UE and Nikon at the NTCIR-13 MedWeb Task

Nga Tran Anh Hang
University of Evora, Portugal
m38310@alunos.uevora.pt

Hiroko Kobayashi
Nikon Corp., Japan
Hiroko.Kobayashi@nikon.com

Yu Sawai
Nikon Systems Inc., Japan
Yuu.Sawai@nikon.com

Paulo Quaresma
University of Evora, Portugal
pq@di.uevora.pt

ABSTRACT

The NTCIR-13 MedWeb (Medical Natural Language Processing for Web Document) Task is a continued task series in NTCIR 10-11-12 in more specific symptoms include influenza, diarrhea/stomachache, hay fever, cough/sore throat, headache, fever, runny nose, and cold written. It is to detect the *"signs"* of disease or symptoms from the pseudo-Twitter messages, and can be formalized as multi-label classification problem. We address this task from three different approaches, namely rules, feature engineering, and distributed representations for Japanese and English tweets correspondingly. Among our approaches, the feature-engineering based approach achieved the highest exact-match (80.4%) and F1 score (86.5%). We figured out that each approach has its own strength and shortcomings through errors analysis.

Team Name

UE

Subtasks

MedWeb, twitter task (English, Japanese)

Keywords

rule-based, random-forest, word2vec

1. INTRODUCTION

In the past few decades, there are several applications developed in Machine Learning (ML) and Natural Language Processing (NLP) fields. Both ML as well as NLP keep great promises for making friendly computer interfaces and reducing distance between human-beings and machines. One of those outstanding practical applications is in medical aspect. Twitter[1] is a popular micro-blogging service and has attracted great attention of researchers because twitter can be a valuable personal information resource. However, a tweet is limited to 140 characters, it is indeterminate and challenging for traditional NLP tools to analyze. Previous studies predicted influenza epidemics by using Twitter[3][1][4][5]. NTCIR-13 MedWeb Task[10] can be paraphrased as the task for "predicting the outbreak of epidemics by multi-label classification on the social-media data set". The task is challenging because it has an aspect of the language used on the web as well as that it is a realistic application of current NLP techniques. In this paper, we compare three different approaches, namely *Rule based*, *Feature engineering based*, and *Distributed representation based*.

At first, we introduce rule based method for this task. This is a primitive approach in natural language processing. We can also think of typical machine learning approach for this task. Using explicit features makes the model informative and thus it is beneficial for industrial applications where the model often needs to be *interpretable* even sacrificing performance a bit. Additionally, in such applications, there are not always plenty of tagged or untagged dataset available, especially for this particular domain: *social-media texts with clues of symptoms*. With these in mind, the model is desired to be able to handle a variety of dataset in terms of both its size and word usage. We introduce a multi-label classification model with simple yet effective features later in the next chapter.

As for more applicable for English language tweets, we applied distributed representations. It is common sense that any natural language corpus a majority of the vocabulary word types will either be absent or occur in low frequency. Therefore, we proposed Skip-gram neural network for Word2Vec model and sub-sampling of frequent words in order to extract valuable words.

2. OUR APPROACH

We applied three different approaches for the MedWeb task as follows: rule based (Ja), feature engineering based (Ja and En) and distributed representation based (En). Distributed representation based model is not applied for Japanese subtask because of the size of training data and the result from preliminary experiment[2].

2.1 Rule-based Method (UE-ja-2)

We analyzed dry-run dataset and 23% of whole tweets had all Negative(N) labels. From the result, we proposed simple dictionaries and rules method. The system overview is illustrated in Figure 1. First, we make negative dictionary and when the tweets include negative words, the tweets code all N labels. Negative dictionary is noun data set which is made by calculating frequency of appearance from coding N labels tweets. For extracting nouns, we use

[1] `https://twitter.com/`

[2] We conducted an preliminary experiments, where we use the classifier with the doc2vec styled dense vector which is derived from the model trained on Japanese-Wikipedia corpus. The performance is not as good as other methods. We think it is due to the difference of domain and document vector construction.

