International Journal of Medical Informatics 129 (2019) 20–28

Contents lists available at ScienceDirect

International Journal of Medical Informatics

journal homepage: www.elsevier.com/locate/ijmedinf

Categorization of free-text drug orders using character-level recurrent neural networks

Yarden Raiskin[a], Carsten Eickhoff[b], Patrick E. Beeler[c,*]

[a] Dept. of Mathematics, Seminar for Statistics, ETH Zurich, Universitätstrasse 6, 8092, Zurich, Switzerland
[b] Center for Biomedical Informatics, Brown University, 233 Richmond Street, Providence, RI, 02912, United States
[c] Department of Internal Medicine, University Hospital Zurich and University of Zurich, Raemistrasse 100, 8091, Zurich, Switzerland

ARTICLE INFO

Keywords:
Drug ordering
Machine learning
Recurrent neural networks
Character-level models

ABSTRACT

Background and purpose: Manual annotation and categorization of non-standardized text ("free-text") of drug orders entered into electronic health records is a labor-intensive task. However, standardization is required for drug order analyses and has implications for clinical decision support. Machine learning could help to speed up manual labelling efforts. The objective of this study was to analyze the performance of deep machine learning methods to annotate non-standardized text of drug order entries with their therapeutically active ingredients.
Materials and methods: The data consisted of drug orders entered 8/2009-4/2014 into the electronic health records of inpatients at a large tertiary care academic medical center. We manually annotated the most frequent order entry patterns with the active ingredient they contain (*e.g.* "Prograf"⟵"Tacrolimus"). We heuristically included additional orders by means of character sequence comparisons to augment the training dataset. Finally, we trained and employed character-level recurrent deep neural networks to classify non-standardized text of drug order entries according to their active ingredients.
Results: A total of 26,611 distinct order patterns were considered in our study, of which the top 7.6% (2028) had been annotated with one of 558 distinct ingredients, leaving 24,583 unlabeled observations. Character-level recurrent deep neural networks achieved a Mean Reciprocal Rank (MRR) of 98% and outperformed the best representative baseline, a trigram-based Support Vector Machine, by 2 percentage points.
Conclusion: Character-level recurrent deep neural networks can be used to map the active ingredient to non-standardized text of drug order entries, outperforming other representative techniques. While machine learning might help to facilitate categorization tasks, still a considerable amount of manual labelling and reviewing work is required to train such systems.

1. Background

Hospitals have been storing increasingly large amounts of electronic health record (EHR) data, including unstructured information such as non-standardized text, which in the clinical setting is sometimes referred to as "free-text". [1,2], Drug orders in particular hold considerable value for clinical decision support, prognostic modelling, inferring conditions and forecasting the risk for adverse events. [3,4], To accomplish such downstream tasks, a standardized representation of drug order information is required. However, standardizing drug order information is a cumbersome and time-consuming manual effort, *e.g.* mapping the pharmaceutically active ingredient to the misspelled brand name as entered by the provider. Machine learning, natural language processing (NLP) and information retrieval methods have the potential to advance the manual process of making non-standardized text more accessible in clinical reasoning, for both humans and data-driven counterparts [5].

2. Natural language processing for drug order information

Non-standardized text entries are an important part of general, as well as drug-related EHR documentation. However, the advantages of coded information in EHRs and the disadvantages of unstructured information are well-recognized and several studies have been published on NLP applications for non-standardized text. [5] While the tasks addressed varied widely from study to study, numerous NLP tools were developed to extract drug information from clinical notes [6]. Many of these focused on dosage information such as the rule-based approach by Karystianis et al. [7] The successful tool "MedEx" developed and published by Xu et al. used a semantic tagger in combination with a parser

* Corresponding author.
E-mail address: patrick.beeler@usz.ch (P.E. Beeler).

https://doi.org/10.1016/j.ijmedinf.2019.05.020
Received 11 August 2018; Received in revised form 25 March 2019; Accepted 21 May 2019

to extract drug names, administration routes, dose information and other features of drug orders [8].

Further efforts were undertaken to apply NLP techniques to clinical text. [9] In particular, researchers tend to concentrate on two variants of the same underlying task a) classifying an isolated stretch of text into one of several possible drug classes, or, b) recognizing named drug entities in longer consecutive stretches of text such as scholarly articles or electronic health records. Patrick and Li developed a hybrid machine-learned and rule-based system for drug information extraction from EHRs [10]. Doan and Hu addressed the same task via a support vector machine with polynomial kernel and literal as well as syntactic and medical domain features, finding that rich manually crafted feature sets deliver the best performance results [11]. Jiang et al. disambiguated clinical entities in hospital discharge summaries using a range of traditional NLP techniques and bag-of-words representations. [9] Hussain and Qamar relied on tokenization and contextual information extraction derived from part-of-speech tagging and lexical operations for drug name matching followed by a term frequency-inverse document frequency (TF-IDF) ranking [12]. While a wide range of vocabulary based information extraction schemes, *e.g.*, based on resources such as the Unified Medical Language System (UMLS) have been proposed in the past [13–15], more recent comparative studies find machine-learning-based models to perform better for this task [6]. For this reason, our empirical performance comparison considered a recently described support vector machine baseline. In the following section we will discuss a range of more modern neural network-based approaches to medical text processing.

3. Neural networks for medical text

Artificial neural networks are inspired by the highly interconnected neurons of the animal brain and consist of groups (so-called layers) of nodes interconnected via mathematical activation functions. Given sufficiently many nodes and layers, these networks have the ability to approximate highly non-linear target functions and have been shown to excel at a wide range of tasks. An introduction into the topic can be found elsewhere. [16] Recurrent neural networks [17] are able to process input sequences of arbitrary length while maintaining an internal representation of state. This property makes them highly suitable for speech recognition and natural language processing tasks. The Long Short-Term Memory (LSTM) module [18] has been especially effective by maintaining two latent state representations, distinguishing between a latent vector used in making predictions and another one used to encode sequential information in the module. This distinction is meant to allow for the representation of long-term dependencies across the input sequence. Unanue et al. used bi-directional LSTM-CRF models on word tokens for extraction of biomedical entities such as drug mentions [19]. The Gated Recurrent Unit (GRU) [20] is a newer alternative RNN variant whose architecture resembles a simplified LSTM, striking a compromise between representation expressiveness and complexity. Gehrmann et al. presented a comparison of neural networks and more traditional natural language processing techniques for the task of patient phenotyping from clinical narratives, finding that the representation learning capabilities of neural network architectures can often outperform the more static behavior of systems relying on manually designed features [21].

