

Comprehend Medical: a Named Entity Recognition and Relationship Extraction Web Service

Parminder Bhatia
Amazon
Seattle, Washington, USA
pamib@amazon.com

Busra Celikkaya
Amazon
Seattle, Washington, USA
busrac@amazon.com

Mohammed Khalilia
Amazon
Seattle, Washington, USA
khalilia@amazon.com

Selvan Senthivel
Amazon
Seattle, Washington, USA
ssenthiv@amazon.com

Abstract—Comprehend Medical is a stateless and Health Insurance Portability and Accountability Act (HIPAA) eligible Named Entity Recognition (NER) and Relationship Extraction (RE) service launched under Amazon Web Services (AWS) trained using state-of-the-art deep learning models. Contrary to many existing open source tools, Comprehend Medical is scalable and does not require steep learning curve, dependencies, pipeline configurations, or installations. Currently, Comprehend Medical performs NER in five medical categories: Anatomy, Medical Condition, Medications, Protected Health Information (PHI) and Treatment, Test and Procedure (TTP). Additionally, the service provides relationship extraction for the detected entities as well as contextual information such as negation and temporality in the form of traits. Comprehend Medical provides two Application Programming Interfaces (API): 1) the NERe API which returns all the extracted named entities, their traits and the relationships between them and 2) the PHId API which returns just the protected health information contained in the text. Furthermore, Comprehend Medical is accessible through AWS Console, Java and Python Software Development Kit (SDK), making it easier for non-developers and developers to use.

Index Terms—Neural Networks, Multi-task Learning, Natural Language Processing, Clinical NLP, Named Entity Recognition, Relationship Extraction

I. INTRODUCTION

Electronic Health Records (EHR) contain a wealth of patients' data ranging from diagnoses, problems, treatments, medications to imaging and clinical narratives such as discharge summaries and progress reports. Structured data are important for billing, quality and outcomes. On the other hand, narrative text is more expressive, more engaging and captures patient's story more accurately. Narrative notes may also contain information about level of concern and uncertainty to others who are reviewing the note. Studies have shown that narrative notes contain more naturalistic prose, more reliable in identifying patients with a given disease and more understandable to healthcare providers reviewing those notes [1]–[5]. Therefore, to have a clear perspective on patient condition, narrative text should be analyzed. However, manual analysis of massive number of narrative text is time consuming, labor intensive and prone to errors.

Many clinical Natural Language Processing (NLP) tools and systems were published to help us make sense of those valuable narrative text. For instance, clinical Text Analysis and Knowledge Extraction System (cTAKES) [6] is an open-

source NLP package based on the Unstructured Information Management Architecture (UIMA) framework [7] and OpenNLP [8] natural language processing toolkit. cTAKES uses a dictionary look-up and each mention is mapped to a Unified Medical Language System (UMLS) concept [9]. MetaMap [10] is another open-source tool aims at mapping mentions in biomedical text to UMLS concepts using dictionary lookup. MetaMap Lite [11] adds negation detection based on either ConText [12] or NegEx [13].

The Clinical Language Annotation, Modeling, and Processing (CLAMP) [14] is one of the most recent clinical NLP systems. CLAMP is motivated by the fact that existing clinical NLP systems need customization and must be tailored to one's task. For NER, CLAMP takes two approaches: machine learning approach using Conditional Random Field (CRF) [15] and dictionary-based, which maps mentions to standardized ontologies. CLAMP also provides assertion and negation detection based on machine learning or rule-based NegEx.

Many of the existing NLP systems rely on ConText [12] and NegEx [13] to detect assertions such as negation. ConText extracts three contextual features for medical conditions: negation, historical or hypothetical and experienced by someone other than the patient. ConText is an extension of NegEx, which is based on regular expression.

Most of the NLP systems discussed above perform linking of mentions to UMLS. They are based on pipelined components that are configurable, rely on dictionary look-up for NER and regular expressions for assertion detection.

Recently, neural network models have been proposed to overcome some of the limitations of rule-based techniques. A feedforward and bidirectional Long Short Term Memory (BiLSTM) networks for generic negation scope detection was proposed in [16]. In [17] a gated recurrent units (GRUs) are used to represent the clinical relations and their context, along with an attention mechanism. Given a text annotated with relations, it classifies the presence and period of the relations. However, this approach is not end-to-end as it does not predict the relations. Additionally, these models generally require large annotated corpus to achieve good performance, but clinical data is scarce.

Kernel-based approaches are also very common, especially in the 2010 i2b2/VA task of predicting assertions. The state-of-the-art in that challenge applied support vector machines

(SVM) to assertion prediction as a separate step after entity extraction [18]. They train classifiers to predict assertions of each concept word, and a separate classifier to predict the assertion of the whole entity. Augmented Bag of Words Kernel (ABoW), which generates features based on NegEx rules along with bag-of-words features was proposed in [19] and a CRF based approach for classification of cues and scope detection was proposed in [20]. These machine learning based approaches often suffer in generalizability.