MeCab[3] which is Japanese part-of-speech and morphological analyzer, and MeCab-ipadic-NEologd[4] which is language resource for MeCab. Table 1 shows examples taken from negative dictionary.

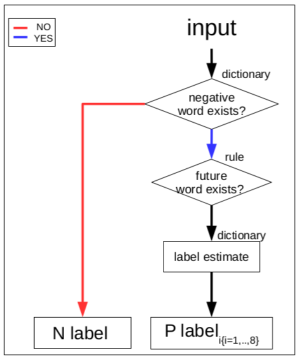

Figure 1: An overview of rule based system.

dictionary name	example
Negative	鳥インフルエンザ、おたふく風邪
Influenza	インフル、インフルエンザ
Diarrhea	下痢
Hayfever	花粉症
Cough	咳、せき、痰
Headache	頭痛
Fever	微熱、高熱、熱
Runnynose	鼻風邪、鼻水、鼻づまり
Cold	風邪、鼻風邪

Table 1: Dictionary examples.

Then, we make future phrase rule which is when the tweet includes future phrase except "(明日) tomorrow" word, we code all N labels to the tweet. We use Japanese parser, KNP[5] for extracting future phrase. Finally, we predict each Positive(P) label$_i\{i = 1, \ldots, 8\}$ by using dictionaries for each symptom. The dictionaries were made similar to negative dictionary. Table 1 is a part of dictionaries.

2.2 Feature Engineering (UE-ja-1, UE-ja-3, UE-en-1)

While the rule based approach performs reasonably good in terms of recall, generalization performance is surely needed for this task, because the variety of words describing symptoms in language on the social-web can be high. Furthermore, for the industrial applications, the model needs to be automated yet human-interpretable and "debuggable". Therefore, we also propose somewhat traditional machine-learning and feature-engineering based approach. We treat this task as a document classification task for short text messages.

2.2.1 *Pre-processing*

Based on the results gained by a preliminary experiment on the dry-run dataset, we use following pre-processing and heuristics:

1. **Normalization**
   This includes common normalization for Japanese sentences before feeding to morphological analyzer, such as normalization of punctuations, symbols and half-width characters. We use similar procedure shown in MeCab-NEologd's pre-processing rules[6].

2. **Lemmatization**
   For Ja. subtask, words are converted into their dictionary form using the meta-data in the dictionary as a part of morphological analysis. For En. subtask, we simply obtain dictionary form using spaCy[7] toolkit, and assign special token for pronouns to reduce sparsity.

3. **Filtering and Extra-dictionary**
   For making the morphological analyzer and the downstream analyzer to work properly, we also filter out the morphemes such as emoticons, delimiters, and symbols.
   Furthermore, we manually add the words which are not chunked properly to the custom-dictionary, and those are selected from the dry-run dataset and heuristics. Note that we only add nouns since estimation of the costs for verbs needs decent amount of tagged corpus.

2.2.2 *Features*

We firstly conducted a preliminary experiment and found that most of "easy-instances", tweets which contain the symptom name itself, can be detected by surface n-gram features as well as the rule-based model mentioned earlier. There are, however, also some cases which can not be handled by short distance relationship. To be precise, those cases are related to long distance relationship between a subject and a verb, named entities, and anaphora resolution.

Therefore, We also employ predicate-argument-structure (PAS) derived features, with things discussed above in mind. After performing trial and errors on the dry-run dataset, our feature set finally consists of the features following:

- Word surface n-grams ($n = 1, 2$)[8]
- Predicate-Argument-Structures
  - Surface of predicate and arguments ("ga", "wo", "ni")
  - Named entity (NE) tag of arguments
  - Pair of surface of predicate and arguments (e.g. "ga=私+pred=引いた")
  - Pair of NE tag of predicate and arguments
  - Triplets of surface of "ga", "wo", "ni", and predicate

[3] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, http://taku910.github.io/mecab

[4] https://github.com/neologd/mecab-ipadic-neologd

[5] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP

[6] https://github.com/neologd/mecab-ipadic-neologd/wiki/Regexp

[7] https://spacy.io/

[8] We tested with simple n-gram features with $n = 1$, which is identical to the Bag of Words model, to $n = 3$, and found that $n = 3$ setting is too sparse for this dataset size.

