Classifying Unstructured Clinical Notes via Automatic Weak Supervision

Chufan Gao * CHUFANG@ANDREW.CMU.EDU

Mononito Goswami * MGOSWAMI@ANDREW.CMU.EDU

Jieshi Chen JIESHIC@ANDREW.CMU.EDU

Artur Dubrawski AWD@ANDREW.CMU.EDU

Auton Lab, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

Healthcare providers usually record detailed notes of the clinical care delivered to each patient for clinical, research, and billing purposes. Due to the unstructured nature of these narratives, providers employ dedicated staff to assign diagnostic codes to patients' diagnoses using the International Classification of Diseases (ICD) coding system. This manual process is not only time-consuming but also costly and error-prone. Prior work demonstrated potential utility of Machine Learning (ML) methodology in automating this process, but it has relied on large quantities of manually labeled data to train the models. Additionally, diagnostic coding systems evolve with time, which makes traditional supervised learning strategies unable to generalize beyond local applications. In this work, we introduce a general weakly-supervised text classification framework that learns from *class-label descriptions only*, without the need to use any human-labeled documents. It leverages the linguistic domain knowledge stored within pre-trained language models and the data programming framework to assign code labels to individual texts. We demonstrate the efficacy and flexibility of our method by comparing it to state-of-the-art weak text classifiers across four real-world text classification datasets, in addition to assigning ICD codes to medical notes in the publicly available `MIMIC-III` database.

1. Introduction

The Electronic Health Record (EHR) system is a digital version of a patient's paper chart. EHRs are almost-real-time, patient-centered records that contain patient history, diagnoses, procedures, medications, and more in an easily accessible format. Since the Health Information Technology for Economic and Clinical Health (HITECH) Act was signed into law in 2009 (Menachemi and Collum, 2011), adoption rates of these systems have steadily increased. Adler-Milstein et al. (2017), who analyzed survey data collected by American Hospital Association found that EHR adoption rates were at 80% in 2017, twice the rate in 2008. With higher adoption rates comes a rising challenge: data processing and analysis of unstructured clinical text. Natural language free-texts are regularly recorded in the form of radiology or discharge notes and are used for diagnostic, research, and billing purposes.

* Authors contributed equally to this research.

To be studied and managed adequately, cohorts of patients with similar clinical characteristics need reliable phenotype labels. However, specific phenotype data is seldom available compared to other EHR data, like clinical texts (Venkataraman et al., 2020). In practice, diagnostic codes are among the most common proxies to true phenotypes. Due to the unstructured nature of clinical notes, providers often employ trained staff and/or third-party vendors to help assign diagnostic codes using coding systems such as the International Classification of Diseases (ICD) (Alharbi et al., 2021). However, manual assignment of codes is both time consuming and error-prone, with only 60–80% of the assigned codes reflecting actual patient diagnoses (Benesch et al., 1997) and significant portion of misjudged severity of conditions and code omissions (Venkataraman et al., 2020). For healthcare providers, billing and coding errors may not only lead to loss of revenue and claim denials, but also federal penalties for erroneous Medicare and Medicaid claims. Thus, there is a clear need for reliable automated classification of unstructured clinical notes.

Prior work introduced the use of Machine Learning (ML) to automatically assign diagnostic codes to clinical notes (Venkataraman et al., 2020; Baumel et al., 2018; Yu et al., 2019; Xu et al., 2019). Yet, most of the involved ML models rely on vast quantities of pointillistically labeled training data, which is often unavailable or costly to collect. In addition, coding systems are periodically revised, rendering already labeled data at least partially obsolete. In fact, ICD is currently in its 10th revision[1], while its 11th revision has already been accepted by the World Health Organization and will come into effect on January 2022 (Wikipedia contributors, 2021). To make matters worse, most providers use their internal coding systems, making traditional supervised ML strategies infeasible to generalize across organizations.

As a potential remedy, we present `KeyClass`, a general weakly supervised text classification framework combining Data Programming (Ratner et al., 2016) with a novel method of automatically acquiring interpretable weak supervision sources (keywords and phrases) from *class-label descriptions only* without the need to access to any labeled documents. The successful application of `KeyClass` to solve an important clinical text classification problem demonstrates its potential for making social impact by allowing quick and affordable development and deployment of effective text classifiers. Our primary contributions include:

- We introduce a general weakly supervised text classification model `KeyClass`, and a novel strategy to effectively and efficiently acquire interpretable weak supervision sources for text, to learn highly discriminative text classifiers only from descriptions of classes, without any human-labeled documents.[2]

- We use `KeyClass` to reliably assign ICD-9 codes to patient discharge notes with no labeled documents and minimal human effort. Experiments on the publicly available `MIMIC-III` dataset reveal that `KeyClass` performs comparably to a robust supervised alternative (Venkataraman et al., 2020) trained using several thousand manually annotated clinical notes.

- We conduct experiments on 4 other common multiclass text classification datasets to benchmark `KeyClass` against previously proposed weakly supervised methods. Re-

1. ICD-10 version was released in 1992; however, in this paper, we restrict our experiments to assigning ICD-9 codes to better evaluate the performance of our proposed methods vis-à-vis prior work.
2. The code for `KeyClass` will be made publicly available at https://github.com/autonlab/KeyClass.

sults reveal that our model efficiently and effectively creats text classifiers that outperform prior work.

- To the best of our knowledge, `KeyClass` is the first to employ data programming for classification in a multiclass multilabel setting. The ICD-9 assignment problem involves assigning *all* relevant codes to each clinical note.

