Extraction of Medication Names from Tweets CLaC at BioCreative VII Track 3

Parsa Bagherzadeh and Sabine Bergler
CLaC Labs, Concordia University, Montreal, Canada
{p_bagher, bergler}@ cse.concordia.ca

***Abstract*—We present a modular model that leverages knowledge sources including specialized gazetteer lists, morphological information, and contextualized language models for the task of medication name extraction from tweets. The proposed system demonstrates high recall (.81) and low precision (.68). We explain the low precision score and show a simple workaround in post-competition evaluation.**

***Keywords*—Multi-input RIM, modular model, drug names, knowledge sources.**

I. Introduction

BioCreative VII Track 3 concerns the extraction of text spans that mention a medication or dietary supplement in tweets. The training dataset comprises around 89,000 tweets, and is extremely imbalanced (only 213 tweets include any drug mention), which makes the task very challenging. When positive samples are rare it is more likely that deep learning models may not perform well due to the terms of the test set not being foreshadowed sufficiently in the training data, i.e. coverage becomes a bottleneck.

To address this problem, external knowledge sources such as gazetteer lists of drug names can be used. Weissenbacher et al. (6) proposed a two stage system where a BERT-based model makes predictions which are filtered using a drug lexicon.

Here, we integrate outside lexical resources with deep learning in a unified model, enabling both to inform on one another and to act in synergy. For extraction of medication names from tweets we leverage different knowledge sources such as gazetteer lists, word embeddings, morphological information, etc. using the multi-input RIM architecture (1). Each knowledge source provides input to an independent module which occasionally interacts with other modules.

II. Proposed system

A. Multi-input RIM

We use the multi-input RIM (mi-RIM) architecture (1), encompassing M independent, yet interacting recurrent modules. At each time step, the number of active modules is controlled by a parameter k, which can force competition among modules. As argued by (3), this competition can lead to specialization on subproblems.

a) Input selection: Each module R_m augments the token input x_t^m to $X_t^m = x_t^m \oplus \mathbf{0}$, where $\mathbf{0}$ is an all-zero vector and $\oplus$ denotes row-level concatenation. Then, using an attention mechanism, unit R_m selects input:

$$A_t^m = softmax\left(\frac{h_{t-1}^m W_m^{query}(K_m)^T}{\sqrt{d_h}}\right)V_m \tag{1}$$

where $h_{t-1}^m W_m^{query}$ is the *query*, $K_m = X_t^m W_m^{key}$ is the *key*, and $V_m = X_t W_m^{val}$ is the *value* in the attention mechanism. If the input x_t is relevant to the task, the attention mechanism in Equation 1 assigns more weight to it (selects it). The $softmax$ values of Equation 1 determine a subset S_t of the k highest ranked units. Among M units, those with the least attention on the null input are the active units. The selected input A_t^m the is used to calculate a temporary hidden state $\tilde{h}_t^m$ for the active units:

$$\tilde{h}_t^m = R_m(h_{t-1}^m, A_t^m) \quad m \in S_t \tag{2}$$

where $R_m(h_{t-1}^m, A_t^m)$ denotes one iteration of updating the recurrent unit R_m based on the previous state h_{t-1}^m and the current input A_t^m. The hidden states of the inactive units R_m ($m \notin S_t$) remain unchanged ($h_t^m = h_{t-1}^m \; m \notin S_t$).

b) Interaction: To obtain the actual hidden states h_t^m, the active units communicate using an attention mechanism:

$$h_t^m = softmax\left(\frac{Q_{t,m}(K_{t,:})^T}{\sqrt{d_h}}\right)V_{t,:} + \tilde{h}_t^m \quad m \in S_t \tag{3}$$

where

$$Q_{t,m} = \tilde{h}_t^m \tilde{W}_m^{query}$$

$$K_{t,:} = [\tilde{h}_t^1 \tilde{W}_1^{key} \oplus \ldots \oplus \tilde{h}_t^M \tilde{W}_M^{key}]$$

$$V_{t,:} = [\tilde{h}_t^1 \tilde{W}_1^{val} \oplus \ldots \oplus \tilde{h}_t^M \tilde{W}_M^{val}]$$

