Medical Persona Classification in Social Media

Nikhil Pattisapu[*], Manish Gupta[*], Ponnurangam Kumaraguru[†], Vasudeva Varma[*]
[*]IIIT-Hyderabad, India
nikhil.pattisapu@research.iiit.ac.in, {manish.gupta,vv}@iiit.ac.in
[†]IIIT-Delhi, India
pk@iiitd.ac.in

***Abstract*—Identifying medical persona from a social media post is of paramount importance for drug marketing and pharmacovigilance. In this work, we propose multiple approaches to infer the medical persona associated with a social media post. We pose this as a supervised multi-label text classification problem. The main challenge is to identify the hidden cues in a post that are indicative of a particular persona. We first propose a large set of manually engineered features for this task. Further, we propose multiple neural network based architectures to extract useful features from these posts using pre-trained word embeddings. Our experiments on thousands of blogs and tweets show that the proposed approach results in 7% and 5% gain in F-measure over manual feature engineering based approach for blogs and tweets respectively.**

I. Introduction

With the advent of social media, users are rapidly sharing (and consuming) medical information, experience through messaging, blogging and video sharing platforms. Medical social media is a subset of the social media, in which the interests of the users are specifically devoted to medicine and health related issues. This includes blogs of health specific portals[1,2] and also blogs and tweets by individuals[3,4].

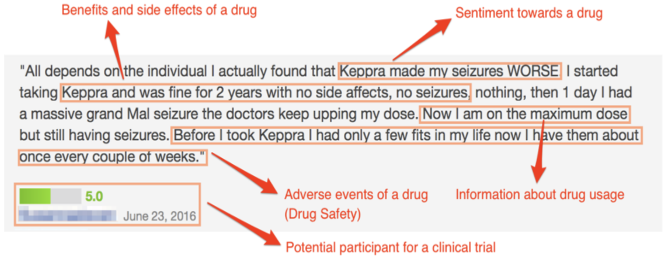
The medical social media contributors usually belong to a variety of *personae*, like patients, caretakers, consultants, researchers, medical journalists and pharmacists [3]. These users use the medical social media for various reasons like seeking answers to specific questions, giving expert advice about a particular drug or a treatment, spreading awareness, sharing experiences, reporting discoveries and findings, voicing opinions and forming communities. This makes medical social media an ideal source of information for several applications like drug marketing and pharmacovigilance[5]. The following examples illustrate some of the uses of medical social media.

- To gather information about drug usage, adverse events, benefits and side effects from patients. Figure 1 shows an excerpt of a post which contains information about a patient's experience with a drug *Keppra*. It contains information about side effects and adverse events of this drug which is crucial for pharmacovigilance.
- To find out the kind of informational assistance sought by caretakers and put such information readily available.
- To identify key opinion leaders in a drug or disease area.
- To find out if a doctor has patients who can take part in a clinical trial.
- To gather information on conversations between pharmacists and others to identify drug dosage, interactions and therapeutic effects.
- To acquire or collaborate on technologies invented by researchers that can be a part of the drug pipeline.
- To gather information about journalists' survey on quality of life of patients.

For most of these applications, identifying medical personae associated with a social media post is of paramount importance, as it helps in categorizing the posts based on persona, which could potentially be useful for several targeted search applications. For example, information about unknown adverse reactions of a drug can be directly obtained from patient or caretaker posts.

Social media consists of social networking sites, blogs, forums, microblogs, wikis and media sharing sites. In this work, we infer the persona from medical blogs and tweets. Both these sources draw a large participation from a variety of medical personae [8], [20]. People of all age groups are comfortable with the technology of blogging. Bloggers have the option of sharing their private medical experiences while maintaining anonymity, thereby avoiding unnecessary embarrassments. In addition to these factors, freshness, ubiquity and diversity makes Twitter one of the most popular social media platforms.

Denecke [3] defines medical persona as user groups and content providers of Web 2.0 applications in health care. Eysenbach et al. [6] identify four categories of medical personae - patients, caregivers, health professionals, biomedical researchers. They also study the behavior of these personae



"All depends on the individual I actually found that Keppra made my seizures WORSE I started taking Keppra and was fine for 2 years with no side affects, no seizures, nothing, then 1 day I had a massive grand Mal seizure the doctors keep upping my dose. Now I am on the maximum dose but still having seizures. Before I took Keppra I had only a few fits in my life now I have them about once every couple of weeks."

5.0
[redacted] June 23, 2016

Benefits and side effects of a drug
Sentiment towards a drug
Adverse events of a drug
(Drug Safety)
Information about drug usage
Potential participant for a clinical trial

Fig. 1. Sample post from drugs.com describing a *patient's* experiences with the drug *Keppra*.

[1] http://blog.patientslikeme.com
[2] http://www.kevinmd.com/blog/
[3] https://twitter.com/SmalIArms/status/825095345453543424
[4] https://lunaoblog.blogspot.in
[5] https://en.wikipedia.org/wiki/Pharmacovigilance

in medical blogs, however, neither of them provide a computational approach to infer the persona from a medical social media post.