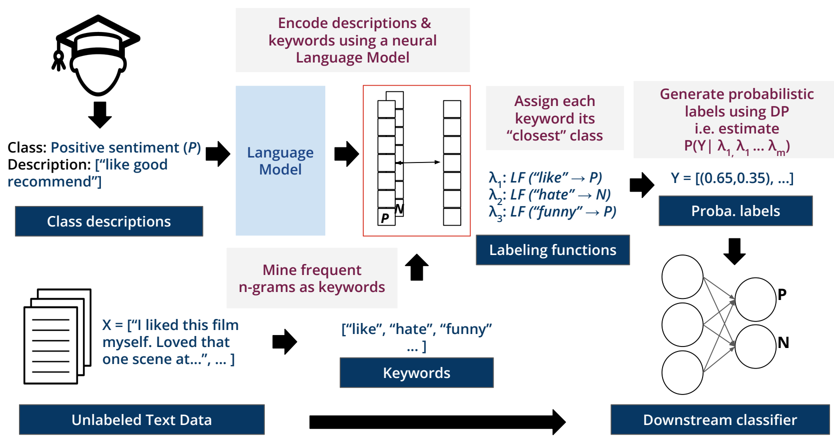

Figure 1: Overview of our methodology. From only class descriptions, `KeyClass` classifies documents without access to any labeled data. It automatically creates interpretable labeling functions (LFs) by extracting frequent *keywords* and *phrases* that are highly indicative of a particular class from the unlabeled text using a pre-trained language model. It then uses these LFs along with Data Programming (DP) to generate probabilistic labels for training data, which are used to train a downstream classifier (Ratner et al., 2016).

Generalizable Insights about Machine Learning in the Context of Healthcare

Managing costs and quality of healthcare is a persistent societal challenge of enormous magnitude and impact on daily lives of all people. Our work targets one very specific aspect of this complex landscape. Our approach proposes a low-cost solution that has the potential to address some of the identified pressing issues with accessibility to affordable yet accurate automated disease coding tools. Our contributions lie in using a novel strategy to efficiently acquire interpretable weak supervision sources from readily available text to learn effective text classifiers without the need for human-labeled data. Results on multiple datasets demonstrate that our method can outperform state-of-the-art baselines in realistic settings, and it can perform comparably to a fully supervised model in an important clinical problem. Our work demonstrates that (1) pre-trained language models can efficiently and effectively inform weakly supervised models for text classification, (2) self-training improves downstream classifier performance, especially when classifiers are initially trained on a subset of the training data, (3) data programming performs on par with simple majority vote when relying on a large number of automatically generated weak supervision sources of similar quality, and (4) key words are excellent sources of weak supervision.

2. Prior Work

2.1. Assigning ICD codes to Clinical Notes

ICD code assignment is hard due to the multiclass and multilabel nature of the problem. ICD-9 for instance defines more than 14,000 unique codes for nuanced classification of diseases, symptoms, abnormal findings, etc. Recent revisions of ICD have a much greater number of codes permitting the classification of new and previously known conditions with higher precision and finer granularity (WHO, 1988). Moreover, patients may be assigned more than one code depending on their diagnoses. Most studies to date tackle the first problem by either classifying a subset of codes or by grouping them based on the first three characters of four-five character codes. For our study, we follow Venkataraman et al. (2020)'s approach and classify general diagnostic categories, of which there are 19.

Numerous studies have used ML to tackle the problem of assigning ICD codes to unstructured clinical text. For example, Baumel et al. (2018) used hierarchical attention bidirectional Gated Recurrent Units (GRU) to tag discharge summaries by identifying sentences related to each label in a hierarchical manner. Outside of the English language domain, Yu et al. (2019) investigated the use of a similar hierarchical attention Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) to assign ICD-10 codes to clinical admission records in Chinese. Some studies have also explored the use multimodal features to improve tagging performance, instead of simply relying on the unstructured text. For example, Xu et al. (2019) proposed an ensemble of modality specific models to predict ICD-10 diagnostic codes. Specifically, they applied a Convolutional Neural Network (Text-CNN) to unstructured free text, an LSTM to semi-structured diagnosis descriptions, and a decision tree to process tabular patient data such as prescriptions, lab and microbiology events. Recently, Venkataraman et al. (2020) proposed `FasTag`, a fully supervised LSTM model which achieved state-of-the-art performance on classifying unstructured patient discharge notes into top-level ICD-9 categories. In addition to ML, some studies have also used information retrieval techniques to support human experts performing tagging. Rizzo et al. (2015), for instance, used transfer learning to expand a skewed dataset, while retrieving the top-K relevant codes and passing them to a human expert to improve tagging accuracy.

Another issue in ICD code assignment is that of low-support labels. As can be seen in Table 4, ICD codes vary significantly in terms of their frequency in data. For example, as many as 70% of the documents contain *supplementary* ICD codes, whereas only 0.003% of the documents are assigned *pregnancy or childbirth complication* codes. To combat this issue, Chapman and Neumann (2020) used label descriptions to improve their model's performance on the least represented ICD for Oncology (version 3) (ICD-O-3) codes. They utilized a Bidirectional Encoder Representations from Transformers (BERT)-like (Devlin et al., 2018) encoder and a word-level attention mechanism between input clinical text and textual descriptions of labels, using the output to a model with a customized loss function that favors recall. Through this method, they were able to consistently produce more varied ICD code predictions, while assigning fewer codes to each clinical text and maintaining a high recall and a competitive F1 score. In the rest of the paper, we restrict ourselves to assigning *high-level* ICD-9 codes to discharge notes in the publicly available `MIMIC-III` database following the same experimental settings as `FasTag` (Venkataraman et al., 2020). Our results (Table 4) reveal that even state-of-the-art ICD-code classification methods such as `KeyClass` and `FasTag` have a hard time predicting low-support categories, which may benefit from further research on classification under high class imbalance.