B. Knowledge sources

a) Word embeddings: Word embeddings provide a meaning representation based on co-occurrence statistics. To embed tokens, we use the representations provided by the last layer of ClinicalBERT (7).

b) Morphology: Drug names often have a specific morphology, favoring certain prefixes, suffixes, etc. The suffix *-statin*, for instance, is observed in drug names of this type, including *torvastatin, lovastatin,* and *pravastatin*. Following (9) we use a character level Convolutional Neural Network to obtain a morphological representation for each token. We use multiple convolution filters $\mathcal{F}_l$ with the range of lengths $l \in \{2, 3, 4, 5\}$. The different filter sizes provide representations that capture character bi-grams, tri-grams, 4-grams and 5-grams, simultaneously. The resulting character-based representations are in a 100-dimensional space.

c) POS: Part-of-speech tags are the most widely used linguistic features and are available from many standard NLP environments. POS tags provide useful information such as types of pronouns and tense for verbs, important clues for sequence labeling. Following (2), we pre-train POS tags using Word2Vec (8) to initialize an embedding layer. We apply ANNIE tweet POS tagger on the training sets of SMM4H 2018 (all tasks) (10), SMM4H 2019 (Tasks 1 and 4) (11), SMM4H 2020 (Task 5) (12), SMM4H 2021 (13) (all tasks except Task 7), and BioCreative VII Track 3, and pre-train the POS embeddings.

d) Gazetteer lists: To encode gazetteer annotations, we use an embedding layer $E_{gaz} \in \mathbb{R}^{(3+1)\times 20}$. Each row in E_{gaz} embeds one of the following three gazetteer lists. A fourth row encodes lack of gazetteer matches.

Drug: DrugBank (5) includes commercial drug names as well as the scientific names of their active ingredients.

Anatomy: Body part mentions are important evidence for drug mention detection. Drug mentions often contain the body part that hurts and for which a drug is consumed. For instance, *muscle relaxer* in Example 1 is a drug mention. Relevant anatomy terms are extracted from sub-tree A of MeSH (4) into a gazetteer list.

Example 1:

> *Just took my first muscle relaxer to help with my back pain*

Disease: Many drug mentions refer to the disease, which the drug attacks (see Example 2). A gazetteer was compiled from subtree C in MeSH which includes terms for *infections, wounds, injuries, pain, etc.*

Example 2:

The whooping cough injection site has killed my arm. 😔

In addition to external knowledge sources, we automatically extract a black list and a white list during the training of our model.

e) Black- and white-list: A black list of forbidden terms and a white list of acceptable terms are automatically complied during training. The black-list collects terms that occur in false positives, intended to improve the precision of the system. The white-list collects terms from false negatives and intends to improve recall. Several examples from black and white lists are provided in Table I.

TABLE I
SAMPLE TERMS IN BLACK AND WHITE LISTS

BlackList	**WhiteList**
lol, lollol, vodafone, xoxo, perrin, virgo, texans, atrophy, alrilyic, preggo, lebron, pumpkin, ...	narcotics, vitamins, pill, pills, opioids, meds, medication ...

C. Training paradigm

We partition the training set into two sub-sets D_1 and D_2. D_1 is used to train the model during the first epoch. At the end of the first epoch the model demonstrates a high recall and low precision. We use this model to make predictions on D_2. All false positive predictions are added to the black-list and all false-negative predictions are added to white-list. We continue the training on the original training data (D_1 and D_2) for another 3 epochs. Note that we keep the black and white lists fixed during Epochs 2–4.

We implement the proposed system using the PyTorch library and optimize it using Adam optimizer with $lr = .1e{-}5$

III. RESULTS

A. Development phase

1) Numerical results: To evaluate the effectiveness of each module, we perform an ablation study. Table II presents the results on the development set provided by the organizers[1]. We compare to ClinicalBERT as the baseline (first row in Table II).

Weissenbacher et al. (6) show that a lexicon-based approach for drug mention detection results in a high recall and low precision. Table II shows, in contrast, that our *Drug* and *Disease* gazetteers achieve rather balanced precision and recall and that both modules independently improve F1.

