec{w}, \vec{w}')}$.
return The predicted label c^* and its confidence cf.

Fig. 6. Algorithms used in Liu et al. 2011 for NER

In a later publication, building up from her research, Liu et al. 2012 constructs a named entity normalization method for tweets that would allow for more efficient and accurate entity recognition and thus account for the variations of NEs in tweets. This proves to be successful in increasing the F1 score by a margin of 3.4% from the baseline as it implements NER and NEN jointly using a factor graph as their model as shown in Figure 7. Through this, they are also able to limit the number of errors propagating from the entity recognition to the named entity normalization (NEN) task.

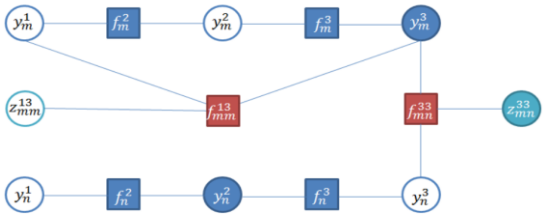

Fig. 7. Factor Graph NER and NEN on tweets. Blue circles represent NE type, and green circles represent normalization variables; circles that are filled are observed random variables and blue rectangles represent the factors connecting neighboring y-serial variables; and red rectangles represent factors connecting distant y-serial and z-serial variable (Liu et al. 2012)

Shubhanshu Mishra and Jana Diesner also took a similar approach to Liu et al. with a semi-supervised NER system and CRFs but also introduced "leverage random feature dropout for up-sampling the training data," as can be seen in figure 8, which allows for understanding new tokens into the system via unsupervised learning. Furthermore, we can analyze the empirical analysis from Derczynski et al. 2015 for named entity recognition and disambiguation to see different systems' performances on noisy texts. They concluded that the most significant drop in the performance of NER approaches comes from poor capitalization and that slang contributes only a minor drop in the performance readings. Despite that, some improvements did come with micro-blog trained POS tagging and normalization.

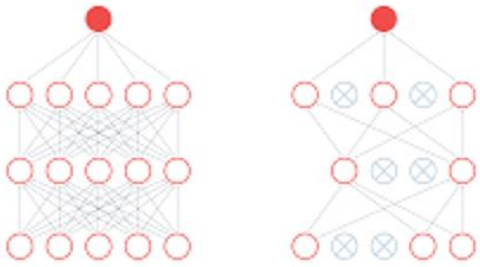

Fig. 8. Random Feature Dropping of interaction and lexical features which is the classifier Mishra used and has large weights. Used to create a large number of noisy samples without re-weighing the feature weights using the dropout probability

3. Data

The data that will be used in this research will be based on the Broad Twitter Corpus (BTC; Derczynski et al. 2016). The commonly used Ritter dataset, in comparison, has a mere 45,000 tokens, which is just 15% the size of the CoNLL'2003 dataset that is popular for news NER. Because of the need for a sizable, highly diverse and quality annotation dataset, the BTC boasts gold-standard named entity annotations from both NLP experts and crowd workers. Socially segmented for non-professional content and news, it is also stratified for time over a six-year period and for different places across the world that account for the different English variations. We use the

recommended training and development test split of section H and use the entirety of section F for testing. Section H is stratified for the month, time of day and day of the week, allowing for copious amounts of varied data across "temporal cycle types" (Derczynski et al. 2016). Section F offers content from individuals providing twitter-based commentary—or the twitterati—and is stratified across regions of the UK and authors from backgrounds ranging from sports and journalism to music and politics.