To the best of our knowledge, all prior work on ICD code assignment utilized fully supervised ML techniques, most of them relying on vast quantities of labeled training data. In this work, we explore the use of our proposed weakly supervised model `KeyClass` to assign top-level ICD-9 codes to long patient discharge summaries. Its training signal is retrieved automatically from readily available descriptions of the ICD codes, therefore it requires no human-produced supervisory feedback to build effective downstream text classifiers.

2.2. Text Classification with Sparse Training Labels

Weakly supervised text classification aims to classify text documents using cheaper albeit potentially noisier sources of supervision such as keywords. The earliest attempts at weak forms of supervision involved mapping documents and label names to Wikipedia concepts in a semantic space. The semantic relatedness between the labels and the documents are then used to classify text documents (Gabrilovich et al., 2007). Since these methods do not use any domain specific unlabeled data, relying purely on general knowledge, they are often referred to as `Dataless` techniques in the literature. Another class of methods use neural models to either generate psuedo documents or detect category indicative words in documents. For instance, the `WeSTClass` model generates pseudo-documents to pre-train a text classifier followed by self-training on labeled data for model refinement (Meng et al., 2018). More recently, Meng et al. (2020) proposed `LOTClass`, which associates semantically related words to label names and finds the implied category of words via masked category pre diction. Finally, their model self-trains itself on unlabeled documents to improve generalization.

Inspired by Meng et al. (2018, 2020), `KeyClass` is self-trained on unlabeled training documents using its own highly confident predictions. However, `KeyClass` differs from prior work in some fundamental ways. First, the foundation of our weak supervision methodology, i.e., frequent keywords and phrases as LFs, is highly interpretable. Secondly, while previously proposed state-of-the-art models are committed to specific language model architectures for linguistic knowledge and representation learning, `KeyClass` offers a high degree of modularity, enabling end users to adapt the neural language model (encoder) and downstream classifiers to specific problems, such as clinical text classification. Finally, we explore the use of weak supervision for multilabel multiclass classification, a problem which, to the best of our knowledge, has not been tackled by prior work on weak text classification.

2.3. Weak Supervision for Clinical Text Classification

Recently, weak supervision has also found use in clinical text classification. For example, Wang et al. (2019) developed a manually annotated, rule-based algorithm combined with data programming (Ratner et al., 2016) to create weak labels for smoking status, and proximal femur (hip) fracture classification. They used pre-trained word embeddings as deep representation features to train simple ML models for classification. Similarly, Cusick et al. (2021) trained weakly supervised models to detect suicidal ideation from unstructured clinical notes using rule-based labeling functions.

Thus, prior work on weakly supervised clinical text classification had an explicit dependence on *manually* created rule-based labeling functions. In this work, however, we demonstrate that we can quickly and automatically create simple keyword based labeling functions, with minimal to no human involvement.

3. Problem Formulation

Given a collection of n documents $\mathcal{D} = \{d_1\}_{i=1\ldots n}$, c class labels $\mathcal{C} = \{c_i\}_{i=1\ldots c}$, and their descriptions $\mathcal{E} = \{e_i\}_{i=1\ldots c}$, our goal is to first "probabilistically" label each document d_i using a label model $\mathcal{L}_\theta$ and then use these labels to train a downstream classifier $\mathcal{M}_\phi$ to assign all relevant class labels $c_i \in \mathcal{S}_i$ to each document $d_i \in \mathcal{D}$, where $\mathcal{S}_i$ is a set of classes and $\mathcal{S}_i \subseteq \mathcal{C}$. Furthermore, for our experiments, with the exception of ICD-9 code assignment, all problems are single-label multiclass in nature ($\mathcal{S}_i$ is a singleton set).

The label model $\mathcal{L}$ parameterized by θ relies on a set of m *labeling functions* (LF) denoted by $\Lambda = \{\lambda_i\}_{i=1\ldots m}$, where each LF $\lambda_i : \mathcal{D} \to \mathcal{S} \subseteq \mathcal{C}$, assigns a label $\hat{p}(c_j \in \mathcal{C} \mid \Lambda)$, to each document $d_i \in \mathcal{D}$. Note that each LF only votes for a single class. In this work, we constrain the set of labeling functions to be simple *keyword-matching rules* of the form:

$$\texttt{If } k_i \texttt{ occurs in } d_j \texttt{ then vote } c_k \texttt{ else abstain} \tag{1}$$

where d_j is the j^{th} document, c_k is the k^{th} class, and k_i belongs to a set of keywords or key-phrases $\mathcal{K}$ automatically mined from $\mathcal{D}$ (See Figure 3).

In the following section, we will review data programming methodology used in `KeyClass` to generate labels from the keyword-matching rules mentioned previously.

3.1. Data Programming for Weak Text Classification

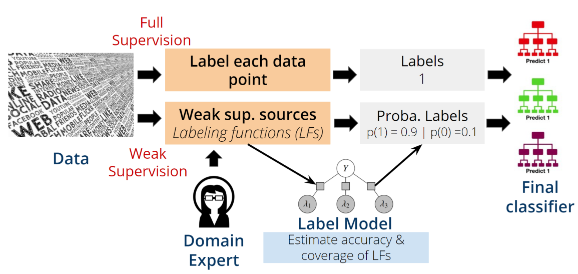

Figure 2: Data programming, or weak supervision compared to fully supervised ML. The orange boxes indicate the effort required by expert annotators. Instead of having to label extensive quantities of data by hand, the effort in data programming framework lies in obtaining labeling functions. In `KeyClass`, these labeling functions are our keyword-matching rules automatically extracted from reference data, to further reduce required human effort.

The label model $\mathcal{L}_\theta$ assumes that each document is associated with an unobserved true class label $c_i \in \mathcal{C}$. Note that Ratner et al. (2016) defined the label model for multiclass but single-label classification. While this assumption holds for all our baseline datasets, it is not true for ICD-9 code assignment which involves multilabel classification. Hence, more generally, we assume that each document is associated with an unobserved set of labels, $\mathcal{S}^* \subseteq \mathcal{C}$, where $\mathcal{S}^*$ is a singleton set for single-label classification problems.

