Alleviating the Symptoms
Extracting Coded MedDRA Events From Raw VAERS Case Reports

Mike Stackhouse and Chris Ventura

Abstract

Medical data today comes from a variety of sources and in widely varying formats. These data can come from standardized device outputs, lab results, questionnaire results directly from the patient, and more. A core challenge, however, is that much of these data are found in free text fields, which typically require manual standardization to become valuable in both a medical and analytical context. An example of one such case are adverse events (i.e. a headache, a heart attack, inflammation at an infusion site, etc.). While often initially recorded as free text by a medical provider during a patient encounter, these data must typically be coded to a standardized dictionary to be used in some sort of analysis or billing procedure.

In this paper, we replicate the work of Mullenbach et al. 2018, who created a novel convolutional neural network with an attention mechanism, named CAML, to extract standard ICD 9 codes from free text ICU medical records. Instead of identifying ICD 9 codes, we instead predict standardized adverse event names from the MedDRA dictionary with data from the FDA Vaccine Adverse Event Reporting System (VAERS) data archives. Using this method, we achieve a precision@8 of 0.332632 and a Micro-F1 of 0.247152. We further demonstrate the value of the explainability of predictions as shown in Mullenbach et al. 2018.

Introduction

While advances in both technology and the personalizing of healthcare have dramatically increased the volume of health data available, more than 80 percent of health data remains unstructured (Kong 2019). Clinical trials involve massive efforts of data standardization. Even today, the vast variety of formats and sources of data that are used throughout the course of a trial prove to be a costly and time consuming effort. One such effort is the standardization of collected adverse events encountered by a subject through the course of a trial.

When an adverse event occurs, the data collected come directly from the clinician at the site or hospital. The adverse event is often described in free text, where the doctor reports the symptoms encountered in their own words. Often there are multiple adverse events encountered at one time. For example, a headache and fever might be listed in the same case report – but these are two separate events that, in a standardized format, should be collected separately.

In the late 1990s, the Medical Dictionary for Regulatory Activities was developed by the International Council for Harmonization of Technical Requirements for Pharmaceuticals for Human Use (ICH) (med a). This gave the industry a vast and highly specific standardized dictionary of medical terminology to facilitate sharing of regulatory information. MedDRA coding of adverse events is additionally required in a regulatory submission to the FDA (fed 2017). Given the highly specific nature of MedDRA and the free text nature of data collection, this proves to be a difficult area to develop efficient automation.

Background

The work done in Mullenbach et al. focuses on medical coding for insurance billing, which is valued as an estimated $10.6B industry (med b). The financial benefits within adverse event coding are very similar, as the faster and more accurately data standardization can occur, the faster analysis can be conducted to safely bring a drug to market.

Coding adverse events from free text essentially breaks down to a classification problem. The nuance to this is that it's not a standard classification problem, but a multiple label classification problem with thousands of potential labels. Problems like these have been approached in natural language processing on numerous occasions and are known as Extreme Multi-Label Text Classification (XMTC). One such example is Chalkidis et al. 2019, where legal documents were labeled with concepts from EUROVOC, a multidisciplinary thesaurus. Some other work on multi-label classification aims to leverage relationships between classified labels. Kurata, Xiang, and Zhou 2016 explore leveraging label co-occurance to enhance predictability.

Accurate predictions in an regulatory environment like clinical trials are simply not enough. Even 95% accuracy leaves a 5% chance of error the must be reviewed and corrected, which will still require large amounts of manual effort. In Mullenbach et al.'s work, their model is not only able to classify ICD 9 codes, but additionally outline k-grams of text that are relevant to each classification. The explainability of their model, Convolutional Attention for Multi-Label classification (CAML), predictions provides a very real po-

	VAERS Data
# documents	729445
vocab size	63962
Mean # tokens per document	80.54
Median # tokens per document	40
Mean # labels per document	3.75
Median # labels per document	3
Max # labels per document	288
Total # labels	11259

Table 1: Descriptive statistics on VAERS data set

tential benefit as a system to aide a human coder, accelerating their work and assisting to check accuracy, even if the work of the human coder is not fully automated.

Data

The primary data set we used for training, prediction, and evaluation of our methodology was the FDA Vaccine Adverse Event Reporting System (VAERS) data archives. The VAERS data archives contain annual files dating back to 1990 with mandatory and voluntarily reported adverse events following the administration of a vaccine. In every archive file, each record contains a string field of the original free text description of the symptom responsible for the event, as well as the standardized MedDRA symptom labels. All records contain at least one symptom label.