We use KNP with JUMAN morphological analyzer[9] as the upstream tagger to extract PAS features, while n-gram features are based on the outputs from MeCab with NEologd dictionary.

Although the characteristics of the pseudo-tweet dataset are supposed to be similar to those of the real-world tweets, such as presence of informal expressions, slangs, or spelling inconsistency, most of them are parsed correctly. It is partially because the domain of the dataset is limited, and those are eased by pre-processing mentioned above.

Note that the PAS derived features are used only for Ja. subtask.

2.2.3 Multi-label Classification Model

As the classification algorithm we choose Random Forests[2, 6], which can inherently perform “multi-label” classification.

The Random Forests is an ensemble of decision trees where the features and training examples are chosen randomly. Since it uses bagging strategy, it can reduce the risk of overfitting. Note that we also utilize the model’s capability for feature selection based on the feature importance, and it is particularly useful for debugging and analysis during the dry-run phase.

The hyper-parameters of the model are determined by randomized search with 5-fold cross-validation on the training set, and then used for the formal run experiment with re-estimation of parameters on the whole training set[10].

We use an off-the-shelf implementation, namely Scikit-learn[9] package, for this task.

2.2.4 Voting-ensemble with Rule-based Model (UE-ja-3)

To utilize rule-based model’s strength, we also use simple ensemble approach. We evaluated the dry-run performance of both the rule-based model and feature-engineering based model, and found that the rule-based model performs several percents better in terms of both recall and precision for `Flu` tag. It also achieves better precision for `Cough` tag.

Our meta model used for `UE-ja-3` takes the outputs of both models. It prefers rule-based model’s output for `Flu` tag and positives detected by rule-based model for `Cough` tag, otherwise it just outputs random-forests’ predictions.

2.3 Distributed Representations (UE-en-2, UE-en-3)

As for the rule-based method performs pretty well in term of recall (Ja-UE-2 recall result is 98%) and feature-engineering based method focuses on generalization performance (Ja-UE-3 precision result is 82% and recall result is 91%), in this section, we present you the Skip-gram neural network for word2vec[11] method which is more applicable for English language tweets.

2.3.1 The Skip-gram Neural Network for Word2Vec Model (UE-en-2)

The objective is to find word presentation that are useful for predict the meaning of surrounding words in a tweet. In this work, we used this model as the output of *UE-en-2*. As shown in the Figure 2 belows, the Skip-gram model by Mikolov et al.,2013 [8] highly focuses on learning high-quality vector representation.

We treat the tweets as individual tokens during the training. We used Skip-gram neural network for word2vec model to figure out the word representations for each tweet. During the training processes, we interestingly found that simple vector addition can often produce meaningful results. For instance, vec(“hay fever”) + vec(“runnynose”) is close to vec(“allergies”), and vec(“stuffy”) + vec(“nose”) is close to vec(“runnynose”). This compositionality implies that by using basic mathematical operations can obtain high degree on the word vector representations.

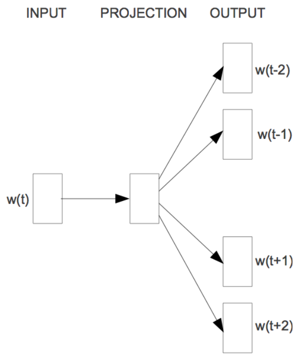

Figure 2: The Skip-gram neural network architecture for Word2Vec.

2.3.2 Sub-sampling of Frequent Words(UE-en-3)

Generally, the most frequent words can easily occur a lot of time in tweets and usually they provide less information value compared with rare words. To counter the imbalance between the rare and frequent words, we used the sub-sampling approach. To notice that by applying sub-sampling approach increases the quality of their resulting word vectors. We applied this method as for the result *UE-en-3*.