The *Morphology* module and the White list, as expected, each improve recall considerably, but lower precision. Nevertheless, the gain in recall ($+.08$), outweighs the loss in precision ($-.02, -.04$) and $F1$ improves by $.02, .03$.

The *Anatomy* module provides only marginal improvements. Further analysis of the development set revealed that there are only 6 drug mentions that include a disease name or the name of a body part. The fact that this module shows improvement on a sample size this small is noteworthy.

The POS embeddings module is precision oriented; these results confirm the observations in (2).

[1]the evaluation is preformed using the script provided by the organizers

TABLE II
PERFORMANCES ON DEVELOPMENT DATA

Modules	k	P	R	F1
ClinBERT (Baseline)	-	.71	.70	.70
ClinBERT, POS	2	.73	.70	.72
ClinBERT, Drug	2	.75	.73	.74
ClinBERT, Disease	2	.71	.73	.72
ClinBERT, Anatomy	2	.69	.72	.71
ClinBERT, Morph	2	.69	.78	.73
ClinBERT, White	2	.67	.78	.72
ClinBERT, Black	2	.85	.63	.74
ClinBERT, Black, White	3	.76	.77	.76
ClinBERT, AllGaz, Black, White	4	.85	.81	.83
ClinBERT, Morph, AllGaz, Black, White	5	.84	.86	.85
ClinBERT, Morph, AllGaz, Black, White, POS	6	.85	.86	.85
ClinBERT, Morph, AllGaz, Black, White, POS	3	.87	.88	.88

2) Error analysis: Here we provide an analysis for some error cases and investigate how each knowledge source affects the predictions:

a) Morphology: The morphology module is devised to capture the drug mentions that are not observed in the training data, are not present in a drug list, or have typographical errors. Example 3 includes a mention of the drug *Vistaril*[2] that has a typo. The Morphology module, however, was able to capture this mention that was not recognized by ClinicalBERT.

Example 3: (true positive)

. . . My doctor Put me On Vastaril to Help settle them down. It helps a bit.

Example 4 provides another example of a true positive prediction (*Hydrocodone bitartrate*) in the presence of the morphology module.

Example 4: (true positive)

Hydrocodone bitartrate & Celebrex 💊💊💊 *I'm so set*

The Morphology module is recall-oriented, thus the module makes false positive predictions, two of which are provided in Examples 5 and 6:

Example 5: (false positive)

@USER #Huaweip10onVodafone would love to be so lucky!

Example 6: (false positive)

@USER have you tried viva vegeria? It's on Nogalitos by the HEB.

Note that we use a hashtag tokenizer and the hashtag *#Huaweip10onVodafone* (Example 5) is segmented into [*#, Huaweip10, on, Vodafone*].

[2]a.k.a *Hydroxyzine*

b) BlackList: To counteract false positives, the BlackList module collects a list of forbidden terms, automatically compiled from false positives in the first epoch. The BlackList module improves precision for instance by compensating for the false positive prediction in Example 5, when the term *Vodafone* occurs already in the BlackList. The BlackList however does not contain *vegeria* (Example 6) and consequently fails to prevent that false positive error.

c) Drug gazetteer: Example 7 shows a drug mention that has not been observed in the training data. The Drug module injects this drug mention for a true positive prediction.

Example 7: (true positive)

fentanyl is where it's at!!!! 👋🏻 *Goodbye pain, Paige feels wasted*

d) Disease gazetteer: A true positive prediction enabled by the Disease gazetteer is *whooping cough injection*:

Example 8: (true positive)

Got my whooping cough vaccine yesterday and now my arm is sore.

B. Evaluation phase

Table III reports our competition results. The recall scores for all three runs are on a par with their corresponding runs on the development set. On the other hand, significant drops in precision scores were unexpected. Adding the Morphology module for Run 2 marginally improves recall over Run 1 and marginally lowers precision. It is interesting to see that this small drop in precision compensated when adding the POS module in Run 3.

All these differences between our three runs are, however small and dwarfed by the unexpected unbalanced nature of our precision and recall scores.

