Clinical Concept Extraction for Document-Level Coding

Sarah Wiegreffe[1], **Edward Choi**[1*], **Sherry Yan**[2], **Jimeng Sun**[1], **Jacob Eisenstein**[1]
[1]Georgia Institute of Technology
[2]Sutter Health
[*] Current Affiliation: Google Inc
saw@gatech.edu, edwardchoi@google.com, yansx@sutterhealth.org,
jsun@cc.gatech.edu, jacobe@gatech.edu

Abstract

The text of clinical notes can be a valuable source of patient information and clinical assessments. Historically, the primary approach for exploiting clinical notes has been information extraction: linking spans of text to concepts in a detailed domain ontology. However, recent work has demonstrated the potential of supervised machine learning to extract document-level codes directly from the raw text of clinical notes. We propose to bridge the gap between the two approaches with two novel syntheses: (1) treating extracted concepts as *features*, which are used to supplement or replace the text of the note; (2) treating extracted concepts as *labels*, which are used to learn a better representation of the text. Unfortunately, the resulting concepts do not yield performance gains on the document-level clinical coding task. We explore possible explanations and future research directions.

1 Introduction

Clinical decision support from raw-text notes taken by clinicians about patients has proven to be a valuable alternative to state-of-the-art models built from structured EHRs. Clinical notes contain valuable information that the structured part of the EHR does not provide, and do not rely on expensive and time-consuming human annotation (Torres et al., 2017; American Academy of Professional Coders, 2019). Impressive advances using deep learning have allowed for modeling on the raw text alone (Mullenbach et al., 2018; Rios and Kavuluru, 2018a; Baumel et al., 2018). However, there exist some shortcomings to these approaches: clinical text is noisy, and often contains heavy amounts of abbreviations and acronyms, a challenge for machine reading (Nguyen and Patrick, 2016). Additionally, rare words replaced with "UNK" tokens for better generalization may be crucial for predicting rare labels.

Clinical concept extraction tools abstract over the noise inherent in surface representations of clinical text by linking raw text to standardized concepts in clinical ontologies. The Apache clinical Text Analysis Knowledge Extraction System (cTAKES, Savova et al., 2010) is the most widely-used such tool, with over 1000 citations. Based on rules and non-neural machine learning methods and engineered for almost a decade, cTAKES provides an easily-obtainable source of human-encoded domain knowledge, although it cannot leverage deep learning to make document-level predictions.

Our goal in this paper is to maximize the predictive power of clinical notes by bridging the gap between information extraction and deep learning models. We address the following research questions: how can we best leverage tools such as cTAKES on clinical text? Can we show the value of these tools in linking unstructured data to structured codes in an existing ontology for downstream prediction?

We explore two novel hybrids of these methods: data augmentation (augmenting text with extracted concepts) and multi-task learning (learning to predict the output of cTAKES). Unfortunately, in neither case does cTAKES improve downstream performance on the document-level clinical coding task. We probe this negative result through an extensive series of ablations, and suggest possible explanations, such as the lack of word variation captured through concept assignment.

2 Related Work

Clinical Ontologies Clinical concept ontologies facilitate the maintenance of EHR systems with standardized and comprehensive code sets, allowing consistency across healthcare institutions and practitioners. The Unified Medical Language System (UMLS) (Lindberg et al., 1993) maintains

Proceedings of the BioNLP 2019 workshop, pages 261–272
Florence, Italy, August 1, 2019.

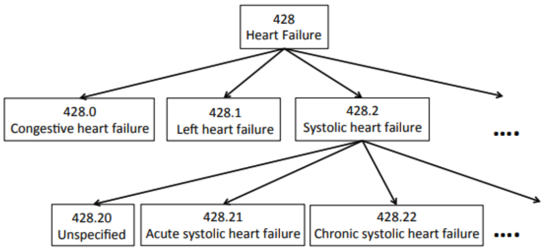


Figure 1: A subtree of the ICD ontology (figure from Singh et al., 2014).

a standardized vocabulary of clinical concepts, each of which is assigned a concept unique identifier (CUI). The Systematized Nomenclature of Medicine- Clinical Terms (SNOMED-CT) (Donnelly, 2006) and the International Classification of Diseases (ICD) (National Center for Health Statistics, 1991) build off of the UMLS and provide structure by linking concepts based on their relationships. The SNOMED ontology has over 340,000 active concepts, ranging from fine-grained ("Adenylosuccinate lyase deficiency") to extremely general ("patient"). The ICD ontology is narrower in scope, with around 13,000 diagnosis and procedure codes used for insurance billing. Unlike SNOMED, which has an unconstrained graph structure, ICD9 is organized into a top-down hierarchy of specificity (see Figure 1).

Clinical Information Extraction Tools There are several tools for extracting structured information from clinical text. Popular types of information extraction include *named-entity recognition*, identifying words or phrases in the text which align with clinical concepts, and *ontology mapping*, labelling the identified words and phrases with their respective clinical codes from an existing ontology.[1] Of the tools which perform both of these tasks, the open-source Apache cTAKES is used in over 50% of recent work (Wang et al., 2017), outpacing competitors such as MetaMap (Aronson, 2001) and MedLEE (Friedman, 2000).

cTAKES utilizes a rule-based system for performing ontology mapping, via a UMLS dictionary lookup on the noun phrases inferred by a part-of-speech tagger. Taking raw text as input, the software outputs a set of UMLS concepts identified in the text and their positions, with functionality to map them to other ontologies such as SNOMED and ICD9. It is highly scalable, and can be deployed locally to avoid compromising identifiable patient data. Figure 2 shows an example cTAKES annotation on a clinical record.