There are numerous examples of word-level RNNs based on part-of-speech tag embeddings corresponding to words or character-level embeddings in the literature. [22,23], A token or word granularity enables the use of pre-trained embeddings, leveraging insights and resources from previous research efforts. However, this benefit comes at a cost. Computation is limited to a specific vocabulary and unable to process previously unseen words during the test or application phase, which has been referred to as an “out-of-vocabulary” issue. Character-level models have been successful at circumventing such effects. [24,25], Lipton et al. applied LSTMs to perform multi-label diagnosis classification, using irregularly sampled multivariate time series of clinical measurements [26]. The success of this model family inspired some of the approaches presented in this article. Gridach proposed bi-directional LSTMs, on character-level embeddings for Biomedical Named Entity Recognition [27]. Hasan et al. presented an attention-based bidirectional LSTM (alongside an encoder-decoder framework) to perform clinical paraphrasing with various applications such as search, summarization, and question answering [28].

To the best of our knowledge, and at the time of writing this article, there has been no prior study using pre-categorized, non-standardized text drug order entries to train RNN-based drug classification schemes. The objective of this study was to analyze the performance of deep machine learning methods to annotate non-standardized text of drug order entries with their therapeutically active ingredients.

4. Materials and methods

4.1. Description of dataset and pre-processing

The data was comprised of 26,611 unique, non-standardized text entries of drug order strings (referred to as “patterns”) collected 8/2009-4/2014 at the University Hospital Zurich, a large tertiary care academic medical center in Zurich, *i.e.* located in the German-speaking part of Switzerland. Drug order entries are almost exclusively German, as entered by the providers.

A typical entry is comprised of the brand name or active ingredient, often with information on the dosage, *e.g.* “Torasemid 10mg”. While some of these patterns, *e.g.* “Diamacron 60″ (misspelled), were only observed once (*i.e.*, only a single order used exactly this string), most of them re-occurred multiple times in the dataset. For instance, the string “Marcoumar 3mg” had been independently entered 17 times. We ordered all patterns descending by their frequency in the dataset and manually categorized the top 7.6% (2028) by annotating them with one of 558 distinct ingredients, leaving 24,583 unlabeled observations. The mean length of the drug order entries was 31.3 characters, an entry could be up to a maximum of 80 characters long, and the most frequent length across all patterns was 14 characters.

We used the Anatomical Therapeutic Chemical Classification System (ATC codes, WHO, Geneva, Switzerland; cf. https://www.whocc.no/) as source of drug ingredients. For instance, the non-standardized and slightly misspelled text “Amlodiipin 5mg” should be classified as a reference to the standardized ATC code C08CA01 for the active ingredient amlodipine, as the code is labeled in the ATC catalogue. For the classification task, we focused only on the non-standardized text of the drug order as entered by the provider on the one hand, and on the other hand, corresponding ATC codes and their labels (ATC catalogue lists codes, and each code is labeled with the active ingredient). One active ingredients may have multiple valid ATC codes, *e.g.* vancomycin is usually administered intravenously (J01XA01), but in rare cases of oral administration of vancomycin for intestinal infections, the ATC code A07AA09 could be used. Nevertheless, we follow [10] in restricting our training dataset to exclusively consist of prescriptions with a single target in order to ensure a feasible task scope as well as accurate performance evaluation.

All drug order patterns were broken up into sequences of characters, making them the atomic tokens on which our method operates. We filtered out infrequent characters and replaced them with a designated “UNKNOWN” token. This step is a common practice in NLP to enable classifiers (neural networks as well as others) to process arbitrary sequences of text even if a particular token has never been encountered during model training. [28] We applied “near-no-filtering”, requiring a token to appear at least twice in the corpus or otherwise be replaced with a designated “UNKNOWN” token.

For the test set, we applied proportionate stratified random sampling, stratifying by ATC codes, with a sampling fraction of 0.25, ensuring that 25% of the examples of each ATC code were included in the

test set. The test set was only comprised of observations from the original labeled data-set (*i.e.* not derived from any of the data augmentation steps described in the following section). In case the share of original labeled observations in the stratum was lower than 25% (this can be the case when most examples stemmed from the augmentation process described below), all original labeled observations of that label were reserved for testing.

4.2. Similarity-based classifier bootstrapping

Neural networks often rely on thousands or tens of thousands of tunable parameters. They have been shown to require large amounts of training data to accurately fit these parameters and thereby fully reach their potential predictive power. [16] However, in our case, labeled data (2028 instances) was scarce, compared to an abundance of unlabeled examples (24,583 instances). We addressed this imbalance by means of a bootstrapping approach that propagated explicit labels to highly similar unlabeled instances before using both explicit as well as inferred labels for model training. To do so, we used a similarity measure, denoted $sim(\cdot,\cdot)$, given by the Jaccard similarity, $Jaccard(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$, where X_i is a bag-of-words representation.

As a concrete example, the similarity between “Marcoumar” and “Marciumar”, a highly related pair of patterns is 0.8 while the similarity between “Marcoumar” and “Recormon” amounts to only 0.31. The similarity-based label-suggestion procedure for a given unlabeled observation $\tilde{X}_j$ and similarity threshold is described by the following steps:

- Compute the similarities $sim(\tilde{X}_j, X_i)$ for all labeled observations X_i.
- Retrieve a set of suggested labels and their corresponding similarities, such that the remaining similarities are above the given similarity threshold. The suggested label set is defined as: $suggested(\tilde{X}_j) = \{(Y_i, sim(\tilde{X}_j, X_i)) | sim(\tilde{X}_j, X_i) \geq threshold\}$
- Aggregate the set of distinct labels among the suggestions. The distinct label set of an input sequence is called **unanimous** if it contains exactly one label. $distinct(\tilde{X}_j) = \{Y_i | (Y_i, \cdot) \in suggested(\tilde{X}_j)\}$

We evaluated similarity threshold values in the range [0.5, 0.85] using 0.05 increments and grouped all patterns in this score range into a discrete similarity bin (*e.g.*, all patterns with similarity scores > 0.5 and $<= 0.55$ are associated to the same bin, patterns with scores > 0.55 and $<= 0.6$ to the next, *etc.*). When choosing a high-similarity threshold, recall (sensitivity) is sacrificed for precision (positive predictive value [PPV]). In order to judge the quality of labels suggested by this procedure, a medical domain expert assessed their correctness on the basis of a sample of 100 suggestions. Within each similarity bin, suggestions were sampled at random. Since we expected the likelihood of error to be greater for low-similarity suggestions, we biased the sampling process to draw linearly more frequently from the low end of the similarity scale. Hence, we drew 19,17,16,14,13,11,10 observations from each bin, respectively.