Corpus	Tokens	Entity schema	Annotator type	Annotator qty.	Notes
Finin et al. (2010)	7K	PLO (3)	Crowd only	Multiple	Low IAA (Fromreide, 2014)
Ritter et al. (2011)	46K	Freebase (10)	Expert	Single	IAA unavailable
Liu et al. (2011)	12K	PLO (3) + Product	?	?	Private corpus
Rowe et al. (2013)	29K	PLO (3) + Misc	Expert	Multiple	No hashtags/usernames
Broad Twitter Corpus	165K	PLO (3)	Expert + Crowd	Multiple	Source JSON available

Fig. 9. Comparison of Different Openly available Corpi (PLO is Person, Location, Organization)

	Feature	Count
Dataset	Documents	9 551
	Tokens	165 739
Entities	Person	5 271
	Location	3 114
	Organization	3 732
	Total	12 117

Fig. 10. BTC statistics

4. Error Analysis of Ritter Algorithms

Typical NER programs have indicated a variety of errors when employed on tweets. Through the error analysis performed on the Ritter datasets by Derczynski et al. 2015, it can be observed that the Ritter model has several main shortcomings:

The primary problem is entity drift. The annotations style behind the creation of the Ritter datasets is a single-annotator corpus, introducing an inherent bias that makes it harder to discern statistically significant differences in results. Moreover, because of the contiguous time period in which the data was collected, the lack of stratification presents more problems in accurately conducting NER as there is a lack of differentiation in the data.

Another problem was in the examination of only the text part of individual tweets. Twitter users often use sources of context outside from the post they are making and so looking at the text in isolation effectively removes this context making NER more challenging. That is why there were many cases of missed entity recognitions and false positives as can also been seen in figure 11. Further, typographic errors can skew results from pre-linking stages such as tokenization, as they lack balance between ignoring correct OOV words and fixing mistyped in-vocabulary words. On this point, shortenings of entities also leads to abbreviations and prevalent use of pronouns. Still, as we understand, with the introduction of a much larger and richly annotated dataset (the Broad Twitter Corpus) the extent of this problem decreases.

Another major source of mistakes stems from capitalization. Often, due to the informal nature of tweets, typical indicators of an entity in English, such as capitalization in the middle of a sentence, are missed or added to non-entities. Capitalization can also be essential in distinguishing between common nouns, proper nouns etc. which would make NER more practical especially in the case of polysemous nouns such as "an apple" vs. "Apple"—the company.

Confusion matrix for Stanford NER performance over the evaluation partition of the Ritter dataset.

Tokens	Gold				
	Loc.	Misc.	Org.	Person	O
Response					
Location	65	3	8	5	28
Misc	9	42	14	6	72
Organization	5	2	18	2	27
Person	9	6	6	87	34
O	20	25	26	14	8974

Fig. 11. Illustrating what entities were misclassified to using Stanford NER tool

5. Baseline Construction

In creating a baseline of our own we randomly sampled 150 tweets or 5425 tokens from Twitter's India platform. In doing so, we aimed to analyze a major English-speaking nation that was exempted from the Broad Twitter Corpus. In the creation of our baseline, we further attempted to understand named entity recognition in the regional differences inherent to the English language in India as compared to other countries analyzed in the BTC. Similar to the annotation system implemented in the BTC, our data was stratified to have content from a host of different entities such as celebrities, sports, news, and politics. Moreover, entity classifications were done with the same three aforementioned factors: person, location and company. Still, a limitation was that there was only single annotator.

In the case of polysemous entities, annotators would classify an entity after understanding it in the context that it is used. For example: *"...and success of dhoni movie the perfect dhoni for the movie…"*. Here "dhoni movie" references a movie about Mahendra Singh Dhoni but is not a person, whereas further ahead in the tweet, "the perfect dhoni for the movie" illustrates a person.

An important thing to note is that although there were measures taken to source tweets from different time periods, such as the 2016 Demonetization issued under the Modi administration, the Triple Talak controversy, and the URI surgical strike on Pakistan from India, the number of tweets collected in that past month have composed a far larger proportion of the baseline data. This invariably exposes the data to entity drift (Masud et al., 2010) where the selected entities may be prevalent currently but change in the future. For example, in the case of 2014, the Prime Minister of India was Manmohan Singh, but today, it is Narendra Modi. In another case, it might be veneration of a deity during Diwali in October and Santa during Christmas. Because of this, there is the possibility of overfitting the data as the model would be trained on data not stratified over different periods of time. As such, in the case of an introduction of a testing set from a different period of time, the results will not be at par.