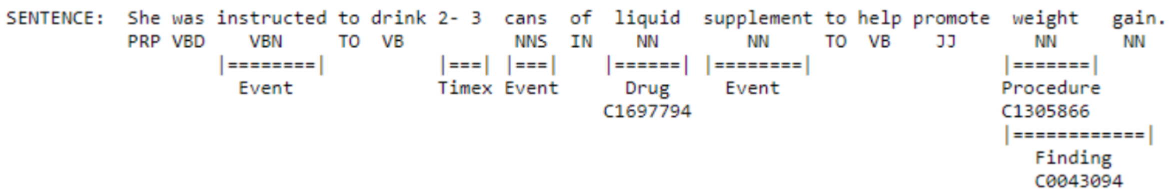
Clinical Named-Entity Recognition (NER) Recent work has focused on developing tools to replace cTAKES in favor of modern neural architectures such as Bi-LSTM CRFs (Boag et al., 2018; Tao et al., 2018; Xu et al., 2018; Greenberg et al., 2018), varying in task definition and evaluation. Newer approaches leverage contextualized word embeddings such as ELMo (Zhu et al., 2018; Si et al., 2019). In contrast, we focus on maximizing the power of existing tools such as cTAKES. This approach is more practical in the near-term, because the adoption of new NER systems in the clinical domain is inhibited by the amount of computational power, data, and gold-label annotations needed to build and train such token-level models, as well as considerations for the effectiveness of domain transfer and a necessity to perform annotations locally in order to protect patient data. Newer models do not provide these capabilities.

NER in Text-based Models Prior works use the output of cTAKES as features for disease- and drug-specific tasks, but either concatenate them as shallow features, or substitute them for the text itself (see Wang et al. (2017) for a literature review). Weng et al. (2017) incorporate the output of cTAKES into their input feature vectors for the task of predicting the medical subdomain of clinical notes. However, they use them as shallow features in a non-neural setting, and combine cTAKES annotations with the text representations by concatenating the two into one larger feature vector. In contrast, we propose to learn dense neural concept embedding representations, and integrate the concepts in a learnable fashion to guide the representation learning process, rather than simply concatenating them or using them as a text replacement. We additionally focus on a more challenging task setting.

Boag and Kané (2017) augment a Word2Vec training objective to predict clinical concepts. This work is orthogonal to ours as it is an unsupervised "embedding pretraining" approach rather than an end-to-end supervised model.

[1] Ontology mapping also serves as a form of text normalization.

[2] Figure from https://healthnlp.github.io/examples/.



```
SENTENCE:  She was instructed to drink 2- 3  cans  of  liquid  supplement to help promote  weight   gain.
           PRP VBD    VBN     TO  VB          NNS  IN    NN        NN      TO  VB    JJ       NN      NN
               |========|            |===| |===|  |======| |========|               |=======|
                  Event              Timex Event    Drug     Event                  Procedure
                                                  C1697794                          C1305866
                                                                                    |============|
                                                                                       Finding
                                                                                       C0043094
```

Figure 2: An example of cTAKES annotation output with part-of-speech tags and UMLS CUIs for named entities.[2]

Automated Clinical Coding The automated clinical coding task is to predict from the raw text of a hospital discharge summary describing a patient encounter all of the possible ICD9 (diagnosis and procedure) codes which a human annotator would assign to the visit. Because these annotators are trained professionals, the ICD codes assigned serve as a natural label set for describing a patient record, and the task can be seen as a proxy for a general patient outcome or treatment prediction task. State-of-the-art methods such as CAML (Mullenbach et al., 2018) treat each label prediction as a separate task, performing many binary classifications over the many-thousand-dimensional label space. The model is described in more detail in the next section.

The label space is very large (tens of thousands of possible codes) and frequency is long-tailed. Rios and Kavuluru (2018b) find that CAML performs weakly on rare labels.

3 Problem Setup

Task Notation A given discharge summary is represented as a matrix $\boldsymbol{X} \in \mathbb{R}^{d_e \times N}$.[3] The set of diagnosis and procedure codes assigned to the visit is represented as the one-hot vector $\boldsymbol{y} \in \{0,1\}^L$. The task can be framed as $L = |\mathcal{L}|$ binary classifications: predict $y_l \in \{0,1\}$ for code l in labelspace $\mathcal{L}$.

Data We use the publically-available MIMIC-III dataset, a collection of deidentified discharge summaries describing patient stays in the Beth Israel Deaconess Medical Center ICU between 2001 and 2012 (Johnson et al., 2016; Pollard and Johnson, 2016). Each discharge summary has been tagged with a set of ICD9 codes. See Figure 3 for an example of a record, and Appendix A for a description of the dataset and preprocessing.

Concept Annotation We run cTAKES on the discharge summaries (described in Appendix B). Results on the extracted concepts are presented in Table 1. Note the difference in number of annotations provided by using the SNOMED ontology compared to ICD9.[4]

ICD9	
Total concepts extracted	1,005,756
Mean # extracted concepts per document	19.10
Mean % words assigned a concept per document	1.26%
SNOMED	
Total concepts extracted	28,090,075
Mean # extracted concepts per document	532.76
Mean % words assigned a concept per document	35.21%
Mean # tokens per document	1513.00

Table 1: Descriptive Statistics on concept extraction for the MIMIC-III corpus.

Base model We evaluate against CAML (Mullenbach et al., 2018), a state-of-the-art text-based model for the clinical coding task. The model leverages a convolutional neural network (CNN) with per-label attention to predict the combination of codes to assign to a diven discharge summary. Applying convolution over $\boldsymbol{X}$ results in a convolved input representation $\boldsymbol{H} \in \mathbb{R}^{d_c \times N}$ (with $d_c < d_e$) in which the column-dimensionality N is preserved. $\boldsymbol{H}$ is then used to predict $\boldsymbol{y}$, by attentional pooling over the columns.