In this work, we try to infer the personae associated with a medical social media post. We pose this as a multi-label text classification task where each medical social media post is assigned one or more of the following personae: patient, caretaker, consultant, pharmacist, researcher, journalist, other. One example of a multi-label post could be a post that contains the conversation between a patient and a consultant. This task is challenging because of a variety of reasons.

- No readily available labeled datasets. Labeling is effort intensive.
- It is very hard to identify cues indicative of a persona from text and it often requires extraction of highly domain specific features like number of mentions of inorganic chemicals.
- Social media posts are highly noisy; they are plagued with incorrect spellings, incorrect grammar, non standard abbreviations, lack of punctuation, slang words and out of vocabulary words. These factors make the analysis of social media posts a challenging task.

In this work, we first transform the multi-label classification problem into single-label classification tasks using state-of-the-art problem transformation techniques like Binary Relevance (BR) and Label Powerset (LP) methods. Subsequently, we learn document representations that capture persona specific information from noisy social media posts. Throughout this work, we refer each social media post as a document in order to keep it aligned with the Information Retrieval terminology. Our major contributions in this work are two-fold.

- Firstly, we discover a variety of features which are useful for the task of medical persona classification. These features are manually engineered with the help of domain experts.
- Secondly, we learn the document representations using various deep learning architectures with pre-trained word embeddings which are useful for this task.

The rest of the paper is organized as follows. Section II covers the related work. Section III describes the problem definition. Section IV provides the details about the data collection process and statistics of the data collected. Section V describes our proposed approach for persona classification. Section VI discusses various evaluation metrics used for multi-label classification tasks. Section VII describes all the experiments conducted along with implementation details. Section VIII consists of results of our experiments on blog and tweet datasets. In Section IX, we provide a detailed analysis of our results and discuss the advantages and limitations of our approach. We conclude with a summary of the work and ideas for future work in Section X.

II. Related Work

Eyesenbach et al. [6], were amongst the first to identify main personas associated with medical social media - patients, health professionals, biomedical researchers. Denecke [5] conducted manual studies and established the existence of a relation between medical persona and blog content. This supports the basic assumption of our work, that social media post content can indeed be used to identify medical personae. However, neither [5] nor [6] provide a computational approach to automatically identify medical persona associated with a particular social media post. To the best of our knowledge, we are the first ones to computationally address this problem in medical domain. The following subsections provide a review of two closely related problems - automatic genre classification and authorship attribution.

A. Automatic Genre Identification

Automatic Genre Identification (AGI) deals with classifying documents based on genres (which includes their form, structure, functional trait, communicative purpose, targeted audience and narrative style) rather than the content, topics or subjects that the documents span [2]. Examples of genre on the web include editorial, interview, news, advertisements, shopping and blog pages. In the past, AGI has been setup both as an open-set [22] and closed-set classification task [12], as well as single-label and multi-label classification task [27]. A variety of features have been explored for AGI - common words, function words, word unigrams, character n-grams, part-of-speech tags, part-of-speech trigrams, document statistics and HTML tags [2]. Sharoff et al. [25] compare a wide range of features and report that word unigrams and character 4-grams are the most useful features for this task. Our task is of closed-set multi-label type, and besides extensive domain-specific feature engineering, we also explore deep learning methods.

B. Authorship Attribution

Authorship attribution is the task of determining the author of a particular document. The underlying assumption in this task is that each individual has a distinctive way of writing, such as patterns of vocabulary usage, unusual language usage, particular syntactic and structural layout traits, stylistic and sub-stylistic features, which will remain constant across multiple documents [10]. The main challenge in this task is to extract these signals. A variety of features have been explored for this task - orthographic features [4], TF-IDF of word unigrams, bigrams and trigrams, hashtag and reply mentions [15], stylistic features, functional words, acronyms, lexical and placement related features. Stamatatos et al. [26] provide a detailed survey of works on authorship attribution. Our work is similar to this area in the sense that each persona can be considered like an author. But within each persona, there are multiple users and hence each persona will itself contain heterogeneity. Hence, solutions for authorship attribution cannot be trivially applied for our task; more complex mechanisms need to be developed carefully.

III. Problem Definition

In this work, we attempt to identify the personae associated with a medical social media post. We set this up as a multi-

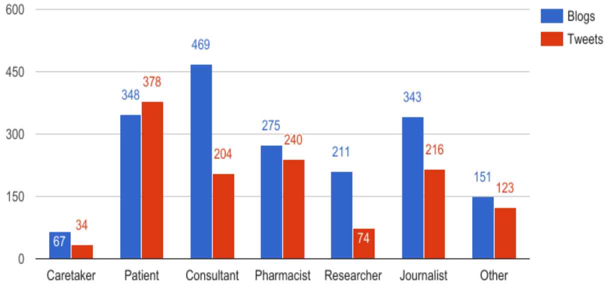


Fig. 2. Class Distribution across blogs and tweets

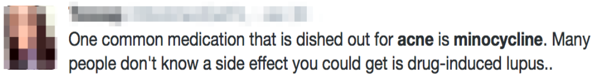

[redacted]
One common medication that is dished out for **acne** is **minocycline**. Many people don't know a side effect you could get is drug-induced lupus..