Fig. 1 plots the observed frequency of correct and incorrect suggestions as functions of the chosen similarity threshold.

We noted that similarity thresholds below 0.7 are highly error prone and should not be considered. It was, however, less clear how the proportions of false suggestions made in the [0.75, 0.8] and [0.7, 0.75] bins compare. They provided an interesting trade-off between label accuracy and breadth of coverage. As a consequence, we evaluated classifiers derived from various similarity thresholds of {0.7, 0.8, 0.9} and eventually also used a data-set without augmentation, which is equivalent to setting the similarity threshold to 1.0.

4.3. Proposed model variants

Recurrent Neural Networks are a neural network family that has been shown to be suitable for processing variable-length input sequences such as text data. [18] These algorithms and their respective performances can vary considerably depending on input data representation, network structure, optimization procedures, or target cost function. This article investigates several network hyper-parameters, RNN cell architectures, data feed directions, input data representations, regularization schemes and optimization protocols. For input data representation, we evaluate both one-hot encoding (one distinct active bit per input sequence) and character-level embeddings [29]. In all settings, we rely on the popular Adam optimizer [30]. The remainder of this section discusses the compared RNN types, data feed directions and regularization techniques.

Another aspect of the RNN architecture that needs to be considered is the manner in which data is parsed by the network. Traditionally, RNNs read their inputs sequentially, in a temporally or otherwise ordered fashion. Bidirectional RNNs, [31] can be applied to a finite sequence, by feeding inputs into an RNN at both the forwards and backwards direction. This procedure has been shown especially effective in sequence-to-sequence translation tasks and was included as an experimental parameter of our study.

To counter over-fitting effects, this study employed L_2 norm regularization (weight decay), dropout, [32] target replication [33] and noisy activation functions [34]. Target replication, first introduced under the label of “companion loss”, makes the task of classifying entire sequences easier by replicating targets at every step. This study adopted a setup inspired by Lipton et al. [26] having only one set of weights that are used both for output prediction and target replication prediction. Target replication is an especially promising choice in settings where input sequences vary as expected for different spellings of drug names.

When applying target replication, an output $\hat{y}^{(t)}$ is generated at every sequence step. The resulting loss is a convex combination of the final loss (at step T) and the average of the losses over all steps, as defined in Equation 1, where $\alpha \in [0,1]$ is a hyper-parameter that determines the relative importance of intermediary targets. At prediction time, only the final step’s output is considered.

$$\alpha \cdot \frac{1}{T} \sum_{t=1}^{T} loss(\hat{y}^{(t)}, y^{(t)}) + (1-\alpha) \cdot loss(\hat{y}^{(T)}, y^{(T)}) \quad (1)$$

Loss Function

Finally, we experimented with adding noise to the non-linear activation function of a neural network, or “noisy activation”, to introduce linear behavior around the zero score range to allow gradients to flow easily when the unit is not saturated, while providing a definitive decision in the saturated regime. We followed the approach by Gülçehre et al. who postulate that the amount of noise added to the activation function should be proportional to the magnitude of saturation of the nonlinearity. [34] Fig. 2 demonstrates this behavior.

4.4. Evaluation procedure

Internally, each RNN model is evaluated using its cost function, multi-label cross-entropy. [35] In general, RNNs should be evaluated based on the cost value, since training is aimed at minimizing that cost. However, when comparing different RNN models among each other to altogether different models, cost function values are not directly comparable. Instead, to allow for easy model comparison, we relied on Mean Reciprocal Rank (MRR) [36] as defined in Equation 2. The MRR metric is used in ranking scenarios with a single true class label. Instead of reporting accuracy measures at making a single guess, this metric lets the evaluated system produce a ranked list of all ATC codes in the sample, ordered by their likelihood of being referred to by the non-standardized input text. Optimal systems will rank the single true ATC code highly (*i.e.* at small numerical ranks $rank_i$) in the output list. This results in a small enumerator component and a large overall MRR score. Overall MRR is reported as the average inverse rank across all test instances. The number of observations in both the training and test sets varies between experiments, from 606 and 177 training and test

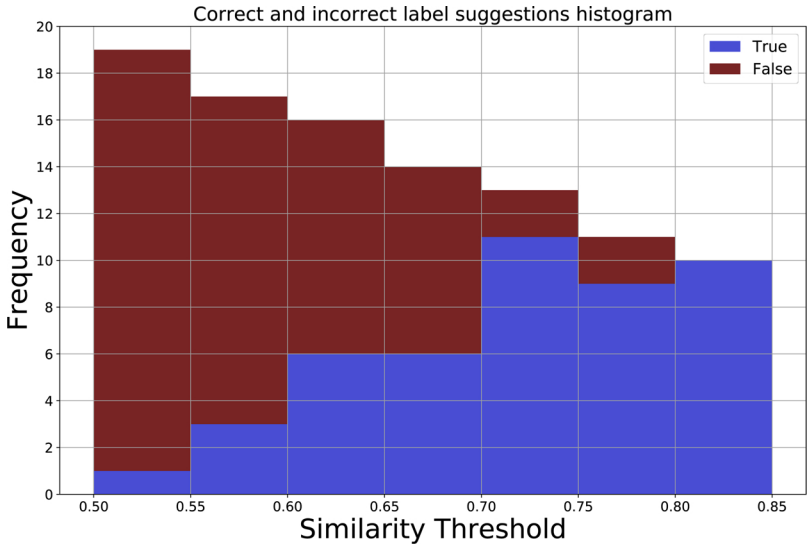

Fig. 1. Bootstrap label propagation accuracy.

examples, respectively (without data augmentation), to 2918 and 780 training and test examples, respectively, under consideration of a similarity threshold of 0.7. This metric is bounded in the [0,1] interval, where an MRR of 1 corresponds to perfect model performance.

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank_i} \quad (2)$$

Mean Reciprocal Rank

To avoid inaccuracies in labels derived from using the similarity bootstrapping procedure, the test set is comprised exclusively of observations from the manually labeled data-set for which there is no doubt as to the ground truth. As a baseline comparison, we included a support vector machine (SVM), with n-gram and bag-of-words features and preprocessing as described above.