Day of the month (June 2020)	14	15	16	17	18	19	20
Baseline	23	27	25	10	14	18	15

Fig. 12. Volume of tweets collected by day of month
Note: 5 Tweets on Demonetization (2016), 1 tweet on Triple Talak (2018), 3 tweets on Ram Mandir (2017), 5 tweets on Uri surgical strike (2016), 4 tweets from Pulwama attack (2019)

6. Learning Approach

The learning approach taken in this research paper will be utilizing the algorithms posited from Ritter and retraining them on the BTC dataset. In the Ritter approach, there is the use of conditional random fields for learning and inference, for instance, in the case of the named entity segmentation task. In the case of capitalization, Support Vector Machines are used for leaning with features including: fraction of words that had tweets capitalized, the fraction that appears in a dictionary as lowercase/or uppercase but are not so in tweets, and the frequency of the word 'I' appearing lower case and whether or not the first word is capitalized. This is really significant as Derczynski et al 2015 concluded that the largest drop in the performance of NER approaches comes from poor capitalization. With features based on capitalization, performance would improve at named entity segmentation (Ritter et al. 2011). Although the number of named entities was significantly larger in the Ritter paper (10), we will prefer the BTC dataset, which has 3 entities that are more clearly delineated.

7. Results

A. Reproducing Ritter Original Train and Development Sets

Statistics of Ritter Original Dataset Features

Number of data sets (groups): 1
Number of instances: 2393
Number of items: 46462
Number of attributes: 98251
Number of labels: 21

processed 16261 tokens with 661 phrases; found: 538 phrases; correct: 204.

Accuracy: 93.59%; Precision: 37.92%; Recall: 30.86%; FB1: 34.03.

Table 1
Results of NER using Ritter Train and development set

Named Entity	Precision	Recall	FB1	Number of entities labeled
company	42.86%	23.08%	30.00	21
facility	13.89%	3.16%	13.51	36
geo-loc	42.45%	0.86%	46.27	139
movie	12.50%	6.67%	8.70	8
Music artist	0.00%	0.00%	0.00	6
Other	32.95%	21.97%	26.36	88
person	47.52%	56.14%	51.47	202
product	4.76%	2.70%	3.45	21
Sports team	40.00%	5.71%	10.00	10
TV show	0.00%	0.00%	0.00	7

B. Derczynski BTC Train and Developmental Data Test with Ritter Algorithms

Statistics the data set(s):

Number of data sets (groups): 1
Number of instances: 998
Number of items: 14441
Number of attributes: 40575
Number of labels: 7

Processed 15002 tokens with 1732 phrases; found: 1618 phrases; correct: 1151.

Accuracy: 94.25%; precision: 71.14%; recall: 66.45%; FB1: 68.72

Table 2
Results of NER using BTC Train on BTC Dev Data

Named Entity	Precision	Recall	FB1	Number of entities labeled
LOC	48.15%	33.33%	**39.39**	108
ORG	42.75%	15.53%	22.78	138
PER	75.80%	86.96%	**81.00**	1372

C. Derczynski BTC Train and Test Set Data Test with Ritter Algorithms

Number of data sets (groups): 1
Number of instances: 998
Number of items: 14441
Number of attributes: 40575
Number of labels: 7

Processed 12308 tokens with 1462 phrases; found: 1345 phrases; correct: 930.

Accuracy: 93.62%; precision: 69.14%; recall: 63.61%; FB1: 66.26.

Table 3
Results of NER using Derczynski BTC Train on BTC Test Data

Named Entity	Precision	Recall	FB1	Number of entities labeled
LOC	45.39%	34.33%	**39.09**	152
ORG	46.97%	18.45%	26.50	132
PER	75.31%	86.38%	**80.46**	1061

D. Derczynski BTC Train and Indian Test Data Test with Ritter Algorithms

Number of data sets (groups): 1
Number of instances: 998
Number of items: 14441
Number of attributes: 40575
Number of labels: 7

Processed 5425 tokens with 502 phrases; found: 450 phrases; correct: 217.