Fig. 3. Sample tweet

label classification problem with the labels, *Patient, Caretaker, Consultant, Journalist, Pharmacist, Researcher, Other*. We experiment with two datasets - blogs and tweets. Equation 1 describes our dataset.

$$D = \{(d_i, l_i) : l_i \subseteq L \; \forall \; i = 1, 2 \cdots N\} \tag{1}$$

where d_i, is the i^{th} document or post, l_i is the label set corresponding to document d_i, N is the total number of documents and L is the superset of all label sets. In Section V, we propose several approaches to discover the label set l_i corresponding to a document d_i.

IV. Dataset

For data collection, we use a set of 50 most popular drug names (obtained from *drugs.com*) as queries. We query the Twitter search API[6] and Twingly Search API[7] to retrieve the matching tweets and blogs respectively. We randomly sample 50 blogs and 30 tweets per query from the retrieved posts. Posts containing ill-parsed content, advertisements, duplicate or near duplicate content are removed using standard techniques. The resulting blogs and tweets were manually looked at by the human annotators and were assigned following persona labels: Patient, Caretaker, Consultant, Pharmacist, Researcher, Journalist, Other. A total of 1581 blogs and 1025 tweets were annotated. The class distribution for both blogs and tweets is shown in Figure 2. The inter-annotator agreement between 4 annotators was found to be 0.708 for blogs and 0.70 for tweets. The label cardinality of blogs and tweets was 1.18 and 1.24 respectively. The maximum label cardinality of a blog was 2 and that of a tweet was 3.

Whenever, using only drugs as queries resulted in a lot of irrelevant content (which is of little use to a pharmaceutical firm), drug-disease pairs were used as queries. Figure 3 shows an example tweet retrieved in response to the query *acne mynocyline* where *acne* is the medical condition or disease name and *minocycline* is the drug used to cure it. We have released our code and dataset to facilitate research in this direction[8].

V. Approach

We first transform the multi-label classification problem into a single-label classification problem using state-of-the-art label transformation methods. We then extract several features for this task using manual feature engineering and word-embedding based methods. Finally we classify the documents into one or more of the aforementioned personae using a supervised machine learning classifier like SVM or softmax. We compare these document representations using metrics proposed in Section VI.

A. Label Transformation Methods

We use two label transformation methods: Label Powerset (LP) and Binary Relevance (BR). Both the methods have been successfully used in several multi-label text classification problems [7], [24].

In the LP method, we train a single binary classifier for each label combination found in the training set. Given an unseen example, we assign the label set corresponding to the classifier which outputs the highest probability of positive outcome as shown in Equation 2, where L is the labelset, 2^L is the label powerset and $p(l,i)$ is the output of the l^{th} classifier for unseen test example i.

$$l_i \; = \underset{l \in 2^L}{\text{argmax}} \; p(l,i) \tag{2}$$

In the BR method, we independently train one classifier for each label. Traditionally, given an unseen example, we typically output the labels corresponding to the classifiers which returned positive outcome with a probability greater than a threshold as shown in Equation 3.

$$l_i \; = \{ l : p(l,i) \geqslant t^l \; \} \quad \forall \; l \in 1, 2, \ldots, |L| \tag{3}$$

The drawback of this approach is that the individual label thresholds t^l need to be calibrated for all labels $l \in L$. We therefore propose a method (shown in Equation 4) to select the label set for the unseen example. In Equation 4, p_i^{max} is $max(p(l,i)) \;\; \forall \; l \in 1, 2, \ldots, |L|$ and t is the single threshold which we need to calibrate. Equation 4 ensures that the predicted label set for any example is never empty. The threshold value t is chosen based on a grid search which results in the best performance on a hold-out validation set. The resultant label set l_i is mapped to its corresponding single-label in the label powerset 2^L.

$$l_i = \{ l : p(l,i) \geqslant t * p_i^{max} \} \quad \forall \; l \in 1, 2, \ldots, |L| \tag{4}$$

B. The N-gram Approach

In this approach, each document is represented as a TF-IDF vector over the entire vocabulary. Then, a Support Vector Machine (SVM) is trained to classify the document into one or more of the pre-defined personae. Apart from word unigrams, we also use character n-grams as features for this task. Both

[6] https://dev.twitter.com/rest/public/search

[7] https://developer.twingly.com/resources/search/

[8] https://drive.google.com/open?id=0B_9ISEpIrWxEWmRIazVJZS1JTFE

these representations have been successfully used in similar tasks like automatic genre identification [2], [25] and authorship attribution [10]. We treat these document representations (word unigrams and character n-grams) as our baselines.

C. Feature Engineering Approach

We worked with subject matter experts to identify potentially useful 89 features in categorizing the medical blogs or tweets based on their personae. Besides the word features, such features can be used to learn various classifiers. We discuss these features in detail.

Document Level features (4 features) capture generic features of the document like number of sentences, number of words, average sentence length in words, and average word length. The rationale for choosing these features is that, irrespective of the semantics of the content, the amount of content and its organization in a medical blog or tweet are heavily influenced by the persona associated with it. For example patient and caretaker blogs tend to be shorter, whereas pharmacist or consultant blogs are long. In our labeled dataset, the average number of sentences in patient blogs is 16 whereas the average number of sentences in pharmacist blogs is 32.