Due to the considerable number of tunable model hyper parameters (see appendix), we determine the optimal hyper parameter settings for each model type via 10-fold cross-validation on the training set. Only once this model-specific optimal configuration is found, is the trained model evaluated a single time on the held-out test set. The results of this test set evaluation will be discussed in the following section.

5. Results

Our study compares two RNN architectures (GRU, LSTM), two data feeding directions (feed-forward, bidirectional) and four similarity-based bootstrapping methods (similarity thresholds of 0.7, 0.8, 0.9, 1.0), resulting in a total of n = 2*2*4 = 16 separately trained model variants. The results are shown in Fig. 3. We can note that all compared methods achieved very high performance levels that lie consistently above the MRR = 0.95 mark.

We see that, across all conditions, GRU architectures outperformed all other models. While the bidirectional GRU architecture's performance was roughly stable across different similarity thresholds, the feed-forward models performed worse as the similarity threshold decreased and more noisy training examples were included. While the GRUs were clearly leading the performance comparison, the remaining methods (SVM and LSTM) were difficult to distinguish from each other, showing very similar performance scores and only local variances depending on bootstrapping similarity thresholds.

With only few local exceptions, feed-forward architectures appeared superior to bidirectional methods. Only in case of high levels of training data noise induced by low bootstrapping similarity thresholds, did

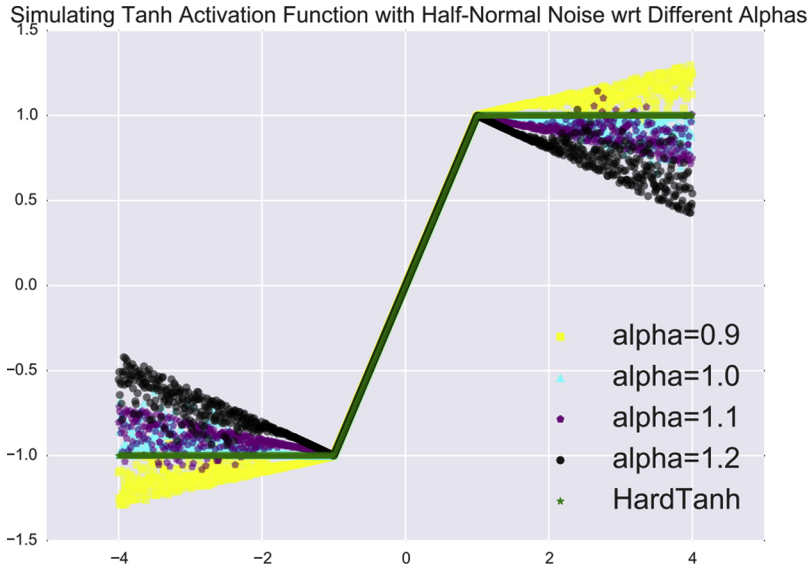

Fig. 2. Noisy Activation.

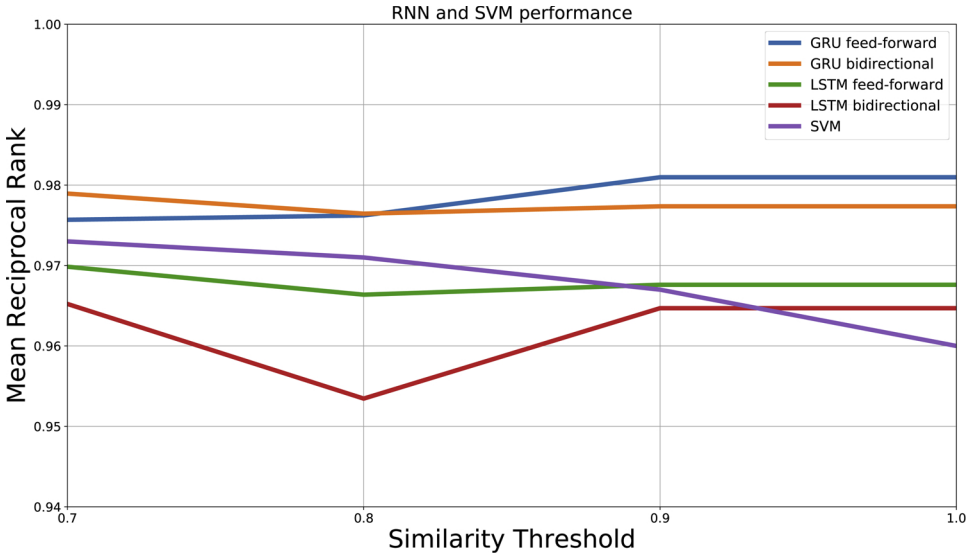

Fig. 3. Global method performance.

bidirectional processing offer advantages.

We can note that the choice of similarity threshold for training data augmentation had only a limited effect on the resulting models' performance. Somewhat surprisingly, it was the SVMs rather than the highly parametric neural networks that benefitted most strongly from the availability of additional noisy training examples. The LSTM models showed a non-monotonic development of MRR scores in response to relaxed similarity thresholds, experiencing a local performance minimum for similarity thresholds of 0.8 from which they recover in both directions. While this augmentation method holds an interesting potential for training different model types (*e.g.*, SVMs vs. neural networks), the overall highest scores are obtained via models trained exclusively on noise-free manually labeled examples.

Aside from these fundamental architectural decisions, we investigated a broad range of hyperparameters including embedding hidden-state sizes, noisy activation, dropout and target replication parameters. While we observed locally effective combinations, the only general trend that could be noted was a beneficial effect of noisy tanh activation over other activation functions. The 16 models presented above are the individually strongest configurations per architecture. Their complete range of hyperparameters is described in the appendix.

Seeing how all compared methods achieve similar performance scores, we conducted a second, alternative performance evaluation in a more challenging setting. Going beyond the previous global performance evaluation, we broke down results per input prescription sequence length, as shown in Fig. 4. The longest individual drug order name was 80 characters long (a hard limitation imposed on this field by the medical center's clinical information system). In this experiment, we restricted the input data that the classifiers receive to only the first k characters, $k \in \{1, \ldots, 80\}$, making it harder for the classifiers to correctly identify the intended ATC code. The models were identical to the previously discussed ones and were not specifically retrained for this more challenging setting. As expected, all models perform better given larger choices of k (*i.e.*, longer input sequence to classify). The extent of this performance detriment, however, varies considerably across model types and data augmentation thresholds. After as few as five processed characters, the feed-forward GRU-based method achieved close-to-optimal performance at recognizing active agents from drug orders. For all other methods, this level of accuracy was reached after having processed approximately 50–70 characters. As the amount of training set noise induced by similarity-based bootstrapping was increased, the number of characters necessary to attain a given performance level increased for all methods. While generally not performing well on short input sequences, again, SVMs show good robustness to noisy training labels.