Once named entities are extracted it is important to identify the relationships between the entities. Several end-to-end models were proposed that jointly learn named entity recognition and relationship extraction [21]–[23]. Generally, relationship extraction models consist of an encoder followed by relationship classification unit [24]–[26]. The encoder provides context aware vector representations for both target entities, which are then merged or concatenated before being passed to the relation classification unit, where a two layered neural network or multi-layered perceptron classifies the pair into different relation types.

Despite the existence of many clinical NLP systems, automatic information extraction from narrative clinical text has not achieved enough traction yet [27]. As reported by [27] there is a significant gap between clinical studies using Electronic Health Record (EHR) data and studies using clinical information extraction. Reasons for such gap can be attributed to limited expertise of NLP experts in the clinical domain, limited availability of clinical data sets due to the HIPAA privacy rules and poor portability and generalizability of clinical NLP systems. Rule-based NLP systems require handcrafted rules, while machine learning-based NLP systems require annotated datasets.

To narrow the clinical NLP adoption gap and to address some of the limitations in existing NLP systems, we present Comprehend Medical, a web service for clinical named entity recognition and relationship extraction. Our contributions are as follows:

- Named entity recognition, relationship extraction and trait detection service encapsulated in one easy to use API.
- Web service that uses deep learning multi-task [28] approach trained on labeled training data and requires no configurations or customization.
- Trait (negation, sign, symptom and diagnosis) detection for medical condition and negation detection for medication.

The rest of the paper is organized as follows: section II presents the methods, section III describes the datasets and experimental settings, section IV contains the results for the NER and RE models, section V talks about the implementation details, section VI gives overview of the supported entities, traits and relationships, section VII presents some of the use cases and we conclude in section VIII.

II. METHODS

In this section we briefly introduce the architectures for named entity recognition and trait detection proposed in [29]

and the relation extraction using explicit context conditioning proposed in [30].

A. Named Entity Recognition Architecture

A sequence tagging problem such as NER can be formulated as maximizing the conditional probability distribution over tags \mathbf{y} given an input sequence \mathbf{x} , and model parameters θ .

$$P(\mathbf{y}|\mathbf{x}, \theta) = \prod_{t=1}^T P(y_t|x_t, y_{1:t-1}, \theta) \quad (1)$$

T is the length of the sequence, and $y_{1:t-1}$ are tags for the previous words. The architecture we use as a foundation is that of [31], [32]. The model consists of three main components: (i) character encoder, (ii) word encoder, and (iii) decoder/tagger.

1) *Encoders*: Given an input sequence $\mathbf{x} \in \mathbb{N}^T$ whose coordinates indicate the words in the input vocabulary, we first encode the character level representation for each word. For each x_t the corresponding sequence $\mathbf{c}^{(t)} \in \mathbb{R}^{L \times e_c}$ of character embeddings is fed into an encoder, where L is the length of a given word and e_c is the size of the character embedding. The character encoder employs two LSTM units which produce $\overrightarrow{h}_{1:t}^{(t)}$ and $\overleftarrow{h}_{1:t}^{(t)}$, the forward and backward hidden representations, respectively, where l is the last timestep in both sequences. We concatenate the last timestep of each of these as the final encoded representation, $h_c^{(t)} = [\overrightarrow{h}_l^{(t)} || \overleftarrow{h}_l^{(t)}]$, of x_t at the character level.

The output of the character encoder is concatenated with a pre-trained word embedding, $m_t = [h_c^{(t)} || \text{emb}_{word}(x_t)]$, which is used as the input to the word level encoder.

Using learned character embeddings alongside word embeddings has shown to be useful for learning word level morphology, as well as mitigating loss of representation for out-of-vocabulary words. Similar to the character encoder we use a BiLSTM to encode the sequence at the word level. The word encoder does not lose resolution, meaning the output at each timestep is the concatenated output of both word LSTMs, $h_t = [\overrightarrow{h}_t || \overleftarrow{h}_t]$.

2) *Decoder and Tagger*: Finally, the concatenated output of the word encoder is used as input to the decoder, along with the label embedding of the previous timestep. During training we use teacher forcing [33] to provide the gold standard label as part of the input.

$$o_t = \text{LSTM}(o_{t-1}, [h_t || \hat{y}_{t-1}]) \quad (2)$$

$$\hat{y}_t = \text{Softmax}(\mathbf{W}o_t + b^s) \quad (3)$$

where $\mathbf{W} \in \mathbb{R}^{d \times n}$, d is the number of hidden units in the decoder LSTM, and n is the number of tags. The model is trained in an end-to-end fashion using a standard cross-entropy objective.

3) *Named Entity Recognition Decoder Model*: Our decoder model provides more context to trait detection by adding an additional input, which is the softmax output from entity extraction. We refer to this architecture as the Conditional Softmax Decoder as shown in Fig. 1 [29]. Thus, the model

learns more about the input as well as the label distribution from entity extraction prediction. As an example, we use negation only for problem entity in the i2b2 dataset. Providing the entity prediction distribution helps the negation model to make better predictions. The negation model learns that if the prediction probability is not inclined towards the problem entity, then it should not predict negation irrespective of the word representation.

$$\hat{y}_t^{Entity}, \text{SoftOut}_t^{Entity} = \text{Softmax}^{Ent}(\mathbf{W}^{Ent} o_t + b^s) \quad (4)$$

$$\hat{y}_t^{Neg} = \text{Softmax}^{Neg}(\mathbf{W}^{Neg}[o_t, \text{SoftOut}_t^{Entity}] + b^s) \quad (5)$$

where, $\text{SoftOut}_t^{Entity}$ is the softmax output of the entity at time step t .