Part of speech (POS) features (33 features) capture the distribution of different parts of speech in the document. The distribution across POS tags reveals useful information about the persona. For example, in our dataset, a consultant is 1.6 times more likely to use adjectives than a journalist.

List lookup based features (7 features) include the average frequency of terms which occur in the document as well as in a particular list. These lists contain the most frequently used words by different personae and are manually created. Typically each list consists of terms which are predominantly used by one persona and sparingly used by others. For example, the terms ‘MD’, ‘Dr.’, ‘MBBS’, ‘FRCS’, ‘consultation fee’, were found to be more frequent in consultant blogs than others.

Syntactic features (7 features) capture the presence or absence of various classes of terms like number of mentions of date, person, location, organization, time, money, and percentage amounts.

Semantic features (35 features) consist of a lot of medical domain specific features like number of disease mentions, drug mentions, chemical mentions, organ mentions, symptom mentions, antibiotics, biomedical occupation, body location, body substance, body system, clinical attributes, conceptual entities, events, findings, food, functional concepts, health care activity, health care organization mentions, inorganic chemical mentions, organs, organic chemical mentions, patient mentions, pharmocologic substances, phenomenon, procedure, research activities, substances, and temporal concepts. The distribution across these features gives significant clues about the persona. We extract these features using MetaMap [1].

Tweet specific features (3 features) consist of number of hashtags, number of user mentions, and if a tweet is a reply or not.

D. Averaged Word Vectors

Recently, word embeddings are increasingly being used to capture document semantics [14], [19]. Averaged word vectors is another variant of bag-of-words model where we represent a document vector as the mean embedding of all the words present in the document as shown in Equation 5.

$$document\ vector(d_i) = \sum_{w_j^i} \frac{word\ embedding(w_j^i)}{len(p_i)} \tag{5}$$

where d_i is the i^{th} document, w_j^i is the j^{th} word in document d_i and $len\ (d_i)$ is the total number of words in document d_i. The word embeddings are usually learnt from an external corpora using algorithms like Word2Vec [18] or GloVe [21]. Recently, averaged word vectors showed comparable performance to state-of-the-art approaches in several text classification tasks like sentiment analysis [11].

E. CNN-LSTM

The Averaged Word Vector model uses only word unigrams. It loses the word order and also assigns equal weight to all the words in a document. This behavior might be undesirable for persona classification. Consider the below example where the first sentence contains no persona specific information, whereas the second sentence gives substantial clue that the document belongs to a *caretaker* persona.

> Winter brings a lot of health related issues. My son catches fever every winter.

We propose a Convolutional Neural Network (CNN) based architecture (Figure 4) to extract features from word n-grams of a sentence. These features are subsequently used for persona classification in tweets. For blogs, where the sequence of words is comparatively long; we propose CNN-LSTM model (Figure 5) where we use Convolutional Neural Networks to extract features from word n-grams of sentences and use LSTMs to preserve persona specific information through sentence sequences. Our architectures are inspired by [17] (network originally trained for medical concept normalization) and [28] (network originally trained for sentiment analysis). We now describe the layers used in both the architectures.

1) Pre-trained Word Embedding Layer: We use pre-trained word embeddings for all our models - averaged word vectors model, CNN and CNN-LSTM model. We experiment with several pre-trained word embeddings which can be majorly categorized into two types: embeddings trained on purely medical text and embeddings trained on generic text. Table I shows the specifics of the word embeddings used[9].

2) Convolution Layer: Convolution layer takes as input the sequence of word embeddings in a sentence of the form $x_1, x_2 .....x_n$ where $x_i \in \mathbb{R}^d$, d is the dimensionality of the word embedding and n is the maximum sentence length in our dataset. In case of processing tweets, the convolution

[9] Embedding ID’s 3, 4 were trained on different Web Crawls.

ID	Training Source	Training Algorithm	#Dim	#Entries	Domain
1	Medical Tweets (ADR) [16]	Word2Vec	200	1344629	Medical
2	Twitter [21]	GloVe	200	1193515	Generic
3	Web crawl 1 [21]	GloVe	300	2196018	Generic
4	Web crawl 2 [21]	GloVe	300	1917495	Generic
5	PubMed, PMC, Wikipedia [23]	Word2Vec	200	5443656	Medical

TABLE I
PRE-TRAINED WORD EMBEDDING DETAILS

layer takes the entire tweet as input. Whenever, the sentence length is less than n, appropriate number of zero vectors are appended to create fixed length sentence representation. For every sentence, we use K filters to learn features from its word n-grams. In a window of h words $x_{i:i+h-1}$ a filter $\boldsymbol{F_k} \in \mathbb{R}^{\boldsymbol{hd}}$ generates a feature map c_i^k (a scalar) as shown in Equation 6.

$$c_i^k = f(\boldsymbol{F_k} \odot \boldsymbol{x_{i:i+h-1}} + \boldsymbol{b^k}) \tag{6}$$

where $\odot$ is the convolutional operator, and $\boldsymbol{b^k}$ is the bias, f is the activation function like the Rectified Linear Unit (ReLU) which is defined as $f(z) = max(0, z)$ or sigmoid which is defined as $f(z) = 1/1 + e^{-z}$. Each filter $\boldsymbol{F_k}$ traverses the input and outputs feature maps $c_1^k, c_2^k, ..., c_{n-h+1}^k$.