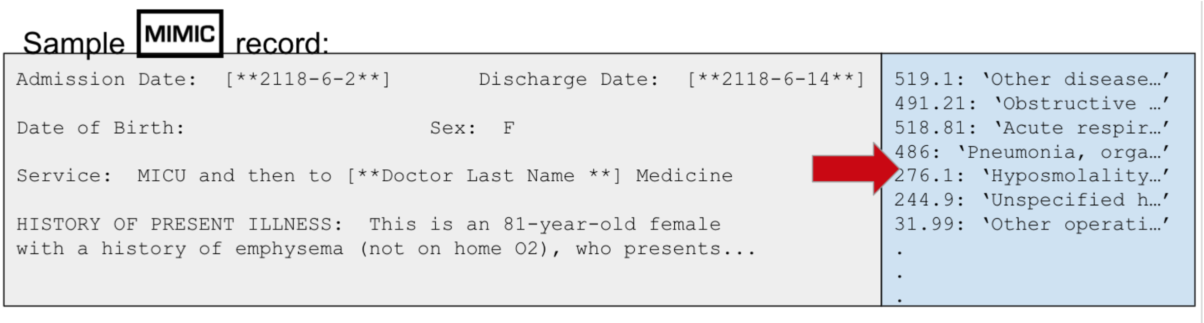
We include implementation details of all methods, including hyperparameters and training, in Appendix A.

4 Approach 1: Augmentation Model

One limitation of learning-based models is their tendency to lose uncommon words to "UNK" tokens, or to suffer from poor representation learning for them. We hypothesize that rare words are important for predicting rare labels, and that text-based

[3] We use notation for a single instance throughout.

[4] Preliminary experiments with sparser ontologies (RXNORM) were not promising, leading us to choose these two ontologies based on their annotation richness (SNOMED) and direct relation to the prediction task (ICD9).



Sample MIMIC record:

```
Admission Date:  [**2118-6-2**]       Discharge Date:  [**2118-6-14**]

Date of Birth:                    Sex:  F

Service:  MICU and then to [**Doctor Last Name **] Medicine

HISTORY OF PRESENT ILLNESS:  This is an 81-year-old female
with a history of emphysema (not on home O2), who presents...
```

```
519.1:  'Other disease…'
491.21:  'Obstructive …'
518.81:  'Acute respir…'
486:  'Pneumonia, orga…'
276.1:  'Hyposmolality…'
244.9:  'Unspecified h…'
31.99:  'Other operati…'
.
.
.
```

Figure 3: An example clinical discharge summary and associated ICD codes.

models may be improved by augmenting word embeddings with concept embeddings as a means to strengthen representations of rare or unseen words. We additionally hypothesize that linking multiple words to a shared concept via cTAKES annotation will reduce textual noise by grouping word variants to a shared representation in a smaller and more frequently updated parameter space.

4.1 Method

Given a discharge summary containing words $w_1, w_2, ..., w_N \in \mathcal{W}^*$ and an embedding function $\gamma : \mathcal{W} \rightarrow \mathbb{R}^{d_e}$, we construct input matrix $\boldsymbol{X} = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, ..., \boldsymbol{x}_N^T] \in \mathbb{R}^{d_e \times N}$ as column-stacked word embeddings, where $\boldsymbol{x}_n = \gamma(w_n)$.

We additionally assume a code embedding function $\phi : \mathcal{C} \rightarrow \mathbb{R}^{d_e}$ and a set of annotated codes for a given document $c_1, c_2, \ldots, c_N \in \mathcal{C}^*$, where $\mathcal{C}$ is the full codeset for the ontology used to annotate the document, and c_n is the code annotated for word token w_n, if one exists (else $c_n = \varnothing$, by abuse of notation). We construct a representation for each document, $\boldsymbol{D}$, of the same dimensionality as $\boldsymbol{X}$, by learning one representation leveraging both the concept and word embedding at each position:

For token n,

$$\boldsymbol{d}_n = \beta_{w_n,c_n} \boldsymbol{\phi}(c_n) + (1 - \beta_{w_n,c_n}) \boldsymbol{x}_n, \quad (1)$$

$\beta_{w_n,c_n} \in [0, 1]$ is a learned parameter specific to each observed word+concept pair, including UNK tokens.[5] Intuitively, if there is a concept associated with index n, a concept embedding $\boldsymbol{\phi}(c_n)$ is generated and a *linear combination* of the word and concept embedding is learned, using a learned parameter specific to that word+concept pair.[6] We fix $\beta_{w_n,c_n=\varnothing} = 0$, which reverts to the word embedding when there is no concept assigned.

We additionally propose a simpler version of this method, *full replace*, in which word embeddings are completely replaced with concept embeddings if they exist (i.e. $\beta_{w_n,c_n} = 1, \forall w_n, c_n \neq \varnothing$). In this formulation, if a concept spans multiple words, all of those words are represented by the same vector. Conversely, the CAML baseline corresponds to a model in which $\beta_{w_n,c_n} = 0, \forall w_n, c_n$.

4.2 Evaluation Setup

Metrics In addition to the metrics reported in prior work, we report average precision score (AP), which is a preferred metric to AUC for imbalanced classes (Saito and Rehmsmeier, 2015; Davis and Goadrich, 2006). We report both macro- and micro- metrics, with the former being more favorable toward rare labels by weighting all classes equally. We additionally focus on the precision-at-k (P@k) metric, representing the fraction of the k highest-scored predicted labels that are present in the ground truth. Both macro-metrics and P@k are useful in a computer-assisted coding use-case, where the desired outcome is to correctly identify needle-in-the-haystack labels as opposed to more frequent ones, and to accurately suggest a small subset of codes with the highest confidence as annotation suggestions (Mullenbach et al., 2018).

Baselines Along with CAML, we evaluate on a *raw codes* baseline where the ICD9 annotations generated by cTAKES $c_1, c_2, \ldots, c_N$ are used directly as the document-level predictions. Formally,

[5] We experimented with models in which this gate was computed element-wise and shared by all word+concept pairs (e.g. by passing $\boldsymbol{x}_n$ and $\phi(c_n)$ through a linear layer or simple multi-layer perceptron to compute $\boldsymbol{d}_n$), but this did not improve performance.

[6] A single token may have multiple concept annotations associated with it. We experiment with an attention mechanism for this case (see Appendix C), but find a heuristic of arbitrarily selecting the first concept assigned to each word performs just as well.