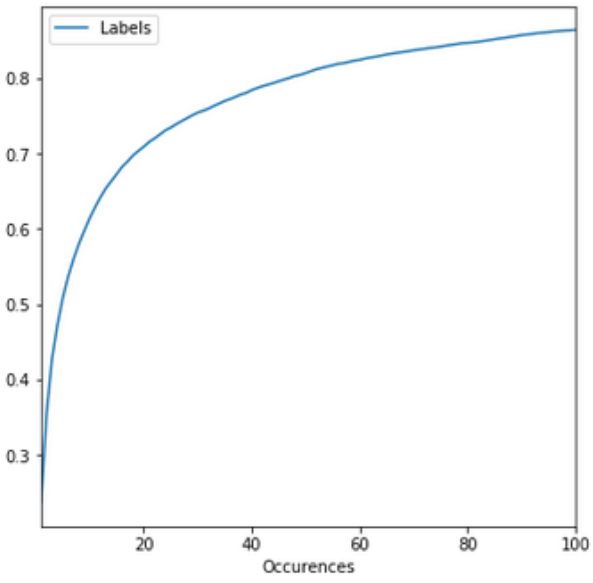

Figure 1: Proportion of labels with N occurrences

Across the archive files are 11,261 unique symptom labels spread across 729,445 unique event records, which we separated using a 80:15:5 Train/Test/Development split, resulting in 553,556 training records, 109,416 test records, and 36,473 development records. Figure 2 shows the frequency of the most frequent 25 labels.

While the most common labels appear quite frequently, the distribution quickly levels off. Figure 1 shows the proportion of labels with N occurrences. Here we see that 70% of labels have fewer than 20 occurrences, which is part of our motivation for including additional data sources in our embeddings (see Embeddings).

The adverse event free text descriptions of symptoms pose some additional challenges not necessarily present in other types of text data. To illustrate this further, we provide one example from one event record illustrating several of the challenges with this type of data.

> ”Pt vaccinated with DTP developed high pitched irritable cry, unconsolable x 1 to 1 1/2 hr. Large golf ball size knot at injection site. Seen by MD fever 102, Given Tylenol & use compress. Only DT in future. 05DEC90 T99.2, knot quarter size”

Perhaps most noticeable to those unfamiliar with medical data is the use of medical jargon, abbreviations, and acronyms. For example, ”Pt” is used instead of patient. The text also lists the medication used to treat the event (Tylenol) as well as the date of the event treatment. Furthermore, the text contains ”T99.2”, a code that could correspond to the original vaccine batch number, but this is not explicitly stated.

While the specifics of this text are common in medical data, they do not resemble most publicly available large text corpora. As a result, we make the conscious choice to only augment our embeddings with medial descriptive text. Additionally, we take some additional preprocessing steps to prevent our model from learning from specific types of strings, such as dates.

Methodology

In our work, we follow the framework of Mullenbach et al. 2018. We use their PyTorch code base (provided here), modified to adapt to our specific use case. We will explicitly mention portions of the code that we had to re-purpose for our use.

Data Preprocessing

Data preprocessing was one area in particular we invested heavily. The VAERS data were stored in separate packages by year, with separate files in each package. Once consolidated and merged, the data sets only contained 5 potential label columns, but case reports could span multiple records, linked by a unique identifier. The maximum number of labels that could actually be attached to a single case report was 288. More descriptive statistics on the data are available in Table 1.

We followed all specific pre-processing instructions as outlined in Mullenbach et al. 2018. We opted to develop our own code to conduct this step, as it fit in with the preparation of our data set from source. The standard tokenizer within Scikit-Learn’s CountVectorizer was used. Following Mullenbach et al., digit-only tokens were dropped, and concepts that occurred in less than 3 documents were removed. We took the additional step to consolidate obvious date tokens into one single concept of DATE, as we noted a large