To model the multilabel nature of the ICD code assignment, we further assume each document d_i to be characterized by an *unobserved* true probability distribution p_i^* over the set of possible categories $\mathcal{C}$, an assumption which is consistent with literature on topic modelling (Blei et al., 2003). Specifically, p_i^* is a categorical distribution over all the categories $\mathcal{C}$, such that $p_i^*(c_j)$ is the probability that document d_i belongs to class c_j. In a fully supervised multilabel classification, we would expect documents to be tagged with categories over which p^* places a large probability mass.

The goal of the label model is then to label each document with $\hat{p}_i(c_j \mid \Lambda)$ given the votes of a set of m labeling functions (LFs), $\Lambda = \{\lambda_i\}_{i=1\ldots m}$. For simplicity, the label model introduced by Ratner et al. (2016) assumes that all LFs are independent given the true class label, and that they vote with better than random accuracy where they do not abstain. However, the LFs do not need to have perfect accuracy and may conflict with one another. We also assert that each LF only votes for a particular class by construction, i.e., we define each LF as $\lambda : \mathcal{D} \rightarrow c_i \in \mathcal{C}$. This allows us to use the same label model to estimate the accuracies and coverage of LFs using their agreements and disagreements via a factor graph, which is then used to infer a document's probabilistic label $\hat{p}_i(c_j \mid \Lambda)$, which is close to the true categorical distribution p_i^* under settings enumerated in Ratner et al. (2016), i.e.:

$$\forall d_i \in \mathcal{D},\ \hat{p}_i(c_j \mid \Lambda) \approx p_i^*(c_j)$$

Let Λ denote the $n \times m$ dimensional matrix of LF votes. In order to learn $\hat{p}(c_j \mid \Lambda)$, we first define a factor for LF accuracy as $\phi^{Acc}(\Lambda_{ij}, c_i) \triangleq \mathbb{1}\{\Lambda_{ij} = c_i\}$ as well as a factor of LF propensity as $\phi^{Lab}(\Lambda_{ij}, c_i) \triangleq \mathbb{1}\{\Lambda_{ij} \neq 0\}$. Following Ratner et al. (2016), we define the model of the joint distribution of Λ and C as:

$$p_\theta(\Lambda, C) = \frac{1}{Z_\theta} \exp\left(\sum_{j=1}^{m}\sum_{i=1}^{n}\left(\theta_j \phi^{Acc}(\Lambda_{ij}, c_i) + \theta_{j+m}\phi^{Lab}(\Lambda_{ij}, c_i)\right)\right)$$

where Z_θ is a normalizing constant and θ are the canonical parameters for the LF accuracy and propensity. We use Snorkel (Ratner et al., 2017) to learn θ by minimizing the negative log marginal likelihood given the observed Λ. Finally, we train a downstream classifier $\mathcal{M}_\phi$ with a noise aware loss function using the estimated probabilistic labels $\hat{p}_i(c|\Lambda)\}$.

4. Methodology

Find Class Descriptions Figure 1 presents an overview of our proposed method. Unlike traditional supervised learning where each document needs to be labeled `KeyClass` only relies on meaningful and succinct class descriptions. This also removes the requirement of expert heuristics as in prior weak supervision work. As a concrete example, consider the `IMDb` movie review sentiment classification problem, where the objective is to classify a movie review as being "*positive*" or "*negative*". In order to initiate the classification process, domain experts provide `KeyClass` with common sense descriptions of a positive (`"good amazing exciting positive"`) and negative review (`"terrible bad boring negative"`). Figure 3 presents two example keyword labeling functions for the `IMDb` dataset.

```
def masterpiece_lf(review):
    return POSITIVE if 'masterpiece' in review else ABSTAIN

def horrible_lf(review):
    return NEGATIVE if 'horrible' in review else ABSTAIN
```

Figure 3: Example code of keyword labeling functions for the `IMDb` dataset. In practice, this is done automatically and implicitly through our pipeline.

In most cases, these descriptions can be automatically generated from Wikipedia articles or reference manuals and validated by domain experts, further reducing manual effort. For instance, for the ICD-9 code assignment problem, we can automatically acquire descriptions of all categories by mining the most frequently occurring words (minus stop words) from the combined text descriptions of all the codes (from CMS.gov[3]) that fall into each of our 19 categories. Table 1 shows examples of the automatically curated descriptions of 2 high-level ICD-9 categories. Our approach overcomes a primary drawback of prior weak supervision methods, which rely on natural language rules manually crafted by domain experts.

Category	Description
Respiratory system diseases	due pneumonia acute chronic respiratory influenza pulmonary lung virus asthma sinusitis bronchitis larynx classified diseases obstruction elsewhere manifestations without identified pneumonitis ...
Genitourinary system diseases	specified chronic lesion female kidney acute glomerulonephritis disorders genital urinary cervix prostate breast ...

Table 1: Example descriptions for two ICD-9 categories. These descriptions were mined using the official descriptions of all the ICD-9 codes that fell into the ranges defined by each category (WHO, 1988).

Find Relevant Keywords Once we have the class descriptions, `KeyClass` automatically discovers highly suggestive *keywords* and *phrases* for each class. Keywords have been shown to be excellent sources of weak supervision (Boecking and Dubrawski, 2019; Ratner et al., 2016). `KeyClass` first obtains frequent n-grams from the training corpus to serve as keywords or key-phrases for its automatically composed labeling functions. Let us denote the set of all keywords and key-phrases as $\mathcal{K}$. In our implementation, we used the `CountVectorizer` function from `scikit-learn` (Buitinck et al., 2013) to return $\{1,2,3\}$-grams having a document frequency strictly greater than 0.001. We post-process the n-grams by removing common English stop-words from a corpus defined in the Natural Language Toolkit (NLTK) (Loper and Bird, 2002).