6. Discussion and conclusion

This study investigated the use of character-level RNN classifiers for automatically categorizing non-standardized text drug orders into groups of ATC codes representing the active ingredients. Our experiments identified an array of neural network architectures that surpassed the quality of more traditional text classification methods such as regression or support vector models.

In comparison with other approaches, Korkontzelos et al. considered DrugBank entries as a dictionary in their investigation. [37] They suggested a boosting approach to increase the number of annotated drugs, in a way similar to the bootstrapping used in the present study to annotate further unlabeled drug orders. However, we did not additionally rely on external resources such as DrugBank.

The sophisticated work by Li et al. used machine learning and NLP in a hybrid algorithm to perform medication reconciliation. [38] This research group analyzed clinical non-standardized text notes and matched identified drugs with their structured drug order counterparts to detect discrepancies. Although they did not apply recurrent neural networks, the involvement of drug order data shows some parallels to our present study.

Still, the use of deep neural networks in "drug name recognition" has been advocated. [39] Chalapathy et al. investigated recurrent neural architectures to recognize drug name mentions in non-standardized text [40].

In summary, to our knowledge, it appears that our study is the first to use recurrent neural networks to categorize non-standardized text drug order entries into groups of active ingredients based on the ATC classification. Since our approach is independent of clinical notes, it could be applied on-the-fly, immediately during computerized provider order entry (CPOE).

Our study has limitations that need to be taken into account in interpreting the results. We used drug ordering data from a single center and only inpatient data were available. While our method achieved high accuracy, coverage remains a concern. In order to be able to assign a given ATC code, the deep learning system requires at least one manually created training example of a non-standardized text order for this active agent. Considering the broad range of available ATC codes, this implies a considerable manual labelling effort in order to arrive at a system that delivers not only high accuracy but also satisfying coverage

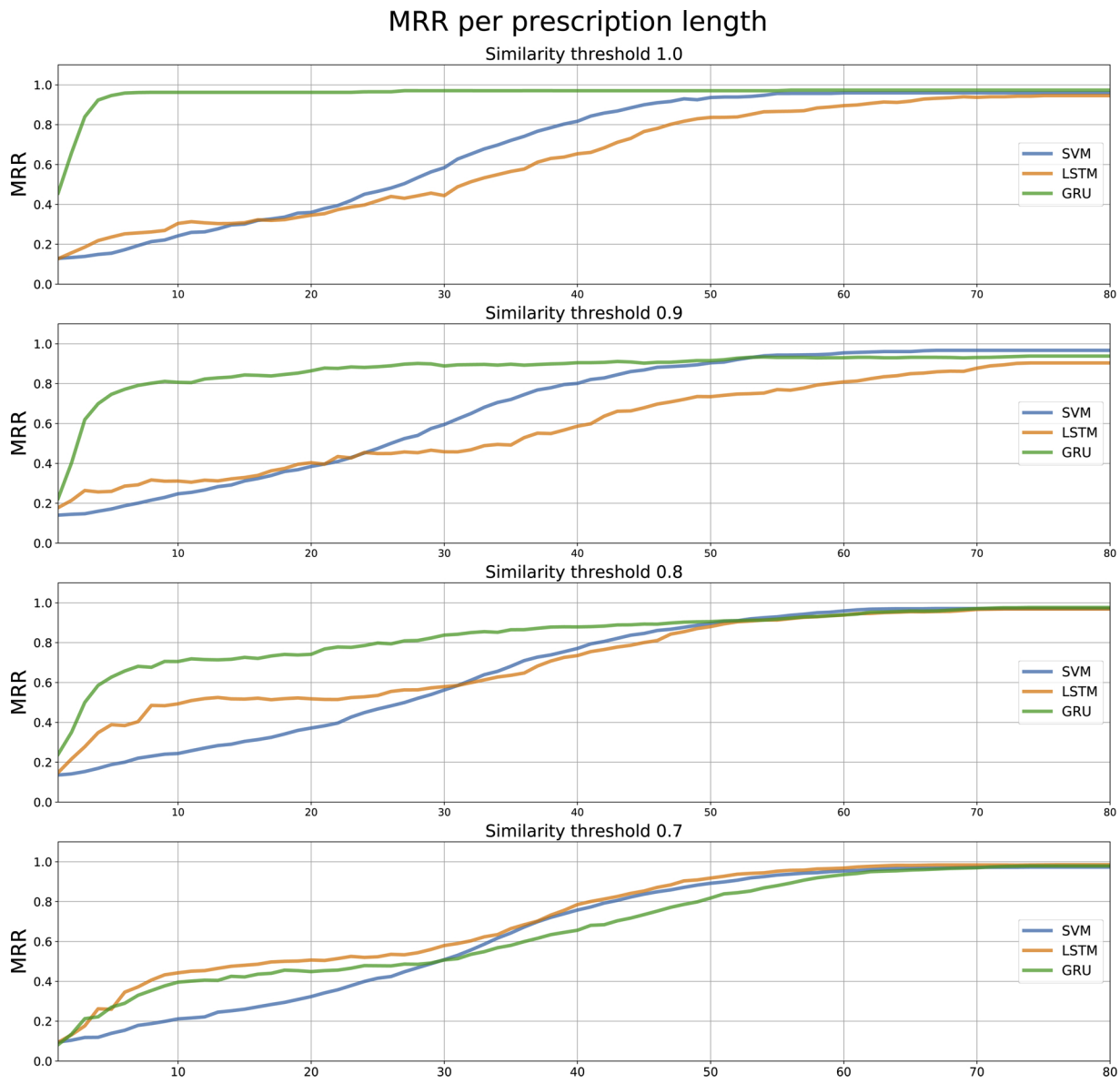

Fig. 4. MRR as a function of input sequence length.

of the common drug order spectrum. In this work, we limited manual labelling to the 7.6% most frequently observed drug order sequences, and in a bootstrapping effort propagated explicit labels to similar unlabeled data points. With the bootstrapping effort, we were able to retrieve additional variations of previously scarcely observed drug order character sequences. This increase in variation enabled us to train our classifiers on these previously rare ATC codes, thus improving the coverage of known ATC codes. However, this does not remedy limited coverage across all drug orders.