$\hat{y}_{c_n} = 1$ when $c_n \in \mathcal{L}$ and $c_n \neq \varnothing$, for all n in integers 1 to N.

4.3 Results

We present results on the test set in Table 2. Overall, the concept-augmented models are indistinguishable from the baseline, and there is no significant difference between annotation type or recombination method, although the linear combination method with ICD9 annotations is the best performing and rivals the baseline.

Following the negative results for our initial attempt to augment word embeddings with concept embeddings, we tried two alternative strategies:

- We concatenated the ICD9 annotations with two other ontologies: RXNORM and SNOMED. While this led to greater coverage over the text (with slightly more than one third of the tokens in the text receiving corresponding concept annotations), it did not improve downstream performance.
- Prior work has demonstrated that leveraging clinical ontological structure can allow models to learn more effective code embeddings in fully structured data models (Singh et al., 2014; Choi et al., 2017). We applied the methodology of Choi et al. (2017) on both the ICD9 and SNOMED annotations, but this did not improve performance. For more details, see Appendix D.

4.4 Error Analysis

Error analysis of the word-to-concept mapping produced by cTAKES exposes limitations of our initial hypothesis that cTAKES mitigates word-level variation by assigning multiple distinct word phrases to shared concepts. Figure 4 demonstrates that the vast majority of the ICD9 concepts in the corpus are assigned to only one distinct word phrase, and the same results are observed for SNOMED concepts. This may explain the virtually indistinguishable performance of the augmentation models from the baseline, because randomly-initialized word and concept embeddings which are observed in strictly identical contexts should theoretically converge to the same representation.[7]

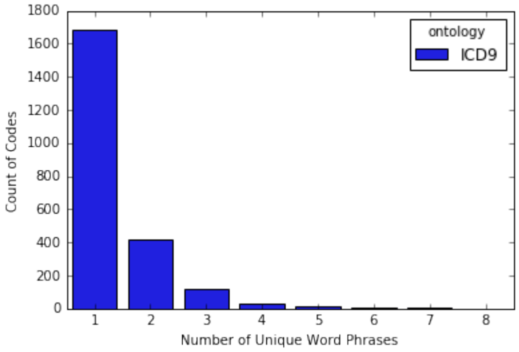


Figure 4: A histogram showing the distribution of ICD9 concepts in $\mathcal{C}$ grouped according to the number of unique word phrases in the MIMIC-III corpus associated with each. We observe the same trend when plotting SNOMED annotations.

The raw codes baseline performs poorly, which aligns with the observation that cTAKES codes assigned to a discharge summary often do not have appropriate or proportional levels of specificity (for example, the top-level ICD9 code '428 Heart Failure' may be assigned by cTAKES, but the gold-label code is '428.21 Acute Systolic Heart Failure'). This may also contribute to the negative result of the proposed model.

Figure 6 (included in the Appendix) illustrates prediction performance as a function of code frequency in the training set, showing that the proposed model does not improve upon the baseline for rare or semi-rare codes.[8]

4.5 Ablations

We separate and analyze the two distinct components of cTAKES' annotation ability for further analysis: 1) how well cTAKES recognizes the location of concepts in the text (*NER*), and 2) how accurately cTAKES maps the recognized positions to the correct clinical concepts (*ontology mapping*). Annotation sparsity (NER) and/or cTAKES mapping error may lend the raw text on its own equally useful, as observed in Table 2. We investigate these hypotheses here. We evaluate performance of ablations relative to the augmentation model and baseline to determine whether each component individ-

*These metrics were computed by randomly selecting k elements from those predicted, since there are no sorted probabilities associated with this baseline. For the same reasons we cannot report AUC or AP metrics.

[7] Simulations of the augmentation method under a contrived setting with more concept annotations per note as well as more unique word phrases mapping to a single concept demonstrate solid performance increases over the baseline. This provides supporting evidence that the findings presented in this section may be the cause of the negative result rather than our proposed architecture.

[8] We use the following grouping criteria: rare codes have 50 or fewer occurrences in the training data, semi-rare have between 50 and 1000, and common have more than 1000.

	AUC		**AP**		**F1**		**R@k**		**P@k**	
Model	Macro	Micro	Macro	Micro	Macro	Micro	8	15	8	15
Baseline (Mullenbach et al., 2018)	0.8892	0.9846	**0.2492**	0.5426	**0.0796**	**0.5421**	**0.3731**	**0.5251**	0.7120	0.5616
Baseline (raw codes)	n/a*	n/a*	n/a*	n/a*	0.0189	0.0877	0.0534*	0.0640*	0.1132*	0.0747*
Augmentation with ICD9										
full replace	0.8846	0.9838	0.2242	0.5329	0.0691	0.5363	0.3688	0.5189	0.7048	0.5564
linear combination	**0.8914**	**0.9849**	0.2467	**0.5427**	0.0763	0.5419	**0.3732**	**0.5267**	**0.7121**	**0.5634**
Augmentation with SNOMED										
full replace	0.8744	0.9830	0.2221	0.5271	0.0724	0.5326	0.3675	0.5177	0.7022	0.5547
linear combination	0.8781	0.9835	0.2238	0.533	0.0692	0.5357	0.3687	0.5194	0.7042	0.5563

Table 2: Test set results using the augmentation methods.

ually adds value. The ablations are:

1. *Dummy Concepts* We replace all word embeddings annotated by cTAKES with 0-vectors, and only use remaining embeddings for prediction. If this alternative shows similar performance to the baseline, then we conclude that the positions in the text annotated by cTAKES (NER) are not valuable for prediction performance.

2. *Concepts Only* We test the complement by replacing all word embeddings *not* annotated by cTAKES with a 0-vector. In contrast to Dummy Concepts, strong performance of this approach relative to the baseline will allow us to conclude that the positions in the text annotated by cTAKES are valuable for prediction performance.

3. *Concepts Only, Concept Embeddings* We replace all word embeddings not annotated by cTAKES with a 0-vector, and then replace all remaining word embeddings with their concept embedding. If this model performs better than Concepts Only, it will demonstrate the strength of cTAKES' ontology mapping component.