3) Max-pooling Layer: In max pooling layer, we subsample the output of each filter $\boldsymbol{F_k}$ by applying a max function $max(c_1^k, c_2^k, ..., c_{n-h+1}^k)$. Max-pooling eliminates non-maximal values thereby reducing computation and preserving salient local features for subsequent layers to process. The output of all the max-pooled filters are subsequently merged to form a fixed size sentence embedding.

4) Sequential Layer: Only the CNN-LSTM model has a sequential layer. This layer takes as input a sequence of sentence embeddings of the form $s_1, s_2 \ldots s_m$, where $s_i \in \mathbb{R}^K$ is the sentence embedding and m is the maximum number of sentences in any document of our dataset. Whenever, number of sentences in a document is less than m, appropriate number of zero vectors of size K are appended to create a fixed length sentence sequence. To capture long-distance dependencies across the sentence sequence, this layer sequentially integrates each sentence embedding into a text embedding. We employ a Long Short Term Memory (LSTM) cell [9] in this layer to aggregate and preserve crucial persona information through the sequence of sentence embeddings. After the LSTM memory cell sequentially traverses through all sentence embeddings, the last hidden state of this layer is regarded as the document embedding.

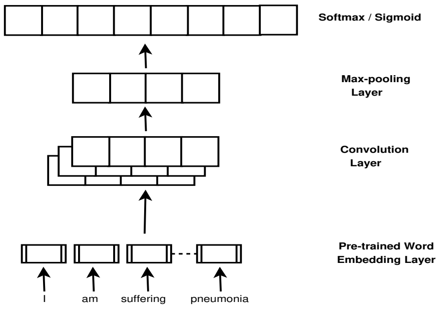


Fig. 4. Architecture of the CNN model

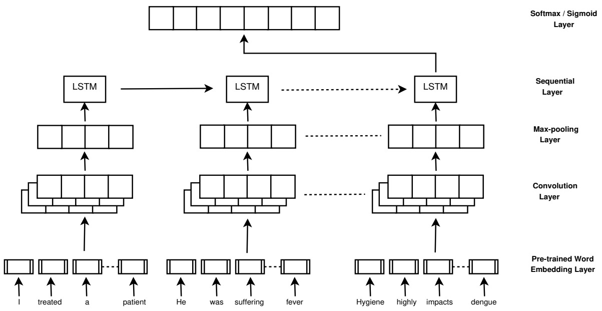


Fig. 5. Architecture of CNN-LSTM model

5) Softmax or Sigmoid Layer: The softmax or sigmoid layer is a fully connected dense layer which takes a document vector X_i (produced by sequential layer in the case of CNN-LSTM model or max-pooling layer in the case of CNN model) as input and produces the output vector $\hat{y_i}$ as shown in Equation 7 where i is the document index, W is the weight matrix, b is the bias vector and f is the activation function. In this layer we experiment with two choices of activation functions - sigmoid (Equation 8) and softmax (Equation 9). Here J is the total number of neurons in the output layer.

$$\hat{y_i} = f\ (W^T X_i + b) \tag{7}$$

$$sigmoid(z)_j = \frac{1}{1 + e^{-z_j}} for\ j = 1, 2, \ldots, |J| \tag{8}$$

$$softmax(z)_j = \frac{e^{z_j}}{\sum_{j=1}^{J} e^{z_j}} for\ j = 1, 2, \ldots, |J| \tag{9}$$

The activation function dictates the choice of loss function. If we use sigmoid activation, we use binary cross entropy loss function (Equation 10). On the other hand, if we use softmax activation function, we use categorical cross entropy loss function (Equation 11)

$$Loss(y, \hat{y}) = \sum_{i=1}^{N} \sum_{l=1}^{|J|} -y_i^l\ log\ (\hat{y}_i^l) +\ (1-y_i^l)\ log\ (1-\hat{y}_i^l) \tag{10}$$

$$Loss(y, \hat{y}) = \sum_{i=1}^{N} \sum_{l=1}^{|J|} y_i^l\ log\ \hat{y}_i^l \tag{11}$$

where N is the total number of documents in training, y_i is the label vector corresponding to label set l_i and y_i^j, $\hat{y}_i^j$ are the j^{th} component of the vectors y_i and $\hat{y}_i$ respectively. Our model is trained using the stochastic optimization method Adam [13] by minimizing the aforementioned loss functions.

VI. Evaluation Metrics

Each evaluation metric is described on a per instance basis which is subsequently averaged over all instances to obtain the aggregate value. Let l and pr be the true label set and predicted label set for document d_i. Exact match is defined as

$$\begin{cases} 1 \; if \; l = pr \\ 0 \; otherwise \end{cases}$$

Jaccard similarity is defined as $|l \cap pr|/|l \cup pr|$, The precision, recall and F-score are defined as $P(d_i) = |l \cap pr|/|pr|$, $R(d_i) = |l \cap pr|/|l|$ and $F(d_i) = 2P(d_i)R(d_i)/(P(d_i) + R(d_i))$ respectively. The Hamming Loss is defined as $\frac{\sum_{j=1}^{|L|} xor(l^j, pr^j)}{|L|}$ where l^j, pr^j denote j^{th} element of l and pr respectively. The Hamming Score is defined as $1 - Hamming \; loss$.