In order to transform the keywords into labeling functions of the prescribed form, `KeyClass` leverage the general linguistic knowledge stored within pre-trained neural language models such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), to map each keyword to the most semantically related category description. If we denote the neural language model encoder as $\mathbb{E} : \texttt{string} \rightarrow \mathbb{R}^d$, then at the end of this step `KeyClass` obtains two sets of equi-dimensional embeddings, $\{\mathbb{E}(k_i)\}_{i=1\ldots|\mathcal{K}|}$ and $\{\mathbb{E}(e_i)\}_{i=1\ldots c}$ corresponding to the keywords and class descriptions, respectively. Thus,

3. www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes

to create a labeling function, KeyClass simply assigns a keyword to its closest category as measured by the cosine similarity between their embeddings. Thus, the keyword k_i is assigned to its closest category c_j, when $j = \arg\min_{\forall k \in c} d_{\texttt{cosine}}\left(\mathbb{E}(k_i), \mathbb{E}(e_k)\right)$. The corresponding labeling function can be then denoted as shown in Equation 1. That is, if a document contains the keyword k_i then, this labeling function votes for class c_j, otherwise it abstains from voting for any particular class.

For all our baseline experiments, we used MPNET (Song et al., 2020) to encode both the keywords/key-phrases and class descriptions into 768-dimensional embeddings using the paraphrase-mpnet-base-v2 implementation in the sentence-BERT (Reimers and Gurevych, 2019) Python package. For ICD-9 code assignment, we instead used BlueBERT (Peng et al., 2019), a BERT model trained specifically on clinical text databases, since it is better suited for clinical text classification problems such as ours. This modularity differentiates our method from previous methods such as LOTClass which are based on the specific neural language architectures (such as BERT) and hence inflexible to special problem domains.

In order to ensure equal representation of all classes, KeyClass sub-samples the top-k labeling functions per class, ordering them by cosine similarity. While theoretically data programming benefits from as many labeling functions as possible, the sampling is required due to computational and space constraints. For example, for the AMAZON and *IMDb* datasets, we choose the top-300 labeling functions, whereas for DBPEDIA which has 14 classes, we choose the top 15 only.

Probabilistically Label Data Next, KeyClass constructs the labeling function vote matrix Λ and generates the probabilistic labels $\hat{p}(c_i \mid \Lambda)$ for all training documents using the label model $\mathcal{L}_\theta$ described earlier. Specifically, we use the open-source label model implementation of the Snorkel Python library released by Ratner et al. (2016).

Train Downstream Text Classifier After obtaining a probabilistically labeled training dataset, KeyClass can train any downstream classifier using rich document feature representations provided by the neural language model $\mathbb{E}$. Instead of using all the automatically labeled documents, KeyClass initially trains the downstream classifier using top-k documents with the most confident label estimates only.

Finally, KeyClass self-trains the downstream model-encoder combination on the entire training dataset to refine the end model classifier. The primary idea of self-training is to iteratively use the model's current predictions $\mathbb{P}$ to generate a target distribution $\mathbb{Q}$ which can guide the model refinement using the following KL-divergence loss:

$$\mathcal{L}_{ST} = \mathbf{KL}(\mathbb{Q}||\mathbb{P}) = \sum_{i=1}^{n}\sum_{j=1}^{c} q_{ij} \log \frac{q_{ij}}{p_{ij}} \tag{2}$$

where p_{ij} is the predicted probability that the i^{th} training sample belongs to the j^{th} class. In order to compute the target distribution $\mathbb{Q}$, KeyClass applies soft-labeling which makes high confidence predictions more confident, and low confidence predictions less so, by squaring and normalizing the current predictive distribution $\mathbb{P}$ (Xie et al., 2016). More formally,

$$q_{ij} = \frac{p_{ij}^2/f_j}{\sum_{j'}^{K}(p_{j'}/f_{j'})},\ \ f_j = \sum_{i}^{N} p_{ij} \tag{3}$$

5. Experiments

5.1. Multilabel ICD-9 Code Category Classification

In order to evaluate `KeyClass` on its ability to assign top-level diagnostic codes, we used free-text discharge summaries and corresponding ICD-9 codes recorded in the Medical Information Mart for Intensive Care (`MIMIC-III`) dataset (Johnson et al., 2016). `MIMIC-III` is a large publicly available single-center dataset comprising of de-identified clinical data of over $40,000$ patients admitted to the critical care units of the Beth Israel Deaconess Medical Center at Harvard Medical School between 2001 and 2012. For a faithful comparison of `KeyClass` with `FasTag`, we used the same $70 : 30$ train-test split and 19 top-level ICD-9 categories used by Venkataraman et al. (2020).

Since this is a multiclass multilabel problem, we encode our target variable as 19-dimensional one-hot vectors, with a 1 corresponding to every diagnosis of a patient. While `KeyClass` does not require input text to be pre-processed, for consistency in comparing our model to `FasTag`, we follow Venkataraman et al. (2020)'s pre-processing by keeping only the most potentially useful for discrimination parts of text in each patient discharge note as ranked by the term frequency - inverse document frequency (TF-IDF) statistic.

To compare our model against the supervised LSTM model in `FasTag`, we compute both aggregate `precision`, `recall`, and `F1` scores and category-specific `F1` scores as well as their confidences [4].