This study has implications for EHR-driven healthcare processes and research in domains such as clinical decision support or automated adverse drug reaction identification in cases of non-standardized text drug order entry, but also for administrative tasks, *e.g.* billing and auditing purposes that require considerable manual involvement, as parts of the electronically available data are non-standardized and cannot be automatically processed a priori. The proposed methods are therefore approaches to reduce the manual burden needed for the extraction of categorical information from non-standardized text input.

Additionally, from a computer science and NLP perspective, the present study investigated how limiting the number of available characters of a non-standardized text drug order may influence a reliable recognition of the correct active agent, and we found that highly accurate predictions can be made already very early in the input sequence. This observation holds considerable potential towards creating smart input assistance for clinical information systems that allow clinical staff to efficiently document drug orders by typing only few characters, while not limiting them to a fixed list of exact drug name spellings. Such technology combines the benefits of flexible, highly expressive non-standardized text input and easily interpretable and auditable categorical drug orders, which can be seen as a more tolerant interpretation of the drug order entered. However, there is still a downside to this approach, which leads to opportunities for future research as suggested below.

It is well known that look-alike, sound-alike drug names are sometimes confused by providers, which is a serious threat to patient safety. [41] Lists of drug and brand names requiring particular caution have therefore been put forward as a strategy to mitigate the risk of ordering the wrong therapy [42]. On the one hand, some approaches described here could help to generate lists of similar drug and brand names that should be investigated in terms of the usefulness of automatically extending lists of drug names prone to erroneous ordering. On the other hand, it is likely that “automated tolerant interpretation” of order

entries as mentioned above may increase the risk of misinterpretation and confusion. Hence, there is room for human factors and NLP research to develop approaches reducing the risks arising from look-alike, sound-alike drug and brand names. Therefore, further research is needed around the "human-in-the-loop" model by investigating questions of trust, acceptance, re-traceability and thus ultimately explainability of the results [43].

To address one of the method's current limitations, namely low coverage, we suggest drawing from external resources such as UMLS, Drugbank or Wikipedia in order to obtain a broad, yet accurate overview of drug identifiers. This approach holds the additional benefit of creating a bridge between different national languages. While non-standardized text drug orders for the same active agent may differ vastly based on language and local branding, structured knowledge repositories typically provide pointers to various language variants of the same entity. Still, these datasets often contain very limited amounts of textual variance per drug, due to their structured and normalized nature. Therefore, in order to effectively train a supervised learning model, it is better to compose a dataset from different sources. This would enable achieving a sufficient level of textual variance, necessary for supervised learning.

In conclusion, character-level recurrent deep neural networks can be used to map the active ingredient to non-standardized text of drug order entries, outperforming other representative techniques. While machine learning might help to facilitate categorization tasks, still a considerable amount of manual labelling and reviewing work is required to train such systems. In order to reduce the manual annotation burden per project, there is significant promise in using semi-supervised learning or weak/distant supervision techniques that leverage existing labels from other, related tasks when training new models. This direction should be carefully investigated in future work. The ultimate goal of the proposed approach for the clinical practice would be the "on-the-fly interpretation" of non-standardized text of drug order entries in order to warn the provider against harm potentially induced by the drug. For instance, "penicilin" [sic] could – although misspelled – still trigger an anaphylactic shock if administered to a patient with a penicillin allergy.

Funding statement

Dr. Eickhoff was supported by a grant from the Swiss National Science Foundation (Ambizione #174025). The funding source played no role in the design and conduct of the study; the collection, management, analysis of the data or the interpretation of the results; the review and approval of the manuscript.

Contributorship statement

All authors contributed to the study design, analyses and data interpretation. All authors contributed to the drafting, review and critical revision of this article. All authors approved the final submitted version of the manuscript.

Conflict of interest statement

The authors have no conflict to state.

Summary points

What is already known?

- Drug orders hold considerable value for clinical decision support, prognostic modelling, inferring conditions and forecasting the risk of adverse events
- However, non-standardized text order entries are a well-known problem hampering the automated interpretation of electronically ordered drug therapies

What does this study add?

- We automatically annotated non-standardized text of drug orders with their active ingredients by means of character-level recurrent deep neural networks
- The proposed method achieved a Mean Reciprocal Rank of 98% and outperformed a range of representative alternatives
- While machine learning might help to facilitate categorization tasks, still a considerable amount of manual labelling and reviewing work is required to train such systems.

;1;

APPENDIX

Hyper-parameters Range

Hyper-parameter	Range
RNN cell architecture	LSTM, GRU
Data feed direction	forward-feed, bidirectional
Data vector representation	4 dimensional character-level embeddings, 8 dimensional character-level embeddings, one-hot character-level encoding
Learning rate	10^{-3}, 10^{-2}, 10^{-1}, dynamically changing
Optimization protocol	ADAM optimizer, Stochastic Gradient Descent
Hidden state dimensionality	16, 32, 64, 128
L_2 norm regularization weight	10^{-3}, 10^{-2}, 10^{-1}
Target replication regularization weight	0.3, 0.5
Dropout keep probability	0.5, 0.7, 1.0
Activation function	Tanh, Noisy tanh
P (noisy activation function parameter)	Learnt, 1.0
Alpha (noisy activation function parameter)	0.9, 1.15
Noise (noisy activation function parameter)	Normal, Half-normal

Top 10 Models At Different Similarity Thresholds

Similarity threshold 1.0

Rank	Model Type	Bidirectional	Optimal MRR	Activation Function	Learn p	Alpha	Half Normal Noise	Learning Rate	Keep Probability	Hidden State Size	L2 Norm Constant	Target Replication Constant
1	GRU	F	0.9810	Noisy tanh	F	0.9	T	0.01	1	128	0.01	0.3
2	GRU	F	0.9801	Noisy tanh	T	1.15	T	0.01	0.5	128	0.001	0.5
3	GRU	F	0.9795	Noisy tanh	F	1.15	F	0.01	0.5	128	0.01	0.5
4	GRU	F	0.9795	Noisy tanh	F	0.9	T	0.01	0.5	128	0.01	0.5
5	GRU	F	0.9790	Noisy tanh	T	1.15	F	0.01	0.7	128	0.01	0.3
6	GRU	F	0.9789	Tanh	NA	NA	NA	0.01	0.7	128	0.001	0.5
7	GRU	F	0.9777	Noisy tanh	T	0.9	F	0.01	0.5	128	0.01	0.5
8	GRU	T	0.9774	Noisy tanh	T	1.15	T	0.01	0.5	128	0.001	0.5
9	GRU	F	0.9769	Noisy tanh	T	1.15	F	0.01	0.5	128	0.001	0.5
10	GRU	F	0.9768	Noisy tanh	T	0.9	T	0.01	1	128	0.01	0.3