The methodology is pretty simple. For each word we encounter in our training text, there is a chance that we will effectively delete it from the text. The probability that we cut the word is related to the word’ s frequency. This process is called subsampling by Mikolov et al., 2013[7].

$$P(W_i) = 1 - \sqrt{\frac{t}{f(W_i)}} \tag{1}$$

where f(W_i)is the frequency of word, W_i and t is the chosen threshold. Take this tweet *I went on a trip and got the flu as a souvenir* as an example, it benefits much less from observing the frequent co-occurrences of “flu” and “the” , as nearly every word co-occurs frequently within a sentence with “the” . In another word, we can process training with the new tweet *I went on a trip and got flu as a souvenir.* which would approach the same result as the given tweet. To conclude, we found this method to work pretty well in practice. It not only accelerates learning but also signifi-

[9] `http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN`

[10] Following hyper-parameters are optimized by the random-search: depth of trees, number of trees, number of features used, and node impurity measure (gini index or cross entropy).

[11] `http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/`

cantly improves the accuracy of the learned vectors of the rare words.

To conclude, as shown in the result of *UE-en-2* and *UE-en-3*, even with different preprocessing approaches, the results show pretty much close to each other which we will discuss more details in results and discussions section.

3. RESULTS AND DISCUSSION

Tables 2 and 3 summarize our formal-run results for each submission. We also conducted error-analysis. Figure 3 shows the counts of each error type.

Rule based approach achieved high recall score but this method got many False Positive errors. Because this method is combination of simple some rules, cases of unrelated to having a disease, inanimate object and question are easy to become errors. The listings below are mis-classified examples of UE-ja-2's result.

- *Unrelated to Having Disease*
  - そろそろインフルの季節だね。
    `Influenza/FN, Fever/FN`
    En: It's almost the flu season.
- *Non-human (inanimate objects)*
  - スマホケースが熱で溶けた・・・
    `Fever/FN`
    En: My phone case melted from the heat...
- *Question*
  - インフルって握手しただけでうつるのかな？
    `Fever/FN`
    En: I wonder if you can catch the flu just from shaking hands?
  - 昨晩から熱が全然下がらない。まさかインフル?
    `Influenza/FN`
    En: My temperature hasn't gone down at all since last night. Don't tell me it's the flu?

Feature engineering based approach outperformed the others in terms of "Exact match" score as shown in Tables 2 and 3. Combined with the rule-based model (UE-ja-3), the recall is slightly improved, whereas the number of false positives got slightly worse because of the rule-based model's characteristics mentioned above. Most of the mis-classifications are found for the tweets with metaphors, discourse structure, and subject-less clauses. The listings below are the excerpts taken from the mis-classified examples of UE-ja-1's result:

- *Metaphors, similes, jokes and slangs on the web*
  - 誰が巫女に**熱あげている**って？
    `Fever/FP`
    (En: Someone has the hots for the miko?)
  - **まるで花粉症のときのように**鼻水が止まらない。
    `Hayfever/FP, Runnynose/FP`
    (En: My nose won't stop running, it's like when I have allergies.)
  - なんか電車が遅延してて**鼻水出た**
    `Runnynose/FP`
    (En: The train is running late and my nose started running)
- *Time-series, tense, and negation (discourse level)*
  - 最近微熱が続くからまさかと思ったら花粉症だった。
    `Hayfever/FN, Fever/FN, Runnynose/FN`
    (En: I've had a slight fever lately and it turns out to be allergies, which I wasn't expecting.)
  - インフルかと思って病院行ったけど、検査したら違ったよ。
    `Influenza/FP, Fever/FP`
    (En: I thought I had the flu so I went to the doctor, but I got tested and I was wrong.)
- *Subject-less clauses (discourse level)*
  - 熱って言ったら弟が心配してくれた。
    `Fever/FN`
    (En: My little brother was worried about me when **I** said I had a fever.)