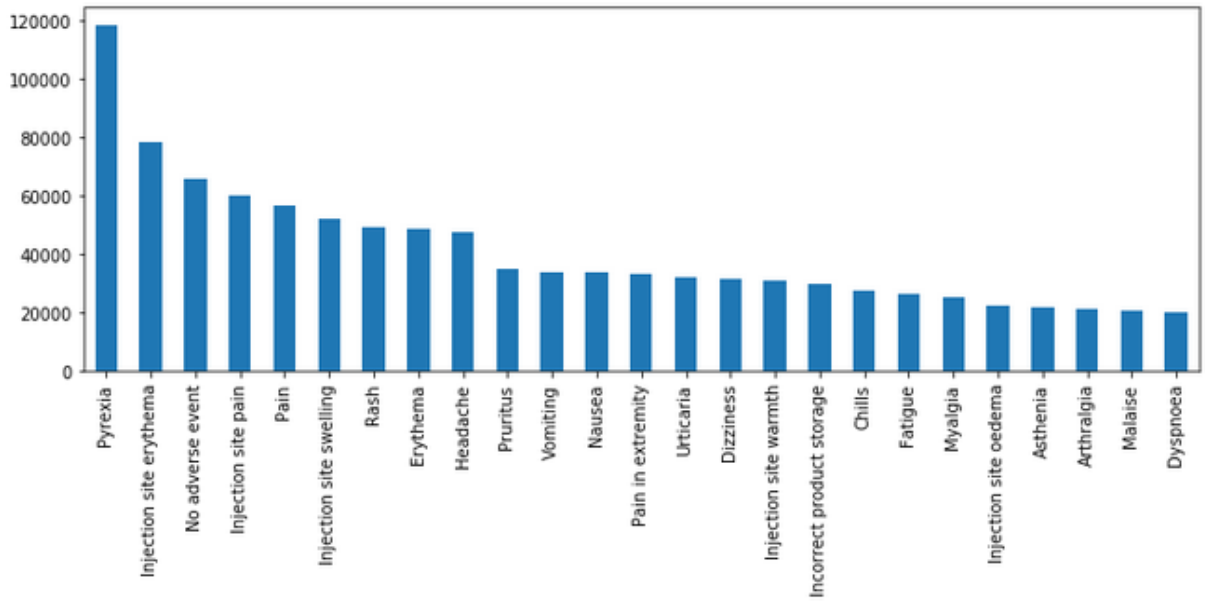

Figure 2: Frequency of most frequent 25 labels

number of unique dates within our vocabulary that added thousands of extra concepts.

This section was particularly challenging and time consuming, as the MIMIC data sets used in the original data sets require a training course to access. In the interest of time, we were not able to complete this. Therefore, we had to reverse engineer the code from the original Mullenbach et al. repository to figure out how our data needed to be to formatted to work within the model architecture.

Embeddings

In Mullenbach et al. 2018, word embeddings were retrained on the MIMIC data set, which provided a large enough corpus to create proper embeddings. Using a pre-trained corpus is not advisable in this setting, as a great deal the text contains medical terminology in contexts likely not prominent in the pre-trained embedding corpora, and would thus leave us with many important unknown tokens (see example in Data). As a result, we created our own embeddings for the purposes of this paper.

The VAERS database has much less data than either MIMIC-II or MIMIC-III, therefore we opted to supplement our data further. In addition to the VAERS data, we extracted case report submissions from the FDA Manufacturer and User Facility Device Experience (MAUDE) database (ope). Furthermore, we included Wikipedia summary data used within the description regularizer (discussed in Regularization) to train the embeddings as well. All of these data combined were trained using Word2Vec. Following Mullenbach et al. 2018, an embedding size of 100 was used.

Here we modified the code used in the original CAML repository to train embeddings on multiple data sources, as given more time we would have sought other data sources and used this as part of our tuning process, exploring which data sources work best and provide the strongest embeddings. For the target audience of this model, for instance, a pharmaceutical company - they would likely be able to provide enough data to properly train strong embeddings.

These embeddings make up the base layer of CAML, and are concatenated into the matrix $X = [x_1, x_2, ..., x_N]$, where N is the document length.