Readers are referred to [29] for more detailed discussion on the conditional softmax decoder model.

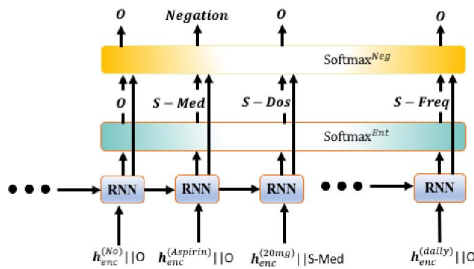


Fig. 1. Conditional softmax decoder model

B. Relationship Extraction Architecture

The extracted entities are not very meaningful by themselves, specially in the healthcare domain. For instance, it is important to know if the procedure was performed bilaterally, on the left or right side. Knowing the correct location will result in more accurate and reliable billing and reimbursement. Hence, it is important to identify the relationships among those clinical entities.

The RE model architecture is described in [30], but we reiterate some of the important details here. Relationships are defined between two entities, which we refer to as head and tail entity. To extract such relationships we proposed relation extraction using explicit context conditioning, where two target entities (head and tail) can be explicitly connected via a context token also known as second order relations. Similar to Bi-affine Relation Attention Networks (BRAN) [24], we first compute the representations for both the head, e_i^{head} , and tail, e_i^{tail} , entities, which are then passed through two multi-layer perceptron (MLP-1) to obtain first-order relation scores, $score^{(1)}(p^{head}, p^{tail})$, as shown in Fig. 2. We also pass e_i^{head} and e_i^{tail} through MLP-2 to obtain second-order relation scores, $score^{(2)}(p^{head}, p^{tail})$, where p^{head} and p^{tail} are the indices for the head and tail entities. The motivation for adding MLP-2 was driven by the need for representations focused on establishing relations with context tokens, as opposed to first-order relations. At the end, the final score for relation between

two entities is given as a weighted sum of first and second order scores.

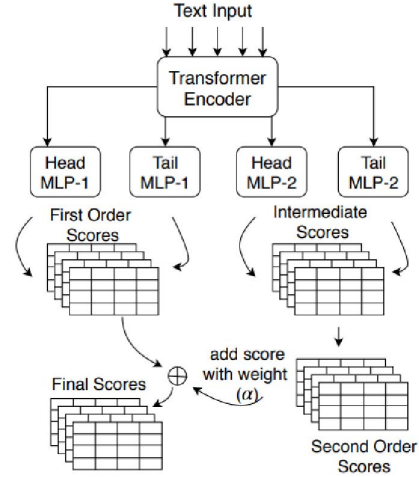


Fig. 2. Relationship extraction model

III. EXPERIMENTS

A. Dataset

We evaluated our model on two datasets. First is the 2010 i2b2/VA challenge dataset for “test, treatment, problem” (TTP) entity extraction and assertion detection, herein referred to as *i2b2*. Unfortunately, only part of this dataset was made public after the challenge, therefore we cannot directly compare with NegEx and ABoW results. We followed the original data split from [34] of 170 notes for training and 256 for testing. The second dataset is proprietary and consists of 4,200 de-identified clinical notes with medical conditions, herein referred to as *DCN*.

The i2b2 dataset contains six predefined relations types including *TrCP* (Treatment Causes Problem), *TrIP* (Treatment Improves Problem), *TrWP* (Treatment Worsens Problem) and one negative relation. The DCN dataset contains seven predefined relationship types such as *with_dosage*, *every* and one negative relation. A summary of the datasets is presented in Table I.

TABLE I
OVERVIEW OF THE I2B2 AND DCN DATASETS.

	i2b2	DCN
Notes	426	4,200
Tokens	416K	1.5M
Entity Tags	13	37
Relations	3,653	270,000
Relation Types	6	7

B. NER Model Settings

Word, character and tag embeddings are 100, 25, and 50 dimensions, respectively. Word embeddings are initialized

using GloVe, while character and tag embeddings are learned. Character and word encoders have 50 and 100 hidden units, respectively, while the decoder LSTM has a hidden size of 50. Dropout is used after every LSTM, as well as for word embedding input. We use Adam as an optimizer. Our model is built using MXNet. Hyperparameters are tuned using Bayesian Optimization [35].

C. RE Model Settings

Our final network had two encoder layers, with 8 attention heads in each multi-head attention sublayer and 256 filters for convolution layers in position-wise feedforward sublayer. We used dropout with probability 0.3 after the embedding layer, head/tail MLPs and the output of each encoder sublayer. We also used a word dropout with probability 0.15 before the embedding layer.

IV. RESULTS

A. NER and Trait Detection Results

We report the results for NER and negation detection for both the i2b2 and DCN datasets in Table II. We observe that our proposed conditional softmax decoder approach outperforms the best model [34] on the i2b2 challenge.