Note that Dummy Concepts and Concepts Only are the decomposition of the baseline CAML. Similarly, Dummy Concepts and Concepts Only, Concept Embeddings are the decomposition of the full-replace augmentation model presented in Section 4.

Results Results are presented in Tables 3 and 4. Results are consistent with previous experiments in that augmentation with concept annotations does not improve performance. For both ontologies, neither the Dummy Concepts nor the Concepts Only models outperform the full-text models (in which both token representations are used). However, there are some interesting findings. Using SNOMED annotations, performance of the Concepts Only model is significantly higher than Dummy Concepts and very close to full-text model performance. This finding is strengthened by considering the concept coverage discussed in Table 1: the Concepts Only model achieves comparable performance receiving only about 35% (1% in the ICD9 setting) of the input tokens which the full-text baseline receives, and the Dummy Concepts Model receives about 65% (99% in the ICD9 setting). Thus, a significant proportion of downstream prediction performance can be attributed a small portion of the text which is recognized by cTAKES in both the SNOMED and ICD9 settings, indicating the strength of cTAKES' NER component.

5 Approach 2: Multi-task Learning

We present an alternative application of cTAKES as a form of distant supervision. Our approach is inspired by recent successes in multi-task learning for NLP which demonstrate that cheaply-obtained labels framed as an auxiliary task can improve performance on downstream tasks (Swayamdipta et al., 2018; Ruder, 2017; Zhang and Weiss, 2016). We propose to predict clinical information extraction system annotations as an auxiliary task, and share lower-level representations with the clinical coding task through a jointly-trained model architecture. We hypothesize that domain-knowledge embedded in cTAKES will guide the shared layers of the model architecture towards a more optimal representation for the clinical coding task.

We formulate the auxiliary task as follows: given each word-embedding or word-embedding span in the input which cTAKES has assigned a code, can the model predict the code assigned to it by cTAKES?

	Token Representation		AUC		AP		F1		R@k		P@k	
Model	Concept Match	No Match	Macro	Micro	Macro	Micro	Macro	Micro	8	15	8	15
Baseline (Mullenbach et al., 2018)	Word	Word	**0.8892**	**0.9846**	**0.2492**	**0.5426**	**0.0796**	**0.5421**	**0.3731**	**0.5251**	**0.7120**	**0.5616**
Dummy Concepts	0	Word	0.8876	0.9839	0.2119	0.5236	0.0732	0.5261	0.3634	0.5141	0.6943	0.5506
Concepts Only	Word	0	0.7549	0.9626	0.0538	0.2487	0.0080	0.1961	0.2063	0.2880	0.4196	0.3197
Concepts Only, Concept Embeddings	Concept	0	0.7534	0.9620	0.0552	0.2464	0.0086	0.1972	0.2058	0.2855	0.4200	0.3166
Augmentation Model (full replace)	Concept	Word	0.8846	0.9838	0.2242	0.5329	0.0691	0.5363	0.3688	0.5189	0.7048	0.5564

Table 3: Test set results of ablation experiments on the MIMIC-III dataset, using ICD9 concept annotations.

	Token Representation		AUC		AP		F1		R@k		P@k	
Model	Concept Match	No Match	Macro	Micro	Macro	Micro	Macro	Micro	8	15	8	15
Baseline (Mullenbach et al., 2018)	Word	Word	**0.8892**	**0.9846**	**0.2492**	**0.5426**	**0.0796**	**0.5421**	**0.3731**	**0.5251**	**0.7120**	**0.5616**
Dummy Concepts	0	Word	0.8472	0.9780	0.1461	0.4375	0.0413	0.4426	0.3202	0.4439	0.6234	0.4804
Concepts Only	Word	0	0.8736	0.9817	0.2059	0.4518	0.0515	0.4295	0.3278	0.4583	0.6300	0.4903
Concepts Only, Concept Embeddings	Concept	0	0.8739	0.9813	0.2019	0.4451	0.0519	0.4258	0.3247	0.4538	0.6254	0.4851
Augmentation Model (full replace)	Concept	Word	0.8744	0.9830	0.2221	0.5271	0.0724	0.5326	0.3675	0.5177	0.7022	0.5547

Table 4: Test set results of ablation experiments on the MIMIC-III dataset, using SNOMED concept annotations.

5.1 Method

We annotate the set of non-null ground-truth codes output by cTAKES for document i in the training data as $\{(a_{i,1}, c_{i,1}), (a_{i,2}, c_{i,2}), \ldots, (a_{i,M}, c_{i,M})\}$, where each anchor $a_{i,m}$ indicates the span of tokens in the text for which concept $c_{i,m}$ is annotated, and $c_{i,m} \neq \varnothing$.

The loss term of the model is augmented to include the multi-class cross-entropy of predicting the correct code for all annotated spans in the training batch:

$$\mathcal{L} = \sum_{i=1}^{I} BCE(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) + \lambda \frac{\sum_{i=1}^{I} \sum_{m=1}^{M_i} - \log p(c_{i,m} \mid a_{i,m})}{\sum_{i=1}^{I} M_i}$$

where $BCE(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i)$ is the standard (binary cross-entropy) loss from the baseline for the clinical coding task, $p(c_{i,m} \mid a_{i,m})$ is the probability assigned by the auxiliary model to the true cTAKES-annotated concept given word span $a_{i,m}$ as input, λ is the hyperparameter to tradeoff between the two objectives, and I is the number of instances in the batch.

Because we use the auxiliary task as a "scaffold" (Swayamdipta et al., 2018) for transferring domain knowledge encoded in cTAKES' rules into the learned representations for the clinical coding task, we must only run cTAKES and compute a forward pass through the auxiliary module at training time. At test-time, we evaluate only on the clinical coding task, so the time complexity of model inference remains the same as the baseline, an advantage of this architecture.

