A Neural Candidate-Selector Architecture for Automatic Structured Clinical Text Annotation

Gaurav Singh
University College London
London, UK
gaurav.singh.15@ucl.ac.uk

Iain J. Marshall
King's College London
London, UK
iain.marshall@kcl.ac.uk

James Thomas
University College London
London, UK
james.thomas@ucl.ac.uk

John Shawe-Taylor
University College London
London, UK
j.shawe-taylor@ucl.ac.uk

Byron C. Wallace
Northeastern University
Boston, MA
b.wallace@northeastern.edu

ABSTRACT

We consider the task of automatically annotating free texts describing clinical trials with concepts from a controlled, structured medical vocabulary. Specifically, we aim to build a model to infer distinct sets of (ontological) concepts describing complementary clinically salient aspects of the underlying trials: the populations enrolled, the interventions administered and the outcomes measured, i.e., the *PICO* elements. This important practical problem poses a few key challenges. One issue is that the output space is vast, because the vocabulary comprises many unique concepts. Compounding this problem, annotated data in this domain is expensive to collect and hence sparse. Furthermore, the outputs (sets of concepts for each PICO element) are correlated: specific populations (e.g., diabetics) will render certain intervention concepts likely (insulin therapy) while effectively precluding others (radiation therapy). Such correlations should be exploited.

We propose a novel neural model that addresses these challenges. We introduce a Candidate-Selector architecture in which the model considers setes of *candidate concepts* for PICO elements, and assesses their plausibility conditioned on the input text to be annotated. This relies on a 'candidate set' generator, which may be learned or relies on heuristics. A conditional discriminative neural model then jointly selects candidate concepts, given the input text. We compare the predictive performance of our approach to strong baselines, and show that it outperforms them. Finally, we perform a qualitative evaluation of the generated annotations by asking domain experts to assess their quality.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Machine learning**; **Neural networks**; • **Applied computing** → **Health informatics**;

CIKM'17, November 6–10, 2017, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3132989

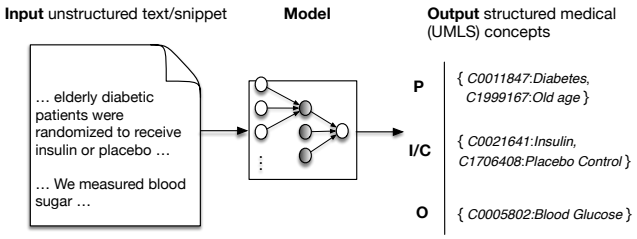


Figure 1: Illustration of the annotation task. The output comprises concepts drawn from the UMLS controlled medical vocabularly, grouped into terms that describe the study Population, Interventions/Comparators and Outcomes.

KEYWORDS

text mining; biomedical informatics; deep learning

ACM Reference format:
Gaurav Singh, Iain J. Marshall, James Thomas, John Shawe-Taylor, and Byron C. Wallace. 2017. A Neural Candidate-Selector Architecture for Automatic Structured Clinical Text Annotation. In *Proceedings of CIKM'17, Singapore, November 6–10, 2017,* 10 pages.
https://doi.org/10.1145/3132847.3132989

1 INTRODUCTION

There has been rapid growth in the volume and diversity of available healthcare data, ranging from electronic health records (EHRs) to biomedical literature. This proliferation of data provides unprecedented opportunity to improve patient care [6, 8, 9, 19], but simultaneously the volume of published information makes it difficult to efficiently retrieve and compile relevant evidence. In this work we focus on biomedical literature, and in particular on texts that describe the conduct and results of randomized controlled trials (RCTs), which are considered the gold standard in evidence for particular interventions.

In general, the clinically salient aspects of an RCT include: (1) the Population(s) enrolled; (2) the Intervention and Comparator treatments administered (the distinction between these is arbitrary, and so these may be grouped); (3) the Outcomes measured. Collectively these are referred to as PICO elements. Clinical questions are widely considered answerable only when mapped onto a PICO frame. However, retrieving all articles that describe trials relevant

to a given PICO frame (and hence question) is non-trivial, in part because reports of RCTs are communicated in unstructured (free-text) articles. Structured representations of articles that explicitly assign ontological terms to distinct PICO elements would support automated retrieval and question-answering systems [4]. We therefore aim to develop an automated approach to mapping from free-texts to distinct sets of terms from the Unified Medical Language System (UMLS) corresponding to each PICO element. This is depicted schematically in Figure 1.