5.2. General Weak Text Classification Performance of `KeyClass` Compared to Baselines

Dataset	Classification Type	# Classes	# Train	# Test
`AGNews`	News topics	4	120,000	7,600
`DBPedia`	Wikipedia Categories	14	560,000	70,000
`IMDb`	Movie Reviews	2	25,000	25,000
`AMAZON`	Amazon Reviews	2	3,600,000	400,000
`MIMIC-III`	Clinical diagnostic categories	19	39,541	13,181

Table 2: Dataset Statistics. All models are trained on the training set, but weakly supervised models do not have access to labels. Unlike other datasets, `MIMIC-III` is a multilabel multiclass classification problem where each clinical note must be assigned to all relevant categories. To best compare our results with prior work, we follow the same train and test splits as Meng et al. (2020) for the `AGNews`, `DBPedia`, `IMDb`, and `AMAZON` datasets. Similarly, for `MIMIC-III` we use the same train and test data as Venkataraman et al. (2020). All datasets except `MIMIC-III` are balanced.

We also compared `KeyClass` with previously proposed state-of-the art weakly supervised models (`Dataless` (Chang et al., 2008), `WeSTClass` (Meng et al., 2018) and `LOTClass` (Meng et al., 2020)) and BERT-based fully-supervised (Devlin et al., 2019) models on four

4. Specifically, we compute the `precision`, `recall` and `F1` scores for each instance and average across all test samples.

real-world text classification problems. Since these previously proposed weakly supervised models were not tested on multilabel classification, we restricted our experiments to the following single-label multiclass problems: (1) movie review sentiment classification on `IMDb` (Maas et al., 2011) and `AMAZON` (McAuley and Leskovec, 2013), (2) news topic classification on `AGNEWS` (Zhang et al., 2015), and (3) Wikipedia article classification on `DBPEDIA` (Lehmann et al., 2015). We used the same train-test splits as prior work (see Table 2) and report the accuracy accordingly. We also conducted ablation experiments to evaluate the impact of self-training and using data programming to probabilistically label the training data.

We used a combination of a neural encoder and a 4-layer MLP with LeakyReLU activations (Maas et al., 2013) as our downstream classifier. Each linear layer was followed by a dropout layer with 0.5 dropout probability (Srivastava et al., 2014). To train the multilabel downstream classifier for ICD code assignment, we used the binary cross-entropy with logits loss. Cross-entropy loss was used to train classifiers for the remaining datasets. We trained each model with a batch size of 128 for a maximum of 20 epochs, allowing for early stopping with a patience of 2. We used Adam optimizer with learning rate of 0.001[5]. All models were built and trained using PyTorch 1.8.1 (Paszke et al., 2019) using Python 3.8.1. Experiments were carried out on a computing cluster, with a typical machine having 40 Intel Xeon Silver 4210 CPUs, 187 GB of RAM, and 4 NVIDIA RTX2080 GPUs.

6. Results and Discussion

6.1. `KeyClass` Assigns ICD-9 Codes Accurately.

Supervision Type	Methods	Recall	Precision	F1
Weakly sup.	`FasTag` (Venkataraman et al., 2020)	0.734 ± 0.00138	0.436 ± 0.00144	0.525 ± 0.00133
Weakly sup.	`KeyClass` (Ours)	$\mathbf{0.896 \pm .0009}$	$\mathbf{0.507 \pm .0016}$	$\mathbf{0.6252 \pm 0.0014}$
Fully sup.	`FasTag` (Venkataraman et al., 2020)	0.671 ± 0.0019	0.753 ± 0.00171	0.678 ± 0.00141

Table 3: `KeyClass` performs on par with the fully supervised baseline `FasTag` (Venkataramam et al.) on the `MIMIC-III` ICD-9 code assignment problem. We also report the performance of `FasTag` when trained using our probabilistic labels (weakly sup. `FasTag`). The superior performance of weakly supervised `KeyClass` over its `FasTag` counterpart is primarily due to better text modeling capabilities of `BlueBert` due to its relevant architecture and pre-training. The results are reported with 95% bootstrap confidence intervals.

Tables 3 and 4 compare the performance of `FasTag` and `KeyClass` on the top-level ICD-9 code assignment problem. Remarkably, fully supervised `FasTag` achieves only 5 points in F1 score over `KeyClass`, trained without any access to pointillistic labels or hand-coded natural language rules, with only minimal human intervention.

5. *Justification of Modeling Decisions*: We consciously did not invest effort into optimizing hyperparameters to not obfuscate the presentation of our core idea. But indeed, there is a good chance that the model could be further improved with manual or automated optimization.

Category Name	Model performance		
	Prevalence	KeyClass	Fastag
Infectious & parasitic	0.255	0.488	**0.608**
Neoplasms	0.157	0.031	**0.656**
Endocrine, nutritional and metabolic	0.626	0.855	**0.862**
Blood & blood-forming organs	0.341	**0.591**	0.559
Mental disorders	0.278	**0.529**	0.384
Nervous system	0.232	0.329	**0.499**
Sense organs	0.068	**0.004**	0.002
Circulatory system	0.760	0.922	**0.936**
Respiratory system	0.447	0.688	**0.709**
Digestive system	0.370	0.610	**0.657**
Genitourinary system	0.378	0.648	**0.728**
Pregnancy & childbirth complications	0.003	0.000	0.000
Skin & subcutaneous tissue	0.107	0.004	**0.090**
Musculoskeletal system & connective tissue	0.170	**0.080**	0.050
Congenital anomalies	0.059	0.018	**0.048**
Perinatal period conditions	0.093	0.000	**0.971**
Injury and poisoning	0.347	**0.608**	0.601
External causes of injury	0.408	**0.607**	–
Supplementary	0.685	**0.830**	–

Table 4: Performance comparison of `KeyClass` and `FasTag` disaggregated by ICD-9 categories, reveal that for some categories such as *mental disorders* and *injury and poisoning*, our model outperformed the fully supervised baseline. Both models had a hard time predicting low-support categories, i.e., categories with low prevalence in the data. Results for the last two categories were unavailable for `FasTag` (Venkataraman et al., 2020).