Convolutional Architecture

Following the notation of Mullenbach et al. 2018, the next stage of CAML is to combine adjacent embeddings using a convolutional filter $W_c \in \mathbb{R}^{kxd_exd_c}$, where k is the filter width, d_e the size of the input embedding, and d_c is the size of the final output. At each step n, we compute

$$h_n = g(W_c * x_{n:n+k-1} + b_c) \quad (1)$$

where $*$ denotes the convolution operator, g is an element-wise nonlinear transformation, and $b_c \in \mathbb{R}^{d_c}$ is the bias. We additionally pad each side of the input with zeros so that the resulting matrix H has dimension $\mathbb{R}^{d_c XN}$

Attention Mechanism

After convolution, the document is represented by the matrix $H \in \mathbb{R}^{d_c XN}$. As the goal of this model is to apply multiple labels, the attention mechanism is applied per label. As different segments of a document may be relevant to separate labels, this allows the model to focus on different representations for each potential label. An added benefit is that the model selects the k-grams form the text that are most relevant to each predicted label - which provides some explainability to the decision making of the model.

Continuing with the notation of Mullenbach et al. 2018, for each label ℓ, we compute the matrix-vector product,

	AUC		F1		P@n	
Model	Macro	Micro	Macro	Micro	8	15
Logistic Regression	0.000431	0.503656	0.007958	0.011133	0.001423	0.001608
CNN	**0.920461**	**0.991388**	**0.069807**	**0.531226**	0.294809	0.179437
DR-CAML	0.89693	0.990811	0.022955	0.247152	**0.332632***	**0.200795**

Table 2: Model Evaluation results. The target metric of training and model selection was precision@8, which DR-CAML achieved the highest score.

$H^T u_\ell$, where $u_\ell \in \mathbb{R}^{d_c}$ is a vector parameter for label ℓ. We then pass the resulting vector through a SoftMax operator, obtaining a distribution over locations in the document,

$$\alpha_l = SoftMax(H^T u_l) \tag{2}$$

where $SoftMax(x) = \frac{exp(x)}{\sum_i exp(x_i)}$ and $\exp(x)$ is the element-wise exponentiation of vector x. The attention vector α is then used to compute vector representations for each label,

$$\nu_l = \sum_{n=1}^{N} \alpha_{l,n} h_n \tag{3}$$

Classification

Continuing with the notation of Mullenbach et al. 2018, given the vector document representation v_ℓ, we compute the probability for label ℓ using another linear layer and a sigmoid transformation:

$$\widehat{y}_l = \sigma(\beta_l^T \nu_l + b_l) \tag{4}$$

$\beta_\ell \in \mathbb{R}^{d_c}$ is a vector of prediction weights, and b_ℓ is a scalar offset. A diagram of the model can be seen in the original paper (Mullenbach et al. 2018).

Regularization

One of the most novel ideas in Mullenbach et al. 2018 was a technique they named description regularization. A large number of labels in their ICD-9 code set, and for our purpose, the MedDRA dictionary, rarely occur - which is inherently the nature of a highly specific code set. To counter this, they added a secondary module to CAML. The module learns to embed the descriptions as vectors, and then uses them as the target of regularization on the model parameters β_ℓ. If a code was rarely seen, then this encourages its parameters to be similar to codes with similar descriptions.

Continuing with their notation, this module is made up of a max-pooling CNN architecture. Let z_ℓ be a max-pooled vector, obtained by passing the description for code ℓ into the module. Let n_y be the number of true labels in a training example. We add the following regularizing object to our loss L,

$$L(X, y) = L_{BCE} + \lambda \frac{1}{n_y} \sum_{l:y_l=1}^{\mathcal{L}} \|z_l - \beta_l\|_2 \tag{5}$$

where λ is a tradeoff hyperparameter that calibrates the performance of the two objectives. Mullenbach et al. refer to this variation of the model as Description Regularized-CAML (DR-CAML).

In our case, we unfortunately did not have access to the full MedDRA dictionary, as this requires a license for use. To supplement our model, we instead scraped Wikipedia summaries for each of our labels. This code can be found here. Not every label had a Wikipedia page explicitly for the text of the label, so we built in some contingencies. If the page did not exist, but other pages were suggested, we would pull out a page that had "medical" or "medicine" in the name, or if there were less than 10 suggestion, take the first one. If no page was found, we would drop the last word and search again. This helped in cases like "ECG Abnormal", where "Abnormal" does not tell us much about "ECG". If none of these attempts worked, we simply used the label itself as its own description. This is reasonable as many labels appear directly in the case report text itself. For embeddings, we used the full summary text. For the description regularizer data, we only used the first sentence to ensure that there was not too much noise in the descriptions, and the first sentence typically served as good and succinct summary. The same pre-processing was applied to these data as described in Data Preprocessing.