We compare our models for negation detection against NegEx [13] and ABoW [19], which has the best results for the negation detection task on i2b2 dataset. Conditional softmax decoder model outperforms both NegEx and ABoW (Table II). Low performance of NegEx and ABoW is mainly attributed to the fact that they use ontology lookup to index findings and negation regular expression search within a fixed scope. A similar trend was observed in the medication condition dataset (Table II). The important thing to note is the low F1 score for NegEx. This can primarily be attributed to abbreviations and misspellings in clinical notes which can not be handled well by rule-based systems.

TABLE II
TEST SET PERFORMANCE WITH MULTI-TASK I2B2 AND DCN DATASETS

Data	Model	Precision	Recall	F ₁
Named Entity				
i2b2	LSTM:CRF [34]	0.844	0.834	0.839
	Conditional Decoder	0.854	0.858	0.855
DCN	LSTM:CRF [34]	0.82	0.84	0.83
	Conditional Decoder	0.878	0.872	0.874
Negation				
i2b2	Negex [13]	0.896	0.799	0.845
	ABoW Kernel [19]	0.899	0.900	0.900
	Conditional Decoder	0.919	0.891	0.905
DCN	Negex [13]	0.403	0.932	0.563
	Conditional Decoder	0.928	0.874	0.899

We also evaluated the conditional softmax decoder in low resource settings, where we used a sample of our training data. We observed that conditional decoder is more robust in low resource settings than other approaches as we reported in [29].

B. RE Results

To show the benefits of using second-order relations we compared our models performance to BRAN. The two models are different in the weighted addition of second-order relation scores. We tune over this weight parameter on the dev set and observed an improvement in MacroF1 score from 0.712 to 0.734 over DCN data and from 0.395 to 0.407 over i2b2 data. For further comparison a recently published model called Hybrid Deep Learning Approach (HDLA) [36] reported a macroF1 score of 0.388 on the same i2b2 dataset. It should be mentioned that HDLA used syntactic parsers for feature extraction but we do not use any such external tools.

Table III summarizes the performance of our relationship model (+SOR) using second-order relations compared to BRAN and HDLA. We refer the readers to [30] for more detailed analysis of our relationship extraction model.

TABLE III
TEST SET PERFORMANCE OF RELATION EXTRACTION ON I2B2 AND DCN DATASETS

Data	Model	Precision	Recall	F ₁
i2b2	HDLA [36]	0.378	0.422	0.388
	BRAN [24]	0.396	0.403	0.395
	+SOR	0.424	0.419	0.407
DCN	BRAN [24]	0.614	0.85	0.712
	+SOR	0.643	0.879	0.734

V. IMPLEMENTATION

Comprehend Medical APIs run in Amazon's proven, high-availability data centers, with service stack replication configured across three facilities in each AWS region to provide fault tolerance in the event of a server failure or Availability Zone outage. Additionally, Comprehend Medical ensures that system artifacts are encrypted in transit and user data is pass through and will not be stored in any part of the system.

Comprehend Medical is available through a Graphical User interface (GUI) within the AWS console and can be accessed using the Java and Python SDK. Comprehend Medical offers two APIs: 1) the NERe API which returns all the extracted named entities, their traits and the relationships between them, 2) the PHId API which returns just the protected health information contained in the text. Developers can easily integrate Comprehend Medical into their data processing pipelines as shown in Fig. 4.

The only input needed by Comprehend Medical is the text to be analyze. No configuration, customization or other parameters needed, making Comprehend Medical easy to use by anyone who has access to AWS. Comprehend Medical outputs the results in JavaScript Object Notation (JSON), which contains named entities, begin offset, end offset, traits, confidence scores and the relationships between the entities. Using the GUI (Fig. 3) users can quickly visualize their results.

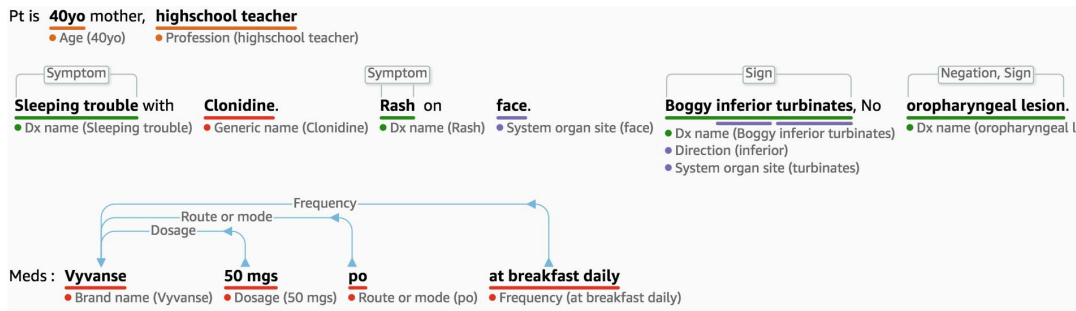


Fig. 3. Rendering of entities, traits and relations by Comprehend Medical UI

VI. ENTITIES, TRAITS AND RELATIONSHIPS

A. Entities

Named entity mentions found in narrative notes are tagged with entity types listed in Table IV. The entities are divided into five categories: Anatomy, Medical Condition, Medication, PHI and TTP. Comprehend Medical is HIPAA eligible and therefore it supports HIPAA identifiers. Some of those identifiers are grouped under one identifier. For instance, Contact Point covers phone and fax numbers, and ID covers social security number, medical record number, account number, certificate or license number and vehicle or device number. An example input text is shown in Fig. 3.