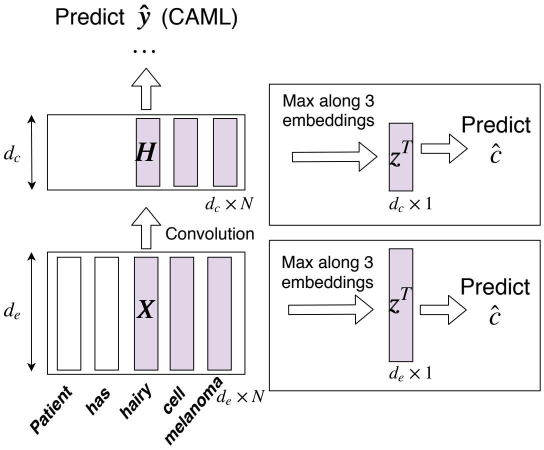


Figure 5: The proposed architecture (for prediction on a single document, i, and auxiliary supervision on a single annotation, m). The bottom box illustrates the pre-convolution model, and the top box post-convolution. The architecture on the left is the baseline.

We model $p(c_{i,m} \mid a_{i,m})$ via a multi-layer perceptron with a Softmax output layer to obtain a distribution over the codeset, $\mathcal{C}$. We additionally experiment with a linear layer variant to combat overfitting on the auxiliary task by reducing the capacity of this module. The input to this module is a single vector, $\boldsymbol{z}_{i,m} \in \mathbb{R}^{d_e}$, constructed by selecting the maximum value over s word embeddings for each dimension, where s is the length of the input span.[9] To facilitate information transfer between the clinical coding and auxiliary task, we experiment with tying both the randomly-initialized embedding layer, $\boldsymbol{X}$, and a higher-level layer of the

[9] While this is simple representation, we find that multi-word concept annotations are rather rare, in which case $\boldsymbol{z}_{i,m}$ is equivalent to $\boldsymbol{x}_{i,m}$.

Shared Features	Auxiliary Model	AUC Macro	AUC Micro	AP Macro	AP Micro	F1 Macro	F1 Micro	R@k 8	R@k 15	P@k 8	P@k 15
Baseline (Mullenbach et al., 2018)	n/a	**0.8892**	**0.9846**	**0.2492**	**0.5426**	**0.0796**	**0.5421**	**0.3731**	0.5251	**0.7120**	0.5616
Pre-convolution	MLP	0.8874	0.9839	0.2365	0.5390	0.0734	0.5376	0.3724	0.5235	0.7102	0.5597
Pre-convolution	Linear Layer	0.8834	0.9838	0.2398	0.5412	0.0766	0.5414	**0.3731**	**0.5265**	0.7113	**0.5633**
Post-convolution	MLP	0.7252	0.9619	0.0578	0.3002	0.0159	0.2966	0.2449	0.3417	0.4879	0.3748
Post-convolution	Linear Layer	0.7562	0.9655	0.0606	0.3035	0.0123	0.2934	0.2461	0.3392	0.4900	0.3700

Table 5: Test set performance on the ICD9 coding task for $\lambda = 1$ and using ICD9 annotations.

Shared Features	Auxiliary Model	Tagging Accuracy After one epoch	Tagging Accuracy After last epoch
Pre-convolution	MLP	0.9343	0.9398
Pre-convolution	Linear Layer	0.8940	**0.9400**
Post-convolution	MLP	0.9102	0.9335
Post-convolution	Linear Layer	0.7524	0.9341

Table 6: Dev set performance on the auxiliary task for $\lambda = 1$ and using ICD9 annotations. Relatively high task performance is achieved even after one epoch with a simple model.

network (e.g. the outputs of the document-level convolution layer $\boldsymbol{H}$ described in Section 3). See Figure 5 for the model architecture.

5.2 Experiment and Results

Results are presented in Table 5 and Table 6 for ICD9 annotations. Overall, the cTAKES span-prediction task does more to hurt than help performance on the main task. Tying the model weights at a higher layer (post-convolution as opposed to pre-convolution) results in worse performance, even though the model fits the auxiliary task well. This indicates either that the model may not have enough capacity to adequately fit both tasks, or that the cTAKES prediction task as formulated may actually misguide the clinical coding task slightly in parameter search space.[10]

We additionally remark that increasing the weight of the auxiliary task generally lowers performance on the clinical coding task, and tuning λ on the dev set does not result in more optimal performance (we include results with $\lambda = 1$ here; see Table 9 in the Appendix). Notably, for even very small values of λ, we achieve very high validation accuracy on the auxiliary task. This performance does not change with larger weightings, indicating that the auxiliary task may not be difficult enough to result in effective knowledge transfer.[11]

6 Conclusion

Integrating existing clinical information extraction tools with deep learning models is an important direction for bridging the gap between rule-based and learning-based methods. We have provided an analysis of the quality of the widely-used clinical concept annotator cTAKES when integrated into a state-of-the-art text-based prediction model. In two settings, we have shown that cTAKES does not improve performance over raw text alone on the clinical coding task. We additionally demonstrate through error analysis and ablation studies that the amount of word variation captured and the differentiation between the named-entity recognition and ontology-mapping tasks may affect cTAKES' effectiveness.

While automated coding is one application area, the models presented here could easily be extended to other downstream prediction tasks such as patient diagnosis and treatment outcome prediction. Future work will include evaluating newly-developed clinical NER tools with similar functionalities to cTAKES in our framework, which can potentially serve as a means to evaluate the effectiveness of newer systems vis-à-vis cTAKES.

Acknowledgments We thank the Research, Development, and Dissemination group at Sutter Health as well as members of the Georgia Tech Computational Linguistics Lab for helpful discussions. Figure 3 was provided by James Mullenbach.

References

American Academy of Professional Coders. 2019. What is medical coding?

Alan R Aronson. 2001. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly

[10] We found similar results using SNOMED annotations.

[11] While the models in Sections 4 did not introduce new hyperparameters to the baseline architecture, hyperparameters for this architecture were selected by human intuition. Room for future work includes more extensive tuning (see Table 8 in Appendix A).

learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Tal Baumel, Jimena Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie Elhadad. 2018. Multi-label classification of patient notes a case study on icd code assignment. In *Proceedings of the 2018 AAAI Joint Workshop on Health Intelligence*.