Accuracy: 89.82%; precision: 48.22%; recall: 43.23%; FB1: 45.59

Table 4
Results of NER using Derczynski BTC Train on Indian Tweets Test Data

Named Entity	Precision	Recall	FB1	Number of entities labeled
Geo-Loc	52.91%	60.22%	56.33	206
Company	20.00%	10.81%	**14.04**	40
Person	49.02%	40.49%	**44.35**	204

E. Ritter Train modified and Indian Test Data Test with Ritter Algorithms

Number of data sets (groups): 1
Number of instances: 2393
Number of items: 46462
Number of attributes: 98251
Number of labels: 7

Processed 5425 tokens with 502 phrases; found: 212 phrases; correct: 139.

Accuracy: 89.79%; precision: 65.57%; recall: 27.69%; FB1: 38.94

Table 5
Results of NER using Ritter Train modified on Indian Tweets Test Data

Named Entity	Precision	Recall	FB1	Number of entities labeled
Geo-Loc	67.94%	49.17%	57.05	131
Company	71.43%	6.76%	**12.35**	7
Person	60.81%	18.22%	**28.04**	74

F. Combined Ritter Train Modified and Derczynski BTC train on Indian tweets test

Processed 5425 tokens with 502 phrases; found: 345 phrases; correct: 213.

Accuracy: 90.97%; precision: 61.74%; recall: 42.43%; FB1: 50.30

Table 6
Results of NER on Indian Tweets Test Data using both BTC and Ritter Train Data

Named Entity	Precision	Recall	FB1	Number of entities labeled
Geo-Loc	68.59%	59.12%	63.50	156
Company	46.15%	8.11%	**13.79**	13
Person	56.82%	40.49%	**47.28**	176

8. Analysis

A. Broad Twitter Corpus

As shown in table 2 and table 3, tests using the BTC training data on the BTC test data and development data have the model outperforming the Ritter original and modified data sets using the Ritter algorithms. I believe the main reason for this is the differing approaches each uses for data selection. Ritter et al. 2011's dataset is anachronistic to some extent since they collected their data in one day, which also means that the data is constrained since it only includes information from those who were active at that period of time (Derczynski et al. 2016). Also, unlike the annotations of the Ritter dataset, which did not incorporate essential entity classification for user mentions and hashtags, the annotations of BTC separated preceding symbols such as # or @ into separate and individual tokens to then be classified as well.

Year	1996	2009	2010	2011	2012	2013	2014
Our corpus	0	3	5	127	2414	275	6022
Ritter (2011)	0	0	6902	0	0	0	0
CoNLL'03	1358	0	0	0	0	0	0

Fig. 12. Describing Spread of data collection by Derczynski et al. 2016

Although holistically testing on BTC shows improvement from the current results, there are still instances in which the performance metrics of an entity fared worse than on the Ritter datasets. Specifically, in the Derczynski development data set and test set, the recall and FB1 scores for the named entity of organization were less than half that of the recall and FB1 score on the Ritter dataset. This is probably because there was not a specific tag for organization in the Ritter algorithms like there was in the BTC, and the closest thing to it was the company tag. We had to map organizations to company but because of this, the model would run on the BTC data and try to find entities that it could classify as a company. And as it was annotated for organization there would have been many instances in which it missed the classification. Another factor could be the lack of information on organizations in the entities dictionaries provided by Ritter. This would make it so that the model does not have sufficient capability to delineate an entity as an organization.

The favorable results of the BTC data are not isolated in our experiments. In tests conducted by Roth et al. 2017 where their NER and POS tagging algorithms were not based off tweets consideration, the utilization of the BTC data for testing and CoNLL 2003 data for training still shows that despite having a cross-domain experiment, their model could still function better on out-of-domain data.