TABLE III
OFFICIAL COMPETITION RESULTS

Run	Modules	k	P	R	F1
1	ClinBERT, AllGaz, Black, White	4	.65	.79	.72
2	ClinBERT, Morph, AllGaz, Black, White	5	.64	.81	.71
3	ClinBERT, Morph, AllGaz, Black, White, POS	3	.68	.81	.73
	Competition mean		.81	.70	.74
	Competition std		–	–	.07

Note that in all three submission runs, the recall-oriented WhiteList module is included (in fact a late addition to our competition runs).

In post-competition experiments, we excluded the WhiteList module. The results in Table IV show balanced precision and recall scores commensurate with our results on development data (and, in fact, above mean $F1$).

The striking difference may be explained by the fact that the test data was manually corrected for some annotation errors just before competition closed, but the training data was not.

This is an important reminder that devices like white lists are very unstable and do not transfer. It is intuitive that the black list did not suffer if we assume that the errors were commission errors. If the corrected errors had been predominantly omission errors, the black list could have backfired.

TABLE IV
POST-COMPETITION RESULTS EXCLUDING WHITELIST

Modules	k	P	R	F1
ClinBERT, AllGaz, Black	3	.78	.78	.78
ClinBERT, Morph, AllGaz, Black,	4	.76	.79	.77
ClinBERT, Morph, AllGaz, Black, POS	3	.77	.80	.78

IV. CONCLUSION

For detection of medication names in tweets we submitted three runs of a modular model that leverages external knowledge sources including specialized gazetteer lists, morphological information, and automatically compiled word lists.

Our systems showed balanced precision and recall during development, supported by white and black lists compiled during the first epoch of training.

The competition runs did not follow this pattern. While recall was substantially above the mean, precision was (more) substantially below the mean.

In post-competition experiments we determined that the WhiteList module was the cause for the drop in precision.

This shows that unlike the other modules that provided generalization for the system, the greedily compiled white and black lists can lead to overfitting. We conclude that such resources have to be further counterbalanced.

REFERENCES

1. Bagherzadeh, P., Bergler, S., (2021) Multi-input Recurrent Independent Mechanisms for leveraging knowledge sources: Case studies on sentiment analysis and health text mining. In Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures.
2. Bagherzadeh, P., Bergler, S., (2021) Leveraging knowledge sources for detecting self-reports of particular health issues on social media. In Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis, pages 38–48,online.
3. Goyal, A., et. al, (2019) Recurrent Independent Mechanisms, arXiv preprint arXiv:1909.10893
4. Lipscomb, C. E., (2000) Medical subject headings (MeSH). Bulletin of the Medical Library Association, 88(3)
5. Wishart, D. S., et al., (2018) Drugbank 5.0: a major update to the drugbank database for 2018. Nucleic acids research, 46(D1):D1074–D1082.
6. Weissenbacher, D., et al., (2021) Addressing Extreme Imbalance for Detecting Medications Mentioned in Twitter User Timelines, in Proceedings of International Conference on Artificial Intelligence in Medicine.
7. Alsentzer, E., et. al., (2019) Publicly available clinical BERT embeddings. In Proceedings of the 2nd Clinical Natural Language Processing Workshop.
8. Mikolov, T., et al., (2013) Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems.
9. Kim, Y., et al., (2016) Character-aware neural language models. In Thirtieth AAAI conference on Artificial Intelligence
10. Weissenbacher, D., et al., (2018) Overview of the third Social Media Mining for Health (SMM4H) Shared Tasks at EMNLP 2018. In Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop and Shared Task, pages 13–16.
11. Weissenbacher, D., et al., (2019) Overview of the fourth Social Media Mining for Health (SMM4H) shared tasks at ACL 2019. In SMM4H 2019.
12. Klein, A. Z., et al., (2020) Overview of the Fifth Social Media Mining for Health applications (SMM4H) shared tasks at COLING 2020. In SMM4H 2020.
13. Klein, A. Z., et al. (2021) Overview of the Sixth Social Media Mining for Health applications (SMM4H) Shared Tasks at NAACL 2021. In Proceedings of the Sixth Social Media Mining for Health Applications (SMM4H) Workshop & Shared Task