Willie Boag and Hassan Kané. 2017. Awe-cm vectors: Augmenting word embeddings with a clinical metathesaurus. *arXiv preprint arXiv:1712.01460*.

Willie Boag, Elena Sergeeva, Saurabh Kulshreshtha, Peter Szolovits, Anna Rumshisky, and Tristan Naumann. 2018. Cliner 2.0: Accessible and accurate clinical concept extraction. *arXiv preprint arXiv:1803.02245*.

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. 2017. Gram: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 787–795. ACM.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM.

Kevin Donnelly. 2006. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279.

Carol Friedman. 2000. A broad-coverage natural language processing system. In *Proceedings of the AMIA Symposium*, page 270. American Medical Informatics Association.

Nathan Greenberg, Trapit Bansal, Patrick Verga, and Andrew McCallum. 2018. Marginal likelihood training of bilstm-crf for biomedical named entity recognition from disjoint label sets. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2824–2829.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035.

Donald AB Lindberg, Betsy L Humphreys, and Alexa T McCray. 1993. The unified medical language system. *Methods of information in medicine*, 32(04):281–291.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1.

National Center for Health Statistics. 1991. *The International Classification of Diseases: 9th Revision, Clinical Modification: ICD-9-CM*.

Hoang Nguyen and Jon Patrick. 2016. Text mining in clinical domain: Dealing with noise. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 549–558. ACM.

T.J. Pollard and A.E.W. Johnson. 2016. The MIMIC-III clinical database.

Anthony Rios and Ramakanth Kavuluru. 2018a. Emr coding with semi-parametric multi-head matching networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 2081–2091.

Anthony Rios and Ramakanth Kavuluru. 2018b. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432.

Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. 2019. Enhancing clinical concept extraction with contextual embedding. *arXiv preprint arXiv:1902.08691*.

Anima Singh, Girish Nadkarni, John Guttag, and Erwin Bottinger. 2014. Leveraging hierarchy in medical codes for predictive modeling. In *Proceedings of the 5th ACM conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 96–103. ACM.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. *CoRR*, abs/1808.10485.

Yifeng Tao, Bruno Godefroy, Guillaume Genthial, and Christopher Potts. 2018. Effective feature representation for clinical text concept extraction. *arXiv preprint arXiv:1811.00070*.

Jacqueline M Torres, John Lawlor, Jeffrey D Colvin, Marion R Sills, Jessica L Bettenhausen, Amber Davidson, Gretchen J Cutler, Matt Hall, and Laura M Gottlieb. 2017. Icd social codes. *Medical care*, 55(9):810–816.

Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, et al. 2017. Clinical information extraction applications: a literature review. *Journal of biomedical informatics*.

Wei-Hung Weng, Kavishwar B Wagholikar, Alexa T McCray, Peter Szolovits, and Henry C Chueh. 2017. Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach. *BMC medical informatics and decision making*, 17(1):155.

Kai Xu, Zhanfan Zhou, Tao Gong, Tianyong Hao, and Wenyin Liu. 2018. Sblc: a hybrid model for disease named entity recognition based on semantic bidirectional lstms and conditional random fields. *BMC medical informatics and decision making*, 18(5):114.

Yuan Zhang and David Weiss. 2016. Stack-propagation: Improved representation learning for syntax. *arXiv preprint arXiv:1603.06598*.

Henghui Zhu, Ioannis Ch Paschalidis, and Amir Tahmasebi. 2018. Clinical concept extraction with contextual word embedding. *arXiv preprint arXiv:1810.10566*.

A Experimental Details

Data Following Mullenbach et al. (2018), we use the same train/test/validation splits for the MIMIC-III dataset, and concatenate all supplemental text for a patient discharge summary into one record. We use the authors' provided data processing pipeline[12] to preprocess the corpus. The vocubulary includes all words occurring in at least 3 training documents. See Table 7 for descriptive statistics of the dataset.

We construct a concept vocabulary for embedding initialization following the same specification as the word vocabulary: any concept which does not occur in at least 3 training documents is replaced with an UNK token. Details on the size of the vocabulary can be found in Table 8.

# training documents	47,723
# test documents	3,372
# dev documents	1,631
Mean # tokens per document	1,513.0
Mean # labels per document	16.09
Total # labels (L)	8,921

Table 7: Dataset Descriptive Statistics.

Training We train with the same specifications as Mullenbach et al. (2018) unless otherwise specified, with dropout performed after concept augmentation for the models in Sections 4, and early stopping with a patience of 10 epochs on the precision at 8 metric, for a maximum of 200 epochs (note that in the multi-task learning models the stopping criterion is only a function of performance on the clinical coding task). Unlike previous work, we reduce the batch size to 12 in order to allow each batch to fit on a single GPU, and we do not use pretrained embeddings as we find this improves performance. All models are trained on a single NVIDIA Titan X GPU with 12,189 MiB of RAM.

We port the optimal hyperparameters reported in Mullenbach et al. (2018) to our experiments. With more extensive hyperparameter tuning, we may expect to see a potential increase in the performance of our models over the baseline. See Table 8 for hyperparameters and other details specific to our proposed model architectures. All neural models are implemented using PyTorch[13], and built on the open-source implementation of CAML.[14]

Parameter	Value
Vocabulary Size	51,917
SNOMED Concept Vocabulary ($\mathcal{C}$) Size	20,775
ICD9 Concept Vocabulary ($\mathcal{C}$) Size	1,529
Embedding Size (d_e)	100
Post-convolution Embedding Size (d_c)	50
Dropout Probability	0.2
Learning Rate	0.0001
Attention Mechanism Hidden State Size	20
Attention Mechanism Activation Function	ReLU
Auxiliary hidden layer size	700
Auxiliary activation function	ReLU

Table 8: Model details.