Evaluating the level of mention detection, the BTC allowed their team to see an 8-point increase in their F1 score in their third experiment and a 3.5 point F1 increase in their fifth experiment.

B. Indian Tweets

As shown in table 3, the performance of the Derczynski BTC training data on the Indian Tweets test data dropped significantly from the in-domain training and testing that was conducted using the BTC. We can clearly see that the precision, recall and FB1 scores all were fractions of their counterparts on the in-domain data of the BTC for the named entities of person and company but that each of the metrics were much larger for the named entity of geo-location in the Indian tweets test. We believe that the reason behind this variation could be because of the extensiveness of the geo-locations listed under the locations dictionary embedded in Ritter's algorithms. If we compare Ritter's training data test on the Indian tweets as illustrated in table 5, the named entity of geo-location had higher performance metrics than the other two entities and was also higher than the other instances of geo-location, as can be seen in table 1, 2, 3 even with the Derczynski training set that was stratified for time, location and types of users. Because the annotations in the Indian tweets for geo-location are limited to only country, states, cities and large villages unlike the other test and development data that include names of mountain ranges, lesser-known abbreviations of locations and institutes, the recognition for geo-location was much more distinguishable as the named entities of geo-location present in the locations dictionary of the Ritter model matched the Indian tweets data more.

Comparing the performance of Derczynski's training set on the Indian tweets with Ritter's training set performance on Indian tweets, we can also see that the former had better FB1 and recall scores than the latter. This is notable because it exemplifies that despite the Indian tweets being out of domain data, because of the stratification over social, temporal and spatial, the Derczynski BTC dataset still performed at appreciable levels and improved upon the results of the Ritter train data. It can also be observed through figure 18 that the effect of combining both the Ritter train data, modified to match the 3 entities specified in the BTC, with the BTC train data to test on the Indian tweets had significantly increased the accuracy and FB1 scores than when they were used separately. This proves the existing discrepancies when each corpus was used for training separate from the other.

Using Ziph's law, we can analyze the statistical distribution

of words in a corpus in which the frequencies of words are inversely proportional to their ranks. For example: some very high frequency words accounting for most of the tokens in a certain piece of text can be "the, of, I" etc. or can be very low frequency words, such as "Dalit" or "barbarism," as in our baseline Indian tweets dataset (Piantadosi, NCBI). In each case, with every subsequent frequency, the number of instances of the words can be seen as fractions of the previous word. Following the general formula $f(r) \propto \frac{1}{r^a}$, we can see in figure 1 and figure 2 (below) the comparison between the named entity mentioned in the newswire dataset, which is classically used for non-tweet based and more formal NER procedure and the BTC dataset.

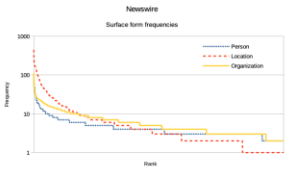

Figure 1: Frequency-Rank curve for entities in CoNLL'03 data.

Figure 2: Frequency-Rank curve for entities in the Broad Twitter Corpus.

Retrieved from Derczynski et al. 2016

9. Conclusions and Future Work

Through manual analysis on the performance of the Ritter algorithms using the BTC train data on BTC test and Indian test data, we can conclude that the BTC is a more reliable corpus than the Ritter one in accounting for variances in the time period, types of users etc. This was clearly shown in the implementation on the Indian tweets baseline data as we observed that the BTC training dataset outperformed the Ritter train dataset and that combination of both datasets improved the results from either individual case.