This is not a perfect solution, but it provided descriptive text of the symptom, and in Table 3 it can also be seen that these descriptions are reasonably informative.

Training

Using the same framework as Mullenbach et al. 2018, model training was done by minimizing the binary cross-entropy loss,

$$L_{BCE}(X, y) = -\sum_{l=1}^{\mathcal{L}} y_l log(\widehat{y}_l) + (1 - y_l) log(1 - \widehat{y}_l) \tag{6}$$

plus the L2 norm of the model weights, using the Adam optimizer. We used an early stopping criteria of exiting training if the precision@8 metric did not improve after 3 epochs. Our model finished training after 15 epochs. Unfortunately, DR-CAML proved extremely expensive to train. 15 epochs took around 11 hours to train on a NVIDIA GTX1070 GPU. While Mullenbach et al. were able to use a batch size of 16, we were only able to run a max batch size of 8 due to memory constraints.

Toothache:	"Toothache, also known as dental pain, is pain in the teeth or their supporting structures..."
	...after vaccination he had **intense pain to upper teeth then pain moved** *to lower teeth jaw and...*
Urticaria:	"Hives, also known as urticaria, is a kind of skin rash with red, raised, itchy bumps."
	...bottom of rt foot started **itching shortly started devel hives on arms upper legs hips following flu** *shot had...*
Hyperhidrosis:	"Hyperhidrosis is a condition characterized by abnormally increased sweating..."
	...bp hr bs o2 **ra very diaphoretic blurry vision** *encouraged to drink water*

Table 3: K-gram segments extracted from attention mechanism

Evaluation Metrics

To allow for comparisons to the results obtained in Mullenbach et al. 2018 and any future work, as well as across our baselines and models, we primarily report three metrics: F1, area under the ROC curve (AUC), and precision@n, the fraction of the n highest scored labels that are found in the label set for that event. Following Mullenbach et al. 2018, we report these metrics as micro-averaged and macro-averaged values, with the micro-averaged values evaluating each text-label pair as a separate prediction while the macro-averaged values average the metrics per-label. Again, following the notation of Mullenbach et al. 2018, the recall metrics would be represented as:

$$Micro - R = \frac{\sum_{l=1}^{|\mathcal{L}|} TP_l}{\sum_{l=1}^{|\mathcal{L}|} TP_l + FN_l} \tag{7}$$

$$Macro - R = \frac{1}{|\mathcal{L}|} \sum_{l=1}^{|\mathcal{L}|} \frac{TP_l}{TP_l + FN_l} \tag{8}$$

with TP and FN corresponding to true positives and false negatives, respectively. While we do not report recall explicitly, recall is used within our F1 scoring. Given the nature of these metrics, the micro-averaged values will place more emphasis on frequent labels, with the macro-averaged values emphasizing rare labels.

Baseline Models

In order to align with Mullenbach et al. 2018, we additionally utilized their baseline comparisons as well. We trained two additional models to provide points of comparison: a convolutional neural network (CNN) consisting of a single convolutional layer and a max pooling layer, and a bag-of-words logistic regression model. Both models were trained using the same embeddings and both models attempt to predict the same output labels as DR-CAML.

Parameter Tuning

As stated under Training, model training takes quite a long time, even on GPU. Due to these constraints, our capability to tune parameters was limited. We had an additional challenge of hitting memory errors late in an epoch, due to documents with higher token counts encountered late in the epoch.

We were able to manually tune on a few parameters. These parameters include d_c as the number of convolutional filters, k as the filter size, q as the dropout probability, and η as the learning rate. Final parameters to our DR-CAML model are shown in Table 4.

Parameter	Value
d_c	50
k	8
η	0.0001
q	0.2

Table 4: Final model parameters

Results

Results for our baselines and model are visible in Table 2. Admittedly, our they are not as strong as we hoped they would be. Using DR-CAML, our highest achieved precision@8 was $\sim$ 0.33. Compared to the 0.709 obtained in Mullenbach et al. 2018, this is quite low.

Interestingly, on other metrics the basic CNN was able to obtain higher scores. Despite this, our preferable model in this case is DR-CAML. A higher F1 score in our case is not necessarily our goal. We want to make sure that of the predictions that we make, most of them are true positives, which is what precision@8 and precision@15 allow us to focus on. Recall, on the other hand, can be inflated by allowing false positives, whereas precision penalizes them. If a case report only has two events, and both are predicted - along with 8 others, recall will be 1. On the other hand, precision will penalize this while F1 balances out the two metrics. Thus, the focus on precision helps our model stay more conservative on the number of labels triggered.