Similarity threshold 0.9

Rank	Model Type	Bidirectional	Optimal MRR	Activation Function	Learn p	Alpha	Half Normal Noise	Learning Rate	Keep Probability	Hidden State Size	L2 Norm Constant	Target Replication Constant
1	GRU	F	0.9810	Noisy tanh	F	0.9	T	0.01	1	128	0.01	0.3
2	GRU	F	0.9801	Noisy tanh	T	1.15	T	0.01	0.5	128	0.001	0.5
3	GRU	F	0.9795	Noisy tanh	F	1.15	F	0.01	0.5	128	0.01	0.5
4	GRU	F	0.9795	Noisy tanh	F	0.9	T	0.01	0.5	128	0.01	0.5
5	GRU	F	0.9790	Noisy tanh	T	1.15	F	0.01	0.7	128	0.01	0.3
6	GRU	F	0.9789	Tanh	NA	NA	NA	0.01	0.7	128	0.001	0.5
7	GRU	F	0.9777	Noisy tanh	T	0.9	F	0.01	0.5	128	0.01	0.5
8	GRU	T	0.9774	Noisy tanh	T	1.15	T	0.01	0.5	128	0.001	0.5
9	GRU	F	0.9769	Noisy tanh	T	1.15	F	0.01	0.5	128	0.001	0.5
10	GRU	F	0.9768	Noisy tanh	T	0.9	T	0.01	1	128	0.01	0.3

Similarity threshold 0.8

Rank	Model Type	Bidirectional	Optimal MRR	Activation Function	Learn p	Alpha	Half Normal Noise	Learning Rate	Keep Probability	Hidden State Size	L2 Norm Constant	Target Replication Constant
1	GRU	T	0.9764	Noisy tanh	F	0.9	F	0.01	0.7	128	0.001	0.5
2	GRU	F	0.9762	Noisy tanh	T	0.9	T	0.01	0.5	128	0.001	0.5
3	GRU	F	0.9762	Noisy tanh	F	0.9	F	0.01	0.5	128	0.001	0.5
4	GRU	T	0.9753	Noisy tanh	F	0.9	F	0.01	0.5	128	0.001	0.5
5	GRU	T	0.9745	Noisy tanh	F	1.15	F	0.01	0.7	128	0.01	0.5
6	GRU	T	0.9742	Noisy tanh	F	1.15	F	0.01	0.5	128	0.001	0.5
7	GRU	F	0.9739	Noisy tanh	T	1.15	T	0.01	0.5	128	0.001	0.3
8	GRU	F	0.9737	Noisy tanh	T	0.9	F	0.01	0.5	128	0.001	0.5
9	GRU	F	0.9733	Noisy tanh	F	0.9	F	0.01	1	128	0.001	0.3
10	GRU	F	0.9731	Tanh	NA	NA	NA	0.1	1	64	0.001	NA

Similarity threshold 0.7

Rank	Model Type	Bidirectional	Optimal MRR	Activation Function	Learn p	Alpha	Half Normal Noise	Learning Rate	Keep Probability	Hidden State Size	L2 Norm Constant	Target Replication Constant
1	GRU	T	0.9789	Noisy tanh	F	0.9	F	0.01	0.5	128	0.01	0.3
2	GRU	T	0.9785	Noisy tanh	F	1.15	T	0.01	0.5	128	0.001	0.5
3	GRU	T	0.9785	Noisy tanh	F	0.9	T	0.01	0.7	128	0.01	0.5
4	GRU	T	0.9760	Noisy tanh	T	0.9	F	0.01	0.5	128	0.001	0.5
5	GRU	F	0.9757	Noisy tanh	F	0.9	T	0.01	0.5	128	0.001	0.3
6	GRU	T	0.9745	Noisy tanh	F	0.9	F	0.01	0.5	128	0.001	0.3
7	GRU	T	0.9742	Noisy tanh	T	0.9	T	0.01	0.5	128	0.001	0.5
8	GRU	T	0.9736	Noisy tanh	T	0.9	F	0.01	0.7	128	0.001	0.5
9	GRU	T	0.9733	Noisy tanh	T	1.15	F	0.01	0.5	128	0.01	0.5
10	GRU	F	0.9732	Tanh	NA	NA	NA	0.01	0.7	128	0.001	0.3

References

[1] H.M. Seidling, M.D. Paterno, W.E. Haefeli, D.W. Bates, Coded entry versus free-text and alert overrides: what you get depends on how you ask, Int. J. Media Inf. Lit. 79 (11) (2010) 792–796.

[2] A. Holzinger, Big data calls for machine learning [internet], in: R. Narayan (Ed.), Encyclopedia of Biomedical Engineering, Elsevier, Oxford, 2019[cited 2018 Dec 21]. p. 258–64.Available from: http://www.sciencedirect.com/science/article/pii/B9780128012383108773.

[3] E. Eschmann, P.E. Beeler, M. Schneemann, J. Blaser, Developing strategies for predicting hyperkalemia in potassium-increasing drug-drug interactions, J Am Med