B Concept Extraction

We build a custom dictionary from the UMLS Metathesaurus that includes mappings from UMLS CUIs to SNOMED-CT and ICD9-CM concepts. We run the cTAKES annotator in advance of training for all 3 dataset splits using the resulting dictionary, allowing us to obtain annotations for each note in the dataset, and the positions of the annotations in the raw text. Note that for the multi-task learning experiments (Section 5), we only require annotations for training data. Annotating the MIMIC-III datafiles using these specifications takes between 4 and 5 hours for 3,000 discharge summaries on a single CPU, and can be parallelized for efficiency.

C Attention for Overlapping Concepts

We implement an attention mechanism (Bahdanau et al., 2014) to compute a single concept embedding $\phi(\mathcal{C}_n) \in \mathbb{R}^{d_e}$ when $\mathcal{C}_n = \{c_1, c_2, \ldots, c_J\}$ represents a set of concepts annotated at position n instead of a single concept. Intuitively, we want to more heavily weight those concepts in the set which have the most similarity to the surrounding text. We define a context vector for position n as:

$$\boldsymbol{v}_n = [\boldsymbol{x}_{n-2}, \boldsymbol{x}_{n-1}, \boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+2}] \in \mathbb{R}^{4d_e}$$

The context is defined as the concatenated word embeddings surrounding position n. We use a context size of $n + / - 2$, where 2 is a hyperparameter. We choose to use a smaller value for computational efficiency.

[12] https://github.com/jamesmullenbach/caml-mimic/blob/master/notebooks/dataproc_mimic_III.ipynb

[13] https://github.com/pytorch/pytorch

[14] https://github.com/jamesmullenbach/caml-mimic

λ	AUC		AP		F1		R@k		P@k		Auxiliary Tagging Accuracy	
	Macro	Micro	Macro	Micro	Macro	Micro	8	15	8	15	After one epoch	After last epoch
0.001	**0.9002**	**0.9848**	0.3129	0.5470	**0.0704**	**0.5511**	0.3902	0.5447	0.7164	0.5631	0.8888	0.9398
0.01	0.8954	0.9842	0.2885	0.5352	0.0636	0.5425	0.3843	0.5328	0.7088	0.5528	0.8938	**0.9401**
0.1	0.9000	0.9846	**0.3145**	0.5465	0.0689	0.5471	**0.3909**	0.5426	**0.7183**	0.5617	0.8940	0.9400
0.5	0.8934	0.9840	0.2892	0.5362	0.0624	0.5386	0.3844	0.5361	0.7089	0.5546	**0.8941**	0.9400
1	0.8975	0.9840	0.3087	0.5460	0.0668	0.5477	0.3886	0.5439	0.7169	0.5624	0.8940	0.9400
10	0.8979	0.9842	0.3122	**0.5484**	0.0678	0.5474	0.3908	**0.5457**	0.7182	**0.5644**	0.8940	0.9400
50	0.8939	0.9837	0.2982	0.5410	0.0638	0.5427	0.3855	0.5391	0.7111	0.5592	0.8940	**0.9401**
100	0.8913	0.9835	0.2943	0.5383	0.0632	0.5407	0.3849	0.5374	0.7096	0.5577	0.8940	**0.9401**
1000	0.8851	0.9827	0.2750	0.5260	0.0564	0.5309	0.3803	0.5290	0.7016	0.5491	0.8940	**0.9401**

Table 9: The effect of tuning λ on dev set performance on the ICD9 coding task, for the pre-convolution model with a linear auxiliary layer and ICD9 annotations. We select $\lambda = 1$ for reporting test results; there isn't a clear value which produces strictly better performance.

We concatenate the word-context vector and each concept embedding c_j in $\mathcal{C}_n$ as $[\boldsymbol{v}_n, \phi(c_j)] \in \mathbb{R}^{5d_e}$, and pass it through a multi-layer perceptron to compute a similarity score: $f : \mathbb{R}^{5d_e} \rightarrow \mathbb{R}^1$. An attention score for each c_j is computed as:

$$\alpha_j = \frac{exp(f(\boldsymbol{v}_n, \phi(c_j))}{\sum_{k=1}^{J} exp(f(\boldsymbol{v}_n, \phi(c_k))}$$

This represents the relevance of the concept to the surrounding word-context, normalized by the other concepts in the set. A final concept embedding $\phi(\mathcal{C}_n) \in \mathbb{R}^{d_e}$ is computed as a linear combination of the concept vectors, weighted by their attention scores:

$$\phi(\mathcal{C}_n) = \sum_{j=1}^{J} \alpha_j \cdot \phi(c_j)$$

D Leveraging Ontological Graph Structure

Following the methodology of Choi et al. (2017), we experiment with learning higher-quality concept representations using the hierarchical structure of the ICD9 ontology. We replace concept embedding $\phi(c_n)$ with a learned linear combination of itself and its parent concepts' embeddings (see Figure 1). For child concepts which are observed infrequently or have poor representations, prior work has shown that a trained model will learn to weight the parent embeddings more heavily in the linear combination. Because the parent concepts represent more general concepts, they have most often been observed more frequently in the training set and have stronger representations. This also allows for learned representations which capture relationships between concepts. We refer the reader to Choi et al. (2017) for details.

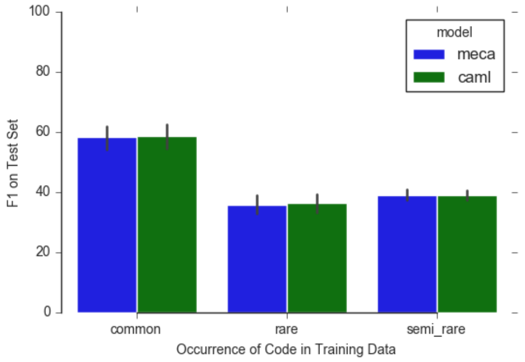


Figure 6: F1 on Test Data based on Frequency of Codes in Training Data, where the metric is defined ('meca' indicates the *linear combination* ICD9 augmentation model).