Table 4 reports F1 scores for each of the top level ICD-9 categories. Surprisingly, `KeyClass` outperformed `FasTag` on some categories such as *mental disorders* and *injury and poisoning*. We observed variance in the performance of `KeyClass` and `FasTag` across the categories. In fact, for some classes with high representation in the data, both models do well. On the contrary, the models report much lower F1 scores for less frequent classes. We believe that further research is required to extensively analyze these lesser represented categories to ensure a high quality automated annotation system.

6.2. `KeyClass` Outperforms Advanced Weakly Supervised Models.

Supervision Type	Methods	AG News	DBPedia	IMDb	Amazon
Weakly sup.	`Dataless` (Chang et al., 2008)	0.696	0.634	0.505	0.501
	`WeSTClass` (Meng et al., 2018)	0.823	0.811	0.774	0.753
	`LOTClass` (Meng et al., 2020)	0.864	0.911	0.865	0.916
	`KeyClass` (Ours)	**0.869** $\pm$ **0.004**	**0.940** $\pm$ **0.001**	**0.871** $\pm$ **0.002**	**0.928** $\pm$ **0.000**
Fully sup.	`BERT` (Devlin et al., 2019)	0.944	0.993	0.945	0.972

Table 5: Classification Accuracy. `KeyClass` outperforms state-of-the-art weakly supervised methods on 4 real-world text classification datasets. We report our model's accuracy with a 95% bootstrap confidence intervals. Results for `Dataless`, `WeSTClass`, `LOTClass`, and `BERT` are reported from Meng et al. (2020).

Experiments on the AGNEWS, DBPedia, IMDb and AMAZON datasets reveal that KeyClass outperforms state-of-the-art weakly supervised models in terms of accuracy. Our model trained without access to any ground truth labels trails fully supervised BERT by less than 10 percentage points (Table 5).

6.3. Ablation Experiments: Self-training helps, Majority Vote is a strong baseline

Table 6 reports the results of our ablation experiments. Consistent with prior work, we observed varying degrees of improvement in downstream model performance from self-training. Self-training the downstream classifier improves its generalisation beyond the initial hypothesis learned from the top-k most confidently labeled documents. We also found that taking the majority vote of labeling functions performs on par with data programming. In fact, on Amazon and IMDb majority vote outperforms data programming. This is most likely since KeyClass automatically creates a sufficiently large number labeling functions of approximately the same accuracy. On the other hand, in practice, data programming shines when there are few labeling functions with vastly different accuracies (Goswami et al., 2021).

Methods	AG News	DBPedia	IMDb	Amazon
LOTClass w/o. self train (Meng et al., 2020)	0.822	0.860	0.802	0.853
LOTClass (Meng et al., 2020)	0.864	0.911	0.865	0.916
KeyClass w/o. self train	0.841 ± 0.004	0.823 ± 0.002	0.836 ± 0.0019	0.832 ± 0.001
KeyClass (Ours)	$\mathbf{0.867 \pm 0.004}$	$\mathbf{0.951 \pm 0.001}$	$\mathbf{0.895 \pm 0.002}$	$\mathbf{0.941 \pm 0.00}$
Label Model (Data Programming)	**0.731**	**0.638**	0.699	0.580
Label Model (Majority Vote)	0.694	0.630	**0.717**	**0.652**

Table 6: Classification Accuracy for Ablation Experiments. Consistent with prior work, self-training improves downstream model performance. Data Programming performs on par compared to majority vote in probabilistically labeling the training data. Label Model accuracies are reported on the training set, whereas the rest of the results are reported on the test set. Results for LOTClass are reported from Meng et al. (2020).

6.4. Keywords are Excellent Sources of Weak Supervision.

Another finding of our study is that keywords and key-phrases are excellent sources of weak supervision. For more complex problems, it may be necessary for experts to manually re-assign some keywords to different categories. However, obtaining supervision at the keyword/key-phrase level is still much more efficient than labeling the entire corpus as any potentially required manual effort in our approach is upper-bounded by the size of the frequent terms vocabulary which is usually much smaller than size of the text corpus.

6.5. Limitations and Future Work.

A limitation of KeyClass is that the automatic labeling function creation capability has not been tested on more complex problems. Moreover, our results on the ICD-9 code assignment problem show room for improvement, especially in classifying low-resource categories. We believe that future work should focus on testing KeyClass on a wider range of complex real world text classification problems, and developing techniques to improve the performance of text classifiers on low support categories.

One should take the labels predicted by KeyClass to be general areas where the clinical note is classified, not as definitive ground-truth–a human-in-the-loop situation would be best for practical applications. It would be interesting to see if our method can be useful for broadly checking the correctness of the already assigned ICD-9 labels in legacy results or reports, since their original hand labeling could yield errors. The resulting auditing tool could be beneficial for both the healthcare providers and insurers to improve accuracy of diagnostic coding, to reduce the risks of negative impact of such errors on quality of care and patient outcomes, and to mitigate the financial risks caused by coding errors in health insurance claims and reimbursement practices.

6.6. Conclusion

Healthcare providers record detailed notes of clinical care delivered to each patient for clinical, research and billing purposes. Due to the unstructured nature of these narratives, providers employ dedicated staff to assign diagnostic codes to patients' diagnoses using the International Classification of Diseases (ICD) coding system. This manual process is, time-consuming, costly, and error-prone.

The challenges are exacerbated by somewhat frequent revisions of the coding systems and their customization for use by particular healthcare organizations, so the currency and universality of the manual coding protocols are difficult to attain in practice, building up the costs and hassle. These challenges also limit practical utility of the existing, primary fully supervised machine learning approaches that rely on availability of substantial amounts of reference data to train reliable models for automated coding purposes.