For metaphors-like cases, even though we used simple preprocessing to replace those metaphorical expressions with equivalent expressions, it did not cover every possible usage on the formal-run dataset. We think that using explicit features for these expressions together with preprocessing can reduce the false positive rate. Mis-classified examples which need to be handled with discourse structures are more fundamental for this task. Sentence level predicate-argument structures features which we used in this task is not sufficient for capturing discourse level structures. In particular, zero-anaphora sentences, which are appeared mostly in Ja-dataset, are crucial cases since we found that those are sometimes easier to solve for En-dataset where most subjects are filled.

The result of skip-gram and sub-sampling is pretty much similar because the test data is pretty "clean". Therefore, when we processed preprocessing methodology for word2vec, they would not affect so much on final results. As shown in Figure 3, most errors were in fever and runny nose cases. The false negative cases cause by misdiagnoses, for example *I have the flu, so I can't go out for a while. (flu positive and fever positive)*, in this case Skip-gram neural network for word2vec could not detect the *fever positive.* This implies that if we do not build a good ontology for diseases, the system will not be able to determine given symptoms. The non-human cases caused a lot of errors too as mentioned above for the Japanese test data.

4. CONCLUSION

As for NTCIR 13 MedWeb task, our team used three approaches for both Japanese and English corpora. While rule based system achieved high recall score whereas it suffered from the number of false positives. Feature engineering based system performed the best in terms of exact-match score. However, it could hardly detect the case where discourse level knowledge is needed. Distributed representation based approach achieved better precision compared to traditional features.

In the future work, we expect that the usage of semantic and discourse level features might improve the performance.

On another the hand, for making clear each diseases or symptoms relationship, we think about utilizing ontologies. Using ontologies can help reducing false negative rate. The implication of this work is significant and far reaching. However, having more data would further raise the performance. Our work also marks the high possibility to build a multilingualism tweet diagnosis system efficiently in the near future.

Acknowledgements

We would like to express my sincere gratitude to the Laboratory for Informatics, Systems and Parallelis (LISP) for the support of the project. The project is partially funded by

System ID	Exact match	F1		Precision		Recall		Hamming loss
		micro	macro	micro	macro	micro	macro	
UE-ja-1	0.8047	0.8652	0.8550	0.8305	0.8190	0.9028	0.9024	0.0328
UE-ja-2	0.7063	0.8153	0.8029	0.6963	0.7018	0.9832	0.9838	0.0520
UE-ja-3	0.8000	0.8658	0.8551	0.8233	0.8124	0.9129	0.9109	0.0330

Table 2: Results on Ja. subtask.

System ID	Exact match	F1		Precision		Recall		Hamming loss
		micro	macro	micro	macro	micro	macro	
UE-en-1	0.7891	0.8581	0.8483	0.8455	0.8307	0.8710	0.8762	0.0336
UE-en-2	0.7453	0.8214	0.8091	0.8606	0.8375	0.7856	0.7996	0.0398
UE-en-3	0.7391	0.8202	0.8153	0.8703	0.8510	0.7755	0.7953	0.0396

Table 3: Results on En. subtask.

FCT through the grant *UID/CEC/4668/2016 - Laboratory for Informatics, Systems and Parallelis*.

5. REFERENCES

[1] E. Aramaki, S. Maskawa, and M. Morita. Twitter catches the flu:detecting influenza epidemics using twitter. pages 1568–1576, 2011.

[2] L. Breiman. Random Forests. *Machine learning*, 45(1):1–33, 2001.

[3] A. Culotta. Detecting influenza outbreaks by analyzing twitter messages. 2010.

[4] H. Iso, S. Wakamiya, and E. Aramaki. Forecasting word model: Twitter-based influenza surveillance and prediction.

[5] A. Lamb, M. J.Paul, and M. Dredze. Separating fact from fear:tracking flu infections on twitter. 2013.

[6] G. Louppe. Understanding Random Forests: From Theory to Practice. (July):223, 2014.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, `https://arxiv.org/abs/1301.3781`. 2013.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality, `https://arxiv.org/pdf/1310.4546.pdf`. 2013.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2012.

[10] S. Wakamiya, M. Morita, Y. Kano, T. Ohkuma, and E. Aramaki. Overview of the NTCIR-13: MedWeb Task. In *Proceeding of the NTCIR-13 Conference*, 2017.