In the future, we plan on integrating tweet normalization technology with Ritter algorithms to convert slang and abbreviations to their "unambiguous canonical forms" (Liu et al. 2012). Specifically, implementing a system like Han and Baldwin, 2011, where there is a preprocessing step of lexical normalization of tweets, can show more robust performance on the Derczynski results. Liu et al. 2011 corroborates that for every named entity in tweets, there is an average of 3.3 variations in a study conducted over 5 days and on over 12k randomly sampled tweets. This would also be especially useful in the case of NER on Indian tweets as it would allow for the model to be flexible in understanding the variations of the English language through learning the context in the tweets themselves and responding to any new data on a case by case basis. The current Ritter algorithms addresses the problems of OOV words and lexical variations by making use of Turian et al.'s clustering method on words that are distributed similarly to capture the different variations of an entity. Still, this does not go so far as to what normalization would be able to accomplish. For example, through named entity normalization (NEN), if there was the case of a tweet such as "... she knew Burger King when he was a Prince!..." and "···I'm craving all sorts of food: mcdonalds, burger king, pizza, chinese…," the NEN system can systematically deduce that 'burger king' cannot be mapped to Burger King. Because of this, the NER model would then be able to construct two different labels to exemplify that fact. Khalid et al. (2008) even concludes that a simple NEN method can lead to better retrieval performance.

Possible avenues of further improvement to the capitalization problem could be training a mirco-blog specific 'caser' that would check for variations in lowercase and uppercase forms in an entity. We also hope to include more people in the annotations for the Indian tweets and create metrics such as IAA scores to better understand the validity of the data and where we are possibly going wrong in our annotations.

References

[1] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In Proceedings of the 2011 Confer- ence on Empirical Methods in Natural Language Processing, pages 1524–1534, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

[2] Andrew Mccallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In HLT-NAACL, pages 188–191.

[3] Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In The 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, USA.

[4] Daniel M. Bikel, Richard Schwartz, Ralph M. Weischedel. 1999. An Algorithm that Learns What's in a Name. In Machine Learning 34, pages 211–231.

[5] George R. Krupka and Kevin Hausman. 1998. Isoquest: Description of the net owl extractor system as used in muc-7. In MUC-7.

[6] Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In EMNLP, pages 141–150.

[7] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10.

[8] Kazuhiro Yoshida and Jun'ichi Tsujii. 2007. Reranking for biomedical named-entity recognition. In BioNLP, pages 209–216.

[9] Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphae¨l Troncy, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. /Information Processing and Management, 51:32–49.

[10] Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad twitter corpus: A diverse named entity recognition resource. In /Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers/, pages 1169– 1179, Osaka, Japan. The COLING 2016 Organizing Committee.

[11] Liu, X., Zhou, M., Wei, F., Fu, Z., & Zhou, X. (2012). Joint inference of named entity recognition and normalisation for tweets. In Proceedings of the association for computational linguistics (ACL'12) (pp. 526–535).

[12] Mahboob Khalid, Valentin Jijkoun, and Maarten de Ri- jke. 2008. The impact of named entity normalization on information retrieval for question answering. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen White, editors, Advances in In-formation Retrieval, volume 4956 of Lecture Notes in Computer Science, pages 705–710. Springer Berlin / Heidelberg.

[13] Martin Jansche and Steven P. Abney. 2002. Informa- tion extraction from voicemail transcripts. In EMNLP, pages 320–327.

[14] Mayhew, S., Tsygankova, T., & Roth, D. (2019). Ner and pos when nothing is capitalized. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). doi: 10.15653/v1/d19-1650

[15] Mohammad M Masud, Qing Chen, Latifur Khan, Charu Aggarwal, Jing Gao, Jiawei Han, and Bhavani Thurais- ingham. 2010. Addressing concept-evolution in concept-drifting data streams. In Data Mining (ICDM), 2010 IEEE 10th International Conference on, pages 929–934. IEEE.

[16] Piantadosi, S. (2014, October). Zipf's word frequency law in natural language: A critical review and future directions. Retrieved July 12, 2020, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4176592/

[17] Shubhanshu Mishra, Jana Diesner. 2016. Semi-supervised Named Entity Recognition in noisy-text. In Proceedings of the 2nd Workshop on Noisy User-generated Text, pages 203-212.

[18] Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in twitter data with crowd- sourcing. In CSLDAMT, pages 80–88.

[19] Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In ACL.