TABLE IV
ENTITIES EXTRACTED BY COMPREHEND MEDICAL

Category	Entity
Anatomy	Direction System Organ Site
Medical Condition	Dx Name Acuity
Medication	Brand Name Generic Name Dosage Duration Frequency Form Route or Mode Strength Rate
PHI	Age Date Name Contact Point Email URL Identifier Address Profession
TTP	Test Name Test Value Test Unit Procedure Name Treatment Name

B. Traits

Comprehend Medical covers four traits, listed in Table V. Negation asserts the presence or absence of a Dx Name

and whether or not the individual is taking the medication. Dx Name has three additional traits: Diagnosis, Sign and Symptom. **Diagnosis** identifies an illness or a disease. **Sign** is an objective evidence of disease and it is a phenomenon that is detected by a physician or a nurse. **Symptom** is a subjective evidence of disease and it is phenomenon that is observed by the individual affected by the disease. An example of traits is shown in Fig. 3.

TABLE V
TRAITS EXTRACTED BY COMPREHEND MEDICAL

Trait	Entity
Negation	Brand/Generic Name, Dx Name
Diagnosis	Dx Name
Sign	Dx Name
Symptom	Dx Name

C. Relationships

A relationship is defined between a pair of entities in the Medication and TTP categories (Table VI). One of the entities in a relationship is the head while the other is the tail entity. In Medication, Generic and Brand Name are the head entity, which can have relationships to tail entities such as Strength and Dosage. An example of relations is shown in Fig. 3.

TABLE VI
RELATIONSHIPS EXTRACTED BY COMPREHEND MEDICAL

Head Entity	Tail Entity
Brand/Generic Name	Dosage Duration Frequency Form Route or Mode Strength
Test Name	Test Value Test Unit

VII. USE CASES

Comprehend Medical reduces the cost, time and effort of processing large amounts of unstructured medical text with high accuracy, making it possible to pursue use cases such

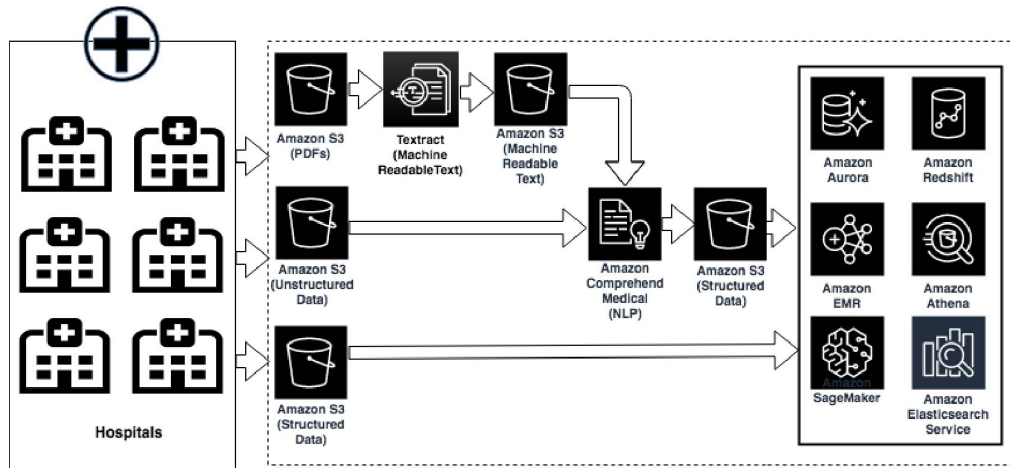


Fig. 4. Integrating Comprehend Medical into data processing pipeline

as clinical trial management, clinical decision support and revenue cycle management.

A. Clinical Trial Management

It can take about 10-15 years for a treatment to be developed from discovery to registration with the Federal Drug Administration (FDA). During that time, research organization can spend six years on clinical trials. Despite the number of year it takes to design those clinical trails, 90% of all clinical trails fail to enroll patients within the targeted time and are forced to extend the enrollment period, 75% of trails fail to enroll the targeted number of patients and 27% fail to enroll any subjects [37].

Life sciences and clinical research organizations can speed up and optimize the process of recruiting patients into a clinical trial as extractions from unstructured text and medical records can expedite the matching process. For instance, indexing patients based on medication, medical condition and treatments can help with quickly identifying the right participants for a lifesaving clinical trial.

Fred Hutchinson Cancer Research Center (FHCRC) utilized Comprehend Medical in their clinical trail management. FHCRC was spending 1.5 hours to annotate a single patient note, about 2.5 hours on manual chart abstraction per patient and per day they can process charts for about three patients. By using Comprehend Medical, FHCRC was able to annotate 9,642 patient notes per hour.

B. Patient and Population Health Analytics

Population health focuses on the discovery of factors and conditions for the a health of a population over time. It aims at identifying patterns of occurrence and knowledge discovery in order to develop polices and actions to improve health of a group or population [38].