VII. Experiments

Throughout this work, we report results for 10-fold cross validation experiments where the dataset is shuffled and sliced into 10 equal parts. 9 of these parts are iteratively selected to train the model and the remaining part is used to test the model. This process is repeated until all of the parts are used for testing.

We use *scikit-learn* for implementing TF-IDF word unigram and character n-grams based models. We conduct experiments with character bigrams, trigrams and 4-grams. We found that, amongst character n-grams, character 4-grams were the most useful features. For feature extraction, we use Stanford Named Entity Recognizer, Stanford Parts-of-speech tagger and 2016 version of MetaMap [1]. MetaMap identifies a wide range of semantic types some of which are shown in Table II. In our dataset, we found a total of 189 semantic mappings. We found that the performance of MetaMap was poor when applied on tweets. This is not surprising as MetaMap is primarily designed to extract medical entities from biomedical documents. Therefore, MetaMap wrongly identifies several *non-medical* concepts as medical concepts. For example the word *I* is mapped to the concept *Iodine*. We observed that, the execution time of MetaMap increases exponentially with increase in document length. Therefore, we split the blogs into sentences first and then extract the semantic categories from the sentence. This however, had adverse affects on the quality of semantic features extracted.

All the experiments involving neural networks were realized using Keras with Theano backend on a Tesla K40 GPU. We ran the CNN model as well as CNN-LSTM model for 100 epochs. In every epoch 10% of labeled data was used as validation set, we stopped training the neural network when the validation loss kept increasing for a fixed number of consecutive epochs to avoid overfitting. The training time for CNN and CNN-LSTM models were 15 minutes and 2.5 hours respectively.

For tuning hyperparameters in CNN and CNN-LSTM models, we used a grid search over the entire hyper-parameter space which includes number of filters, filter sizes, activation functions for CNN ($ReLU$ and $sigmoid$), activation functions of the output layer ($softmax$ or $sigmoid$), size of the sequence layer hidden unit, number of epochs after which training should be stopped, if validation loss kept increasing. These hyper parameters were evaluated against a hold-out validation set. The parameters resulting in maximum F-Score were used for training the network.

For blogs dataset, we found $h \;=\; 2,\; 3,\; 4$ to be the most optimum filter sizes. We realized 200 features maps (with ReLU activation) for each filter size, thereby representing each sentence over 600 dimensions (F_l). For tweets dataset, we found $h \;=\; 2,\; 3,\; 5$ to be the most optimum filter sizes. We realized 100 feature maps (with ReLU activation) for each filter size, thereby representing a tweet over 300 dimensions. For both CNN, CNN-LSTM methods, we stopped training the network if validation loss kept increasing continously for 8 epochs. Similarly, we found sigmoid activation to be effective for BR method and softmax activation to be effective for LP method in both CNN and CNN-LSTM models. We also conducted experiments with paragraph2vec based document representations which resulted in sub-optimal performance and therefore are not discused here.

VIII. Results

Tables III and IV show the comparative performance of all approaches and Label Transformation methods (LT Method) across four metrics - Jaccard Similarity (JS), Exact Match (EM), Hamming Score (HS) and F-score. For approaches, which depend on word embeddings, we report the results with the best embedding. Tables III, IV show that embedding 4 was most effective for both blogs and tweets. Random word embedding initializations (in CNN and CNN-LSTM models) resulted in sub-optimal performance (average decrease in F-score was found to be between 8-12%). In both CNN, CNN-LSTM models, we keep the pre-trained word embeddings fixed. Experiments where the word embeddings were trained along with other parameters resulted in sub-optimal performance (average decrease in F-score was between 3% - 5%).

For blogs dataset, Averaged Word2Vec model and for tweet dataset, CNN model outperforms other approaches. The performance of both these approaches across various embeddings is shown in Tables V and VI.

IX. Analysis and Discussion

Tables III and IV show that our proposed approaches significantly outperform both the baselines. For blogs dataset, Averaged Word2Vec model and for tweet dataset, CNN model outperforms other approaches. We observe that even though the CNN-LSTM performs better than the word unigrams, character n-grams and feature engineering based approaches, it fails to outperform averaged Word2Vec model. We attribute this to huge number of model parameters and lesser number of training examples. Tables V and VI show that the performance of pre-trained word embeddings 3, 4 [21] is superior to other embeddings which have a relatively, wider coverage of medical concepts [16], [23]. This shows that coverage of medical concepts is not crucial for the task of persona classification.