To address this issue, we propose KeyClass, a general weak text classification model and a novel strategy to efficiently acquire interpretable weak supervision sources. KeyClass quickly and automatically creates highly interpretable heuristics based on keywords sourced from reference data, and enables end users to adapt its components to specific problems through support of domain specific language models. In contrast, previously proposed weakly supervised methods either rely on manually created heuristics, were uninterpretable due lack of transparency in the pseuo-labeling process, or were highly inflexible due their commitment to specific model architectures.

We successfully applied KeyClass to reliably assign ICD-9 codes over a large public dataset comprising of several thousand physician notes. We compared its performance with a state-of-the-art fully supervised model. We also found that KeyClass performs comparably, and for some code categories, even better than a supervised model trained using several thousand labeled clinical notes. Additional experiments on four standard NLP multiclass text classification problems confirm our proposed model's competitive position compared to previous methods. Although further research is necessary to comprehensively validate the proposed method across a wider range of complex data and use cases, KeyClass's impressive performance on a challenging problem which plagues the healthcare industry, demonstrates its potential in helping broaden the adoption of beneficial machine learning technology in multiple application domains.

Acknowledgments

This work was partially supported by a fellowship from Carnegie Mellon University's Center for Machine Learning and Health to M.G. The authors would also like to thank the anonymous reviewers and the program committee for their insightful feedback.

References

Julia Adler-Milstein, A Jay Holmgren, Peter Kralovec, Chantal Worzala, Talisha Searcy, and Vaishali Patel. Electronic health record adoption in us hospitals: the emergence of a digital "advanced use" divide. *Journal of the American Medical Informatics Association*, 24(6):1142–1148, 2017.

Musaed Ali Alharbi, Godfrey Isouard, and Barry Tolchard. Historical development of the statistical classification of causes of death and diseases. *Cogent Medicine*, 8(1): 1893422, 2021. doi: 10.1080/2331205X.2021.1893422. URL https://doi.org/10.1080/2331205X.2021.1893422.

Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie Elhadad. Multi-label classification of patient notes: case study on icd code assignment. In *Workshops at the thirty-second AAAI conference on artificial intelligence*, 2018.

Curtis Benesch, DM Witter, AL Wilder, PW Duncan, GP Samsa, and DB Matchar. Inaccuracy of the international classification of diseases (icd-9-cm) in identifying the diagnosis of ischemic cerebrovascular disease. *Neurology*, 49(3):660–664, 1997.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

Benedikt Boecking and Artur Dubrawski. Pairwise feedback for data programming. *arXiv preprint arXiv:1912.07685*, 2019.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835, 2008.

Kathryn Annette Chapman and Günter Neumann. Automatic icd code classification with label description attention mechanism. In *IberLEF@ SEPLN*, pages 477–488, 2020.

Marika Cusick, Prakash Adekkanattu, Thomas R Campion Jr, Evan T Sholle, Annie Myers, Samprit Banerjee, George Alexopoulos, Yanshan Wang, and Jyotishman Pathak. Using weak supervision and deep learning to classify clinical notes for identification of current suicidal ideation. *Journal of psychiatric research*, 136:95–102, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Evgeniy Gabrilovich, Shaul Markovitch, et al. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611, 2007.

Mononito Goswami, Benedikt Boecking, and Artur Dubrawski. Weak supervision for affordable modeling of electrocardiogram data. In *AMIA Annual Symposium Proceedings*, volume 2021, page 536. American Medical Informatics Association, 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

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. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172, 2013.

Nir Menachemi and Taleah H Collum. Benefits and drawbacks of electronic health record systems. *Risk management and healthcare policy*, 4:47, 2011.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 983–992, 2018.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65, 2019.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access, 2017.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in neural information processing systems*, pages 3567–3575, 2016.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Stefano Giovanni Rizzo, Danilo Montesi, Andrea Fabbri, and Giulio Marchesini. Icd code retrieval: Novel approach for assisted disease classification. In *International Conference on Data Integration in the Life Sciences*, pages 147–161. Springer, 2015.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Guhan Ram Venkataraman, Arturo Lopez Pineda, Oliver J Bear Don't Walk IV, Ashley M Zehnder, Sandeep Ayyar, Rodney L Page, Carlos D Bustamante, and Manuel A Rivas. Fastag: Automatic text classification of unstructured medical narratives. *PLoS one*, 15 (6):e0234647, 2020.

Yanshan Wang, Sunghwan Sohn, Sijia Liu, Feichen Shen, Liwei Wang, Elizabeth J Atkinson, Shreyasee Amin, and Hongfang Liu. A clinical text classification paradigm using weak supervision and deep representation. *BMC medical informatics and decision making*, 19 (1):1–13, 2019.

WHO. International classification of diseases—ninth revision (icd-9). *Weekly Epidemiological Record = Relevé épidémiologique hebdomadaire*, 63(45):343–344, 1988.

Wikipedia contributors. International classification of diseases — Wikipedia, the free encyclopedia, 2021. URL https://en.wikipedia.org/wiki/International_Classification_of_Diseases. [Online; accessed 07-September-2021].

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

Keyang Xu, Mike Lam, Jingzhi Pang, Xin Gao, Charlotte Band, Piyush Mathur, Frank Papay, Ashish K Khanna, Jacek B Cywinski, Kamal Maheshwari, et al. Multimodal machine learning for automated icd coding. In *Machine Learning for Healthcare Conference*, pages 197–215. PMLR, 2019.

Ying Yu, Min Li, Liangliang Liu, Zhihui Fei, Fang-Xiang Wu, and Jianxin Wang. Automatic icd code assignment of chinese clinical notes based on multilayer attention birnn. *Journal of biomedical informatics*, 91:103114, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657, 2015.