Inform Assoc JAMIA 24 (1) (2017) 60–66.
[4] P.M. van den Bemt, A.C. Egberts, A.W. Lenderink, et al., Risk factors for the development of adverse drug events in hospitalized patients, Pharm World Sci PWS 22 (2) (2000) 62–66.
[5] K. Kreimeyer, M. Foster, A. Pandey, et al., Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review, J. Biomed. Inform. 73 (2017) 14–29.
[6] O. Uzuner, I. Solti, E. Cadag, Extracting medication information from clinical text, J. Am. Med. Inform. Assoc. 17 (5) (2010) 514–518.
[7] G. Karystianis, T. Sheppard, W.G. Dixon, G. Nenadic, Modelling and extraction of variability in free-text medication prescriptions from an anonymised primary care electronic medical record research database, BMC Med. Inform. Decis. Mak. 16 (2016) 18.
[8] H. Xu, S.P. Stenner, S. Doan, K.B. Johnson, L.R. Waitman, J.C. Denny, MedEx: a medication information extraction system for clinical narratives, J Am Med Inform Assoc JAMIA 17 (1) (2010) 19–24.
[9] M. Jiang, Y. Chen, M. Liu, et al., A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries, J Am Med Inform Assoc JAMIA 18 (5) (2011) 601–606.
[10] J. Patrick, M. Li, High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge, J Am Med Inform Assoc JAMIA 17 (5) (2010) 524–527.
[11] S. Doan, H. Xu, Recognizing medication related entities in hospital discharge summaries using support vector machine [Internet], [cited 2018 Dec 21]. p..Available from: Association for Computational Linguistics, 2010, pp. 259–266 http://dl.acm.org/citation.cfm?id=1944566.1944596.
[12] F. Hussain, U. Qamar, Identification and Correction of Misspelled Drugs' Names in Electronic Medical Records (EMR) [Internet], [cited 2018 May 18]. p. 333–8.Available from: (2018) http://www.scitepress.org/PublicationsDetail.aspx?ID=8X6z1WV934s=&t=1.
[13] I. Solt, D. Tikk, Yet Another Rule-based Approach for Extracting Medication Information From Discharge Summaries, (2009).
[14] A Simple Rule-based Medication Extraction System, in: Cyril Grouin, Louise Deléger, Pierre Zweigenbaum (Eds.), i2b2 Workshop Proc, 2009 (cf. https://perso.limsi.fr/grouin/bibtexbrowser.php?key=grouin-2009i2b2&bib=publis.bib).
[15] J. G Mork, O. Bodenreider, D. Demner-Fushman, et al., NLM's I2b2 Tool System Description, (2010).
[16] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[17] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[18] S. Hochreiter, Schmidhuber J. Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[19] I. Jauregi Unanue, E. Zare Borzeshi, M. Piccardi, Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition, J. Biomed. Inform. 76 (2017) 102–109.
[20] K. Cho, B. van Merrienboer, C. Gulcehre, et al., Learning Phrase Representations using RNN Encoder-decoder for Statistical Machine Translation, ArXiv14061078 Cs Stat [Internet] 2014;Available from: http://arxiv.org/abs/1406.1078.
[21] S. Gehrmann, F. Dernoncourt, Y. Li, et al., Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives, PLoS One 13 (2) (2018) e0192360.
[22] P. Wang, Y. Qian, F.K. Soong, L. He, H. Zhao, Part-of-Speech Tagging with Bidirectional Long Short-term Memory Recurrent Neural Network, ArXiv151006168 Cs [Internet];Available from: (2015) http://arxiv.org/abs/1510.06168.
[23] B. Plank, A. Søgaard, Y. Goldberg, Multilingual Part-of-Speech Tagging with Bidirectional Long Short-term Memory Models and Auxiliary Loss, ArXiv160405529 Cs [Internet];Available from: (2016) http://arxiv.org/abs/1604.05529.
[24] W. Ling, I. Trancoso, C. Dyer, A.W. Black, Character-based Neural Machine Translation, ArXiv151104586 Cs [Internet];Available from: (2015) http://arxiv.org/abs/1511.04586.
[25] P. Bojanowski, A. Joulin, T. Mikolov, Alternative Structures for Character-level RNNs, ArXiv151106303 Cs [Internet];Available from: (2015) http://arxiv.org/abs/1511.06303.
[26] Z.C. Lipton, D.C. Kale, C. Elkan, R. Wetzel, Learning to Diagnose with LSTM Recurrent Neural Networks, ArXiv151103677 Cs [Internet];Available from: (2015) http://arxiv.org/abs/1511.03677.
[27] M. Gridach, Character-level neural network for biomedical named entity recognition, J. Biomed. Inform. 70 (2017) 85–91.
[28] S.A. Hasan, B. Liu, J. Liu, et al., Neural clinical paraphrase generation with attention, Proc Clin Nat Lang Process Workshop Clin (2016) 42–53.
[29] Y. Kim, Y. Jernite, D. Sontag, A.M. Rush, Character-Aware Neural Language Models, ArXiv150806615 Cs Stat [Internet];Available from: (2015) http://arxiv.org/abs/1508.06615.
[30] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, ArXiv14126980 Cs [Internet];Available from: (2014) http://arxiv.org/abs/1412.6980.
[31] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, Trans Sig Proc 45 (11) (1997) 2673–2681.
[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[33] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-Supervised Nets, ArXiv14095185 Cs Stat [Internet];Available from: (2014) http://arxiv.org/abs/1409.5185.
[34] C. Gulcehre, M. Moczulski, F. Visin, Y. Bengio, Mollifying Networks, ArXiv160804980 Cs [Internet];Available from: (2016) http://arxiv.org/abs/1608.04980.
[35] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, Mit Press, Cambridge, 2016.
[36] N. Craswell, Mean reciprocal rank, Encyclopedia of Database Systems, Springer, Boston, MA, 2009 [cited 2018 Jun 6]. p. 1703–1703.Available from: https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_488.
[37] I. Korkontzelos, D. Piliouras, A.W. Dowsey, S. Ananiadou, Boosting drug named entity recognition using an aggregate classifier, Artif. Intell. Med. 65 (2) (2015) 145–153.
[38] Q. Li, S.A. Spooner, M. Kaiser, et al., An end-to-end hybrid algorithm for automated medication discrepancy detection, BMC Med Inform Decis Mak, (2015), p. 15. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4427951/.
[39] S. Liu, B. Tang, Q. Chen, Wang X. Drug name recognition: approaches and resources, Information 6 (4) (2015) 790–810.
[40] R. Chalapathy, E.Z. Borzeshi, M. Piccardi, An Investigation of Recurrent Neural Architectures for Drug Name Recognition, ArXiv160907585 Cs [Internet];Available from: (2016) http://arxiv.org/abs/1609.07585.
[41] W.L. Galanter, M.L. Bryson, S. Falck, et al., Indication alerts intercept drug name confusion errors during computerized entry of medication orders, PLoS One 9 (7) (2014) e101977.
[42] P. Marco, E. Lopez-Abadia, J. Lucas, More on thromboprophylaxis: electronic alerts in hospitalized patients at risk of venous thromboembolism, Thromb. Haemost. 100 (4) (2008) 525–526.
[43] A. Holzinger, Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Inform. 3 (2) (2016) 119–131.