This multilabel and multitask setting presents formidable challenges from a machine learning perspective. In particular, the output space is vast: there are hundreds of thousands of terms in the controlled medical vocabulary we are targeting (UMLS). Second, as is the case in many biomedical tasks, we have a relative dearth of available training data with which to estimate model parameters. Third, outputs (i.e., sets of UMLS terms corresponding to the respective PICO elements) are correlated: a given study population constrains the space of plausible interventions and outcomes. For example, if the population comprises *adult males*, it is unlikely that the outcome will be *time to labor induction*. These correlations between label outputs should be exploited. We address these problems in this paper by introducing a novel neural approach involving two parts: *candidate term generation* and *selection/classification*.

The specific contribution of this work is a novel method for multilabel classification into multiple distinct, but correlated label sets using a neural model that considers 'candidate' label tuples, conditioned on the text being annotated. Our approach addresses training data sparsity by re-framing the annotation task as a two step process in which we first generate a set of candidate annotations relevant to the input text, and then we select and group these. In our case, we generate candidates using both (a) a multitask model directly trained to generate candidate concepts, and, (b) the *MetaMap* tool.[1] We then use a neural discriminative model to infer plausible triplets of concepts from the unstructured candidate set, conditioned on the free-text being annotated. We demonstrate that this model improves performance (compared to relevant baselines) on the important task of automatically annotating biomedical literature with structured UMLS concepts. As far as we aware, this is the first work to tackle this challenging problem.

While our motivating application concerns biomedical literature processing, we emphasize that the problem we consider is general, and the candidate-generator/discriminator approach we propose may have broad application for similarly structured tasks.

2 METHODS

Our proposed approach comprises two components. The first is a *candidate generator*, responsible for inducing an unstructured set of 'candidate' UMLS concepts deemed likely to apply to a given input text. Ideally this would be a high-recall (but possibly low-precision) set of terms. The second component is a *selector*, which accepts the candidate concepts as input, along with the text to be annotated, and conditioned on these selects and outputs likely structured sets of concepts, i.e., concepts pertaining to the aforementioned PICO elements.

[1] https://metamap.nlm.nih.gov/

Formally, denote an input text by $\mathbf{x}$. Then we run through this our candidate generator, g:

$$C = g(\mathbf{x}) \tag{1}$$

and the outputs are consumed by the selector s:

$$\mathcal{Y} = s(C, \mathbf{x}) = s(g(\mathbf{x})). \tag{2}$$

Here $\mathcal{Y}$ is assumed to be structured, i.e., include particular concepts corresponding to the PICO elements. Thus $\mathcal{Y} = \{\mathcal{Y}_{\text{P}}, \mathcal{Y}_{\text{I/C}}, \mathcal{Y}_{\text{O}}\}$.

This component approach affords the important advantage of allowing g to effectively map from the vast universe of possible structured terms (here, UMLS terms) to a relatively small set of those deemed reasonably likely for the text at hand. The selector model s can then perform more in-depth processing of candidates to infer likely configurations of candidate terms across the {P, I/C, O} elements, taking into consideration correlations between these subsets. In our case, this architecture was motivated in part by the existence of MetaMap, a tool that uses rules and heuristics to map from free text snippets to possible terms. This forms one part of our generator model, complementing a purely data-driven approach.

2.1 Selector Model

We begin by describing the selector model, s, which assigns a subset of the concepts contained in an unstructured candidate set C to the respective PICO elements, conditioned on the input text. An instance of this model (described in greater detail below) is depicted in Figure 2. Following [17, 33], we adopt a convolutional neural network (CNN) to encode texts. Concretely, we accept input texts to be annotated as sequences of words that are passed to an embedding layer that associates vectors (distributed representations) with words, thus forming an input matrix. We initialize word embeddings to pre-trained vectors induced over the entire set of abstracts indexed on MEDLINE, a repository of biomedical literature; we update these representations during model training via back-propagation. We apply independent convolutional filters of varying length over this matrix. That is, these filters consume one or more consecutive word embedding inputs at a time. Outputs from each filter are passed through a *max-pooling* operation to extract one scalar per filter (note that we use multiple filters of each filter size). These scalars are concatenated to form a final vector representation of the input text with a number of dimensions equal to the total number of convolutional filters. We will denote this induced representation of input i by $\mathbf{x}_i$.

The input text encoding approach just described is the same across the different Candidate-Selector (CS) model variants we discuss. What differs between them is the handling of the candidate concept(s) under consideration. For all variants we use embedded representations of controlled (UMLS) terms. We initialize these to pre-trained embeddings induced via *DeepWalk* [25], an approach to unsupervised distributed representation learning for graph-structured entities. During candidate classification, embedded representations of one or more candidate concepts are considered and the task is to decide whether these apply to the text under consideration, and if so, which PICO element they describe. We next describe three variants of our candidate-selector architecture, in ascending order of complexity.