Examples of population health analytics include patient stratification, readmission prediction and mortality measurement. Automatically unlocking important information from the

narrative text is invaluable to organizations participating in value-based healthcare and population health. Structured medical records do not fully identify patients with medical history of diabetes, which results in an underestimation of disease prevalence [39]. The inability to identify patient cohorts from structured data represents a problem for the development of population health and clinical management systems. It also negatively affects the accuracy of identifying high-risk and high-cost patients [40]. Ref. [41] identified three areas that may have an impact on readmission, but that are poorly documented in the EMR system, thus the need for NLP-based solutions to extract such information. Also, some symptoms and illness characteristics that are necessary to develop reliable predictors are missing in the coded billing data [42]. Ref. [43] performed mortality prediction and reported a 2% increase in the Area Under the Curve when using features from both structured data and concepts extracted from narrative notes and [44] found that the predictive power of suicide risk factors found in EMR systems become asymptotic, leading them to incorporate analysis on clinical notes to predict risk of suicide.

As seen from the examples above, NLP-based approaches can assist in identifying concepts that are incorrectly codified or are missing in EMR system. Population health platforms can expand their risk analytics to leveraged unstructured clinical data for prediction of high risk patients and epidemiologic studies on outbreaks of diseases.

C. Revenue Cycle Management

In healthcare, Revenue Cycle Management (RCM) is the process of collecting revenue and tracking claims from healthcare providers including hospitals, outpatient clinics, nursing homes, dentist clinics and physician groups [45].

RCM process has been inefficient as most healthcare systems use rule-based approaches and manual audits of documents for billing and coding purposes [46]. Rule-based systems are time consuming, expensive to maintain, require attention and frequent human intervention. Due to these ineff-

fective processes, data coded at point care, which is the source for claims data, can contain errors and inconsistencies.

Coding is the process of encoding the details of patient encounters into standardized terminology [38]. A study by [47] shows that 48 errors found in 38 of the 106 finished consultant episodes in urology and 71% of these errors are caused by inaccurate coding. Ref. [48] measured the consistency of coded data and found that some of these errors were significant enough to change the diagnostic related group.

RCM companies can use Comprehend Medical to enhance existing workflows around computer assisted coding, and validate submitted codes by providers. In addition, claim audits, which often requires finding text evidence for submitted claims and is done manually, could be done more accurately and faster.

D. Pharmacovigilance

The aim of pharmacovigilance is to monitor, detect and prevent adverse drug events (ADE) of medical drugs. Early system used for pharmacovigilance is the spontaneous reporting system (SRS), which provided safety information on drugs [49]. However, SRS databases are incomplete, inaccurate and contain biased reporting [49], [50]. A newer generation of databases was created that contains clinical information for large patient population, such as the Intensive Medicines Monitoring Program (IMMP) and the General Practice Research Database (GPRD). Such databases included data from structured fields and forms, but very small amount of details are stored in the structured fields. Researchers then started to look into EHR data for pharmacovigilance. However, most valuable information in patient records are contained in the unstructured text.

Using NLP to extract information from narrative text have shown improvement in ADE detection and pharmacovigilance [51]. Ref. [50], [52] also reported that ADEs are underreported in EHR systems and they used NLP techniques to enhance ADE detection.

VIII. CONCLUSION

Studies have shown that narrative notes are more expressive, more engaging and captures patient's story more accurately compared to the structured EHR data. They also contain more naturalistic prose, more reliable in identifying patients with a given disease and more understandable to healthcare providers reviewing those notes, which urges the need for a more accurate, intuitive and easy to use NLP system. In this paper we presented Comprehend Medical, a HIPAA eligible Amazon Web Service for medical language entity recognition and relationship extraction. Comprehend Medical supports several entity types divided into five different categories (Anatomy, Medical Condition, Medication, Protected Health Information, Treatment, Test and Procedure) and four traits (Negation, Diagnosis, Sign, Symptom). Comprehend Medical uses state-of-the-art deep learning models and provides two APIs, the NERE and PHId API. Comprehend Medical also comes with four different interfaces (CLI, Java SDK, Python SDK and GUI)

and contrary to many other existing clinical NLP systems, it does not require dependencies, configuration or pipelined components customization.