Text	Identified Phrase	Semantic types	Interpretation
I treated a cancer patient yesterday.	I	Inorganic Chemical	Iodides
		Amino Acid, Peptide, or Protein, Immunologic Factor	Blood group antibody I
	treated	Functional Concept	Treating
	cancer patient	Patient or Disabled Group	-

TABLE II
PERFORMANCE OF METAMAP ON SAMPLE TEXT

Approach	LT Method	Emb Id	JS	EM	HS	F-Score
Word unigrams	BR	-	0.446	0.393	0.870	0.520
	LP		0.566	0.511	0.865	0.570
Character n-grams	BR	-	0.460	0.401	0.871	0.530
	LP		0.577	0.523	0.868	0.580
Feature Engineering	BR	-	0.461	0.409	0.872	0.530
	LP		0.574	0.518	0.867	0.580
Averaged Word2Vec	BR	3	0.608	0.521	0.880	0.600
	LP	4	**0.627**	**0.568**	**0.886**	**0.640**
CNN-LSTM	BR	3	0.496	0.421	0.846	0.460
	LP	3	0.586	0.514	0.869	0.600

TABLE III
RESULTS OF ALL APPROACHES FOR BLOGS

Approach	LT Method	Emb Id	JS	EM	HS	F-Score
Word unigrams	BR	-	0.427	0.352	0.862	0.500
	LP		0.518	0.441	0.846	0.510
Character n-grams	BR	-	0.421	0.353	0.864	0.480
	LP		0.513	0.435	0.845	0.490
Feature Engineering	BR	-	0.450	0.366	0.865	0.520
	LP		0.540	0.455	0.852	0.540
Averaged Word2Vec	BR	3	0.563	0.469	0.863	0.560
	LP	4	0.544	0.462	0.853	0.520
CNN	BR	4	**0.593**	**0.499**	**0.873**	**0.590**
	LP	4	0.582	0.489	0.864	0.580

TABLE IV
RESULTS OF ALL APPROACHES FOR TWEETS

LT Method	Emb Id	JS	EM	HS	F-Score
BR	1	0.392	0.273	0.784	0.380
	2	0.587	0.502	0.874	0.580
	3	0.608	0.521	0.880	0.600
	4	0.602	0.516	0.879	0.600
	5	0.585	0.498	0.872	0.570
LP	1	0.222	0.194	0.760	0.160
	2	0.607	0.550	0.879	0.610
	3	0.618	0.560	0.883	0.630
	4	**0.627**	**0.568**	**0.886**	**0.640**
	5	0.594	0.536	0.875	0.600

TABLE V
RESULTS OF AVERAGED WORD VECTOR APPROACH FOR BLOGS

LT Method	Emb Id	JS	EM	HS	F-Score
BR	1	0.507	0.416	0.845	0.500
	2	0.574	0.482	0.867	0.570
	3	0.580	0.488	0.867	0.570
	4	**0.593**	**0.499**	**0.873**	**0.590**
	5	0.562	0.460	0.864	0.560
LP	1	0.482	0.401	0.830	0.490
	2	0.572	0.480	0.861	0.570
	3	0.565	0.474	0.859	0.560
	4	0.582	0.489	0.864	0.580
	5	0.548	0.462	0.854	0.550

TABLE VI
RESULTS OF CNN APPROACH FOR TWEETS

Another plausible explanation for a poor performance of [16], [23] is that, they are trained on medical text written by a specific persona (For example, embedding 5 [23] is trained on text written by *Researchers* and embedding 1 [16] is trained on text written by *Patients*). Therefore, these embeddings fail to capture "aspects" of other personae.

For feature analysis, we rank all the features according to their χ^2 value. Table VII shows the top three features in each feature group. Total number of features in blogs and tweets were 40762 and 6343 respectively. We note that, feature engineering approach outperforms the baseline methods. After analyzing the confusion matrices [10] for different approaches, we found that across all the approaches, the *caretaker* class has the least accuracy, this is due to less number of training examples in this class and high vocabulary overlap with the *patient* class.

X. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the problem of medical personae classification of social media posts. We investigated manual feature engineering as well as neural network based approaches for this problem. We experiment with two datasets - blogs, tweets. We find that CNN model gave the best performance on tweet dataset and Averaged Word2Vec model gave the best performance on blog dataset. We experiment with various kinds of pre-trained word embeddings and find that coverage of medical concepts is not as crucial as the diversity of text (w.r.t persona) on which the embeddings were learnt.

Both CNN and CNN-LSTM models have a large number of parameters and need more labeled training examples for higher performance. However, adding more training examples involves more labeling which is very costly (effort intensive, and time consuming). This motivates us to look for collecting heuristically labeled examples (techniques motivated by distant supervision). It is possible to find persona specific content on several web portals. For example, *kevinsMD.com* has many blog posts from consultants and patients which are actually tagged with semantic tags like 'physician' and 'patient'. These semantic tags can be exploited to crawl and segregate such blogs. However, the problem in using these (distant supervision based) methods is the trade-off between quantity of training data and quality of labeling. In such situations, the document representations which are robust to noisy labels are expected to give superior performance. We want to extend our current work by collecting heuristically