Both the CNN and DR-CAML well outperformed BOW Logistic Regression, but this is not shocking in the least bit, as the relationships within these data are far too complex for a linear relationship to handle.

Evaluating Interpretability

Despite the lower-than-desired metrics, we were still able to observe some explainability of the decision making of the model that had interesting results. Examples can be viewed in Table 3. The attention mechanism in the model was able to make the association that "Toothache" relates to "pain to upper teeth", that "Urticaria" relates to itching and hives, and that "Hyperhidrosis" and "diaphoretic" are both associated with increased sweating. These successes are evidence that the model is indeed learning proper relationships, and that the model could indeed be capable of fulfilling the use case we envisioned. The table further points out that the model was able to identify that the label related to this subset of text from the case report.

To extract relevant k-grams of text for a label, influential tokens are extracted using the attention mechanism. The

SoftMax output a_ℓ is put through a max-pooling CNN layer and weighted by the final layer weights. Following the notation of Mullenbach et al. 2018, an argmax vector a is defined as the result from the max pooling step

$$a_i = \underset{j \in \{1,\ldots,m-k+1\}}{\operatorname{argmax}} (H_{ij}) \tag{9}$$

From here we can compute the importance of position i for label ℓ,

$$\alpha_{il} = \sum_{j:a_j=i}^{d_c} \beta_{l,j} \tag{10}$$

With these results, the most important k-gram for a given label can be selected as $argmax_i \alpha_{i\ell}$

Conclusion and Next Steps

There's a great deal of room for improvement in our results, and the issues can be broken down on a few levels.

A fundamental problem we had is our data. While the VAERS database enabled us to make an applied attempt at this problem, it is flawed. The MedDRA dictionary evolves over time, and the dictionary codes from 1999 are not necessarily the same as today. New codes appear over time and may be missing from earlier data, and codes may change interpretation over time. Therefore, we have a consistency issue.

Furthermore, we did not have enough data to make truly robust embeddings. We assembled data from multiple sources that we we felt would help enhance the embeddings on our data set, but reliable embeddings require very large corpora.

Lastly, as we did not have a license for the MedDRA dictionary itself, we were not able to build a description regularizer. Though we feel we made a reasonable attempt to proxy these data, utilizing event descriptions from the MedDRA dictionary itself, similar to the ICD 9 descriptions used in Mullenbach et al. 2018, would likely perform significantly better.

All of this being said - we believe these problems are solvable. An organization that would want to apply this in a production setting would have the necessary resources to correct these problems. They would already require a MedDRA license, would have much more consistent data, and have much more data in general. Regardless, we feel we were able to demonstrate the potential and value of DR-CAML and the work of Mullenbach et al. applied in this setting.

References

[Chalkidis et al.] Chalkidis, I.; Fergadiotis, M.; Malakasiotis, P.; Aletras, N.; and Androutsopoulos, I. 2019. Extreme multi-label legal text classification: A case study in EU legislation. *CoRR* abs/1905.10892.

[fed] 2017. Electronic study data submission; data standards; support for version update of the medical dictionary for regulatory activities. https://www.federalregister.gov/documents/2017/08/31/2017-18471/electronic-study-data-submission-data-standards-support-for-version-update-of-the-medical-dictionary.

[Kong] Kong, H.-J. 2019. Managing unstructured big data in healthcare system. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6372467/.

[Kurata, Xiang, and Zhou] Kurata, G.; Xiang, B.; and Zhou, B. 2016. Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 521–526. San Diego, California: Association for Computational Linguistics.

[med] English. https://www.meddra.org/how-to-use/support-documentation/english.

[med] Medical coding market size analysis: Industry report, 2018-2025. https://www.grandviewresearch.com/industry-analysis/medical-coding-market.

[Mullenbach et al.] Mullenbach, J.; Wiegreffe, S.; Duke, J.; Sun, J.; and Eisenstein, J. 2018. Explainable prediction of medical codes from clinical text. *CoRR* abs/1802.05695.

[ope] openfda downloads. https://open.fda.gov/downloads/.