REFERENCES

- [1] S. T. Rosenbloom, J. C. Denny, H. Xu, N. Lorenzi, W. W. Stead, and K. B. Johnson, "Data from clinical notes: a perspective on the tension between structure and flexible documentation," *Journal of the American Medical Informatics Association*, vol. 18, no. 2, pp. 181–186, mar 2011.
- [2] K. M. Fox, M. Reuland, W. G. Hawkes, J. R. Hebel, J. Hudson, S. I. Zimmerman, J. Kenzora, and J. Magaziner, "Accuracy of medical records in hip fracture." *Journal of the American Geriatrics Society*, vol. 46, no. 6, pp. 745–50, jun 1998.
- [3] K. A. Marill, E. S. Gauharou, B. K. Nelson, M. A. Peterson, R. L. Curtis, and M. R. Gonzalez, "Prospective, randomized trial of template-assisted versus undirected written recording of physician records in the emergency department." *Annals of emergency medicine*, vol. 33, no. 5, pp. 500–9, may 1999.
- [4] A. M. van Ginneken, "The physician's flexible narrative." *Methods of information in medicine*, vol. 35, no. 2, pp. 98–100, jun 1996.
- [5] A. J. Cawsey, B. L. Webber, and R. B. Jones, "Natural language generation in health care." *Journal of the American Medical Informatics Association : JAMIA*, vol. 4, no. 6, pp. 473–82, 1997.
- [6] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute, "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications." *Journal of the American Medical Informatics Association : JAMIA*, vol. 17, no. 5, pp. 507–13, jan 2010.
- [7] D. Ferrucci and A. Lally, "UIMA: an architectural approach to unstructured information processing in the corporate research environment." *Natural Language Engineering*, vol. 10, no. 3-4, pp. 327–348, sep 2004.
- [8] J. Baldrige, "The Apache OpenNLP Project," URL: <https://opennlp.apache.org/>, 2005.
- [9] O. Bodenreider, "The Unified Medical Language System (UMLS): integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, no. 90001, pp. 267D–270, jan 2004.
- [10] A. R. Aronson and F.-M. Lang, "An overview of MetaMap: historical perspective and recent advances." *Journal of the American Medical Informatics Association : JAMIA*, vol. 17, no. 3, pp. 229–36, jan 2010.
- [11] D. Demner-Fushman, W. J. Rogers, and A. R. Aronson, "MetaMap Lite: an evaluation of a new Java implementation of MetaMap," *Journal of the American Medical Informatics Association*, vol. 24, no. 4, p. ocw177, jan 2017.
- [12] H. Harkema, J. N. Dowling, and T. Thornblade, "ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports," *Journal of Biomedical Informatics*, vol. 42, no. 5, pp. 839–851, oct 2009.
- [13] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan, "A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries," *Journal of Biomedical Informatics*, vol. 34, no. 5, pp. 301–310, oct 2001.
- [14] E. Soysal, J. Wang, M. Jiang, Y. Wu, S. Pakhomov, H. Liu, and H. Xu, "CLAMP a toolkit for efficiently building customized clinical natural language processing pipelines," *Journal of the American Medical Informatics Association*, vol. 25, no. 3, pp. 331–336, mar 2018.
- [15] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proceedings of the 18th International Conference on Machine Learning*, vol. 951. Citeseer, 2001, pp. 282–289.
- [16] F. Fancellu, A. Lopez, and B. Webber, "Neural networks for negation scope detection," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 495–504.
- [17] L. Rumeng, N. Jagannatha Abhyuday, and Y. Hong, "A hybrid neural network model for joint prediction of presence and period assertions of medical events in clinical notes," in *AMIA Annual Symposium Proceedings*, vol. 2017. American Medical Informatics Association, 2017, p. 1149.
- [18] B. de Bruijn, C. Cherry, S. Kiritchenko, J. Martin, and X. Zhu, "Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 557–562, 2011.