[10] not shown here due to space constraints

Feature Group	Best Feature (Blogs)	Best Feature (Tweets)
Document Feature	# characters (3) # words (4) # sentences (5)	# characters (8) # sentences (10) # words (18)
Syntactic Feature	# Money mentions (2) # location mentions (27163) # organization mentions (36685)	# Money mentions (6) # Percentage mentions (7) # organization mentions (4683)
List lookup Feature	# matching words with consultant list (1) # matching words with journalist list (410) # matching words with researchers list (546)	# matching words with patient word list (29) # matching words with researchers list (64) # matching words with journalist list (126)
Semantic Feature	# Inorganic chemical (38) # research activity (201) # temporal concepts (649)	# research activity (34) # inorganic chemical (36) # of diseases (596)
POS Feature	# Foreign word (163) # Verb past tense (182) # Personal Pronoun (256)	# Personal Pronoun (116) # Nouns (257) # Determiner (375)
Tweet specific Features	-	# hashtags (9) if a tweet is a reply (10) # user mentions (12)

TABLE VII

FEATURE ANALYSIS FOR BLOGS AND TWEETS BASED ON χ^2 METRIC. NUMBER IN THE PARENTHESIS INDICATES FEATURE RANK (LESSER THE BETTER)

labeled posts and study the performance gain / loss across various approaches. Also, as the current features are limited to a post's content, we would like to explore other features like social features, for example, number of followers on Twitter.

REFERENCES

[1] A. R. Aronson. Effective Mapping of Biomedical Text to the UMLS Metathesaurus: the MetaMap Program. In *Proceedings of the AMIA Symposium*, pages 17–21. American Medical Informatics Association, 2001.

[2] N. R. Asheghi, K. Markert, and S. Sharoff. Semi-supervised Graph-based Genre Classification for Web Pages. *TextGraphs-9*, pages 39–47, 2014.

[3] K. ckecke and W. Nejdl. How valuable is Medical Social Media data? Content Analysis of the Medical Web. *Information Sciences*, 179(12):1870–1880, 2009.

[4] O. De Vel, A. M. Anderson, M. W. Corney, and G. M. Mohay. Multi-topic E-Mail Authorship Attribution Forensics. In *Proceedings of the ACM Conference on Computer Security - Workshop on Data Mining for Security Applications*. ACM, 2001.

[5] K. Denecke. Social Media Data For Healthcare. In Y. Zhang, editor, *Health Web Science*, chapter 6, pages 33–49. Springer, 2015.

[6] G. Eysenbach. Medicine 2.0: Social Networking, Collaboration, Participation, Aomediation, and Openness. *Journal of medical Internet research*, 10(3):e22, 2008.

[7] S. Gopal and Y. Yang. Multilabel Classification with Meta-level Features. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322. ACM, 2010.

[8] C. Hawn. Take two Aspirin and Tweet me in the Morning: how Twitter, Facebook, and other Social Media are Reshaping Health Care. *Health affairs*, 28(2):361–368, 2009.

[9] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] J. Houvardas and E. Stamatatos. N-gram Feature Selection for Authorship Identification. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 77–86. Springer, 2006.

[11] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759*, 2016.

[12] B. Kessler, G. Numberg, and H. Schütze. Automatic Detection of Text Genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 32–38. Association for Computational Linguistics, 1997.

[13] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] M. J. Kusner, Y. Sun, N. I. Kolkin, K. Q. Weinberger, et al. From Word Embeddings To Document Distances. In *ICML*, volume 15, pages 957–966, 2015.

[15] R. Layton, P. Watters, and R. Dazeley. Authorship Attribution for Twitter in 140 Characters or Less. In *Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second*, pages 1–8. IEEE, 2010.

[16] N. Limsopatham and N. Collier. Adapting Phrase-based Machine Translation to Normalise Medical Terms in Social Media Messages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1675–1680, 2015.

[17] N. Limsopatham and N. Collier. Normalising Medical Concepts in Social Media Texts by Learning Semantic Representation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1014–1023, 2016.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[19] J. Mitchell and M. Lapata. Vector-based Models of Semantic Composition. In *ACL*, pages 236–244, 2008.

[20] K. OConnor, P. Pimpalkhute, A. Nikfarjam, R. Ginn, K. L. Smith, and G. Gonzalez. Pharmacovigilance on Twitter? Mining Tweets for Adverse Drug Reactions. In *AMIA annual symposium proceedings*, volume 2014, page 924. American Medical Informatics Association, 2014.

[21] J. Pennington, R. Socher, and C. D. Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[22] D. Pritsos and E. Stamatatos. The Impact of Noise in Web Genre Identification. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 268–273. Springer, 2015.

[23] S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, and S. Ananiadou. Distributional Semantics Resources for Biomedical Text Processing. *Proceedings of Languages in Biology and Medicine*, 2013.

[24] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers. Statistical Topic Models for Multi-Label Document Classification. *Machine learning*, 88(1-2):157–208, 2012.

[25] S. Sharoff, Z. Wu, and K. Markert. The Web Library of Babel: Evaluating Genre Collections. In *LREC*, 2010.

[26] E. Stamatatos. A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009.

[27] V. Vidulin, M. Lustrek, and M. Gams. Multi-Label Approaches to Web Genre Identification. *JLCL*, 24(1):97–114, 2009.

[28] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang. Dimensional Sentiment Analysis using a Regional CNN-LSTM Model. In *The 54th Annual Meeting of the Association for Computational Linguistics*, volume 225, 2016.