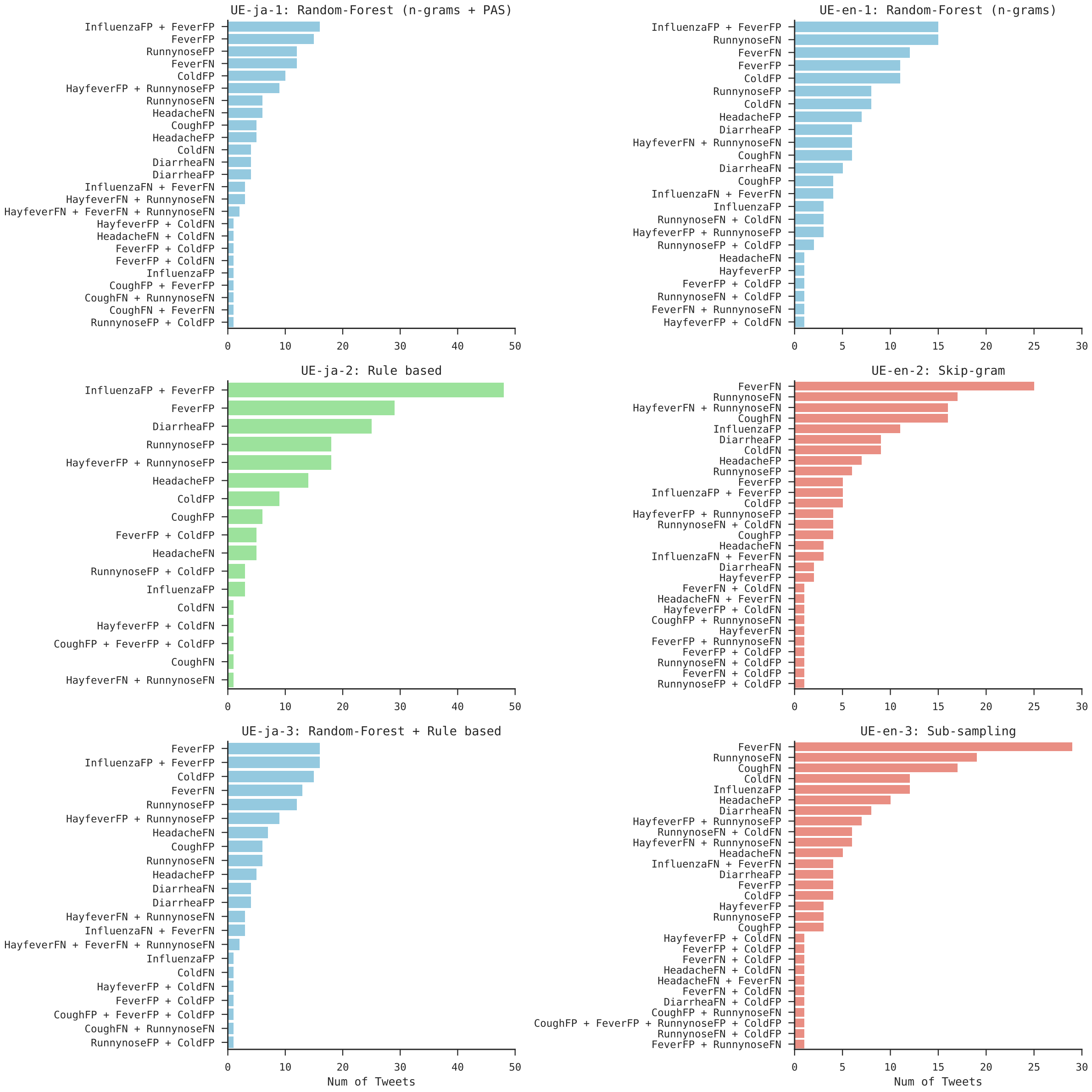

Figure 3: Error type statistics for each approach. FP and FN correspond to false-positive and false-negative respectively.