- [19] C. Shivade, M.-C. de Marneffe, E. Fosler-Lussier, and A. M. Lai, "Extending negex with kernel methods for negation detection in clinical text," in *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, 2015, pp. 41–46.
- [20] K. Cheng, T. Baldwin, and K. Verspoor, "Automatic negation and speculation detection in veterinary clinical text," in *Proceedings of the Australasian Language Technology Association Workshop 2017*, 2017, pp. 70–78.
- [21] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 1105–1116.
- [22] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1227–1236.
- [23] H. Adel and H. Schütze, "Global normalization of convolutional neural networks for joint entity and relation classification," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 1723–1729.
- [24] P. Verga, E. Strubell, and A. McCallum, "Simultaneously self-attending to all mentions for full-abstract biological relation extraction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 872–884.
- [25] F. Christopoulou, M. Miwa, and S. Ananiadou, "A walk-based model on entity graphs for relation extraction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 81–88.
- [26] Y. Su, H. Liu, S. Yavuz, I. Gur, H. Sun, and X. Yan, "Global relation embedding for relation extraction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 820–830.
- [27] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, and H. Liu, "Clinical information extraction applications: A literature review," *Journal of Biomedical Informatics*, vol. 77, pp. 34–49, jan 2018.
- [28] P. Bhatia, K. Arumae, and E. B. Celikkaya, "Dynamic transfer learning for named entity recognition," in *International Workshop on Health Intelligence*. Springer, 2019, pp. 69–81.
- [29] P. Bhatia, B. Celikkaya, and M. Khalilia, "Joint Entity Extraction and Assertion Detection for Clinical Text," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 954–959.
- [30] G. Singh and P. Bhatia, "Relation Extraction using Explicit Context Conditioning," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2019, pp. 1442–1447.
- [31] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of NAACL-HLT*, 2016, pp. 260–270.
- [32] Z. Yang, R. Salakhutdinov, and W. Cohen, "Multi-task cross-lingual sequence tagging from scratch," *arXiv preprint arXiv:1603.06270*, 2016.
- [33] R. J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, jun 1989.
- [34] R. Chalapathy, E. Z. Borzeshi, and M. Piccardi, "Bidirectional LSTM-CRF for Clinical Concept Extraction," in *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, Osaka, Japan, 2016, pp. 7–12.
- [35] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.
- [36] V. R. Chikka and K. Karlapalem, "A hybrid deep learning approach for medical relation extraction," *arXiv preprint arXiv:1806.11189*, 2018.
- [37] R. B. Giffin, Y. Lebovitz, R. A. English, and Others, *Transforming clinical research in the United States: challenges and opportunities: workshop summary*. National Academies Press, 2010.
- [38] K. Giannangelo and S. Fenton, "EHR's effect on the revenue cycle management Coding function," *Journal of healthcare information management : JHIM*, vol. 22, no. 1, pp. 26–30, 2008.
- [39] L. Zheng, Y. Wang, S. Hao, A. Y. Shin, B. Jin, A. D. Ngo, M. S. Jackson-Browne, D. J. Feller, T. Fu, K. Zhang, X. Zhou, C. Zhu, D. Dai, Y. Yu, G. Zheng, Y.-M. Li, D. B. McElhinney, D. S. Culver, S. T. Alfreds, F. Stearns, K. G. Sylvester, E. Widen, and X. B. Ling, "Web-based Real-Time Case Finding for the Population Health Management of Patients With Diabetes Mellitus: A Prospective Validation of the Natural Language Processing-Based Algorithm With Statewide Electronic Medical Records," *JMIR medical informatics*, vol. 4, no. 4, p. e37, nov 2016.
- [40] D. W. Bates, S. Saria, L. Ohno-Machado, A. Shah, and G. Escobar, "Big Data In Health Care: Using Analytics To Identify And Manage High-Risk And High-Cost Patients," *Health Affairs*, vol. 33, no. 7, pp. 1123–1131, jul 2014.
- [41] J. L. Greenwald, P. R. Cronin, V. Carballo, G. Danaei, and G. Choy, "A Novel Model for Predicting Rehospitalization Risk Incorporating Physical Function, Cognitive Status, and Psychosocial Support Using Natural Language Processing," *Medical Care*, vol. 55, no. 3, pp. 261–266, mar 2017.
- [42] A. Rumshisky, M. Ghassemi, T. Naumann, P. Szolovits, V. M. Castro, T. H. McCoy, and R. H. Perlis, "Predicting early psychiatric readmission with natural language processing of narrative discharge summaries," *Translational Psychiatry*, vol. 6, no. 10, pp. e921–e921, oct 2016.
- [43] M. Jin, M. T. Bahadori, A. Colak, P. Bhatia, B. Celikkaya, R. Bhakta, S. Senthivel, M. Khalilia, D. Navarro, B. Zhang, T. Doman, A. Ravi, M. Liger, and T. Kass-hout, "Improving Hospital Mortality Prediction with Medical Named Entities and Multimodal Learning," *Neural Information Processing Systems workshop on Machine Learning for Health*, 2018.
- [44] C. Poulin, B. Shiner, P. Thompson, L. Vepstas, Y. Young-Xu, B. Goertzel, B. Watts, L. Flashman, and T. McAllister, "Predicting the Risk of Suicide by Analyzing the Text of Clinical Notes," *PLoS ONE*, vol. 9, no. 1, p. e85733, jan 2014.
- [45] V. Mindel and L. Mathiassen, "Contextualist inquiry into IT-enabled hospital revenue cycle management: bridging research and practice," *Journal of the Association for Information Systems*, vol. 16, no. 12, p. 1016, 2015.
- [46] P. Schouten, "Big data in health care: solving provider revenue leakage with advanced analytics," *Healthcare Financial Management*, vol. 67, no. 2, pp. 40–43, feb 2013.
- [47] A. Ballaro, S. Oliver, and M. Emberton, "Do we do what they say we do? Coding errors in urology," *BJU International*, vol. 85, no. 4, pp. 389–391, mar 2000.
- [48] D. P. Lorence and I. A. Ibrahim, "Benchmarking variation in coding accuracy across the United States," *Journal of health care finance*, vol. 29, no. 4, pp. 29–42, 2003.
- [49] X. Wang, G. Hripcsak, M. Markatou, and C. Friedman, "Active Computerized Pharmacovigilance Using Natural Language Processing, Statistics, and Electronic Health Records: A Feasibility Study," *Journal of the American Medical Informatics Association*, vol. 16, no. 3, pp. 328–337, may 2009.
- [50] A. Henriksson, M. Kvist, H. Dalianis, and M. Duneld, "Identifying adverse drug event information in clinical notes with distributional semantic representations of context," *Journal of Biomedical Informatics*, vol. 57, pp. 333–349, oct 2015.
- [51] Y. Luo, W. K. Thompson, T. M. Herr, Z. Zeng, M. A. Berendsen, S. R. Jonnalagadda, M. B. Carson, and J. Starren, "Natural Language Processing for EHR-Based Pharmacovigilance: A Structured Review," *Drug Safety*, vol. 40, no. 11, pp. 1075–1089, nov 2017.
- [52] N. Shang, H. Xu, T. C. Rindfleisch, and T. Cohen, "Identifying plausible adverse drug reactions using knowledge extracted from the literature," *Journal of Biomedical Informatics*, vol. 52, pp. 293–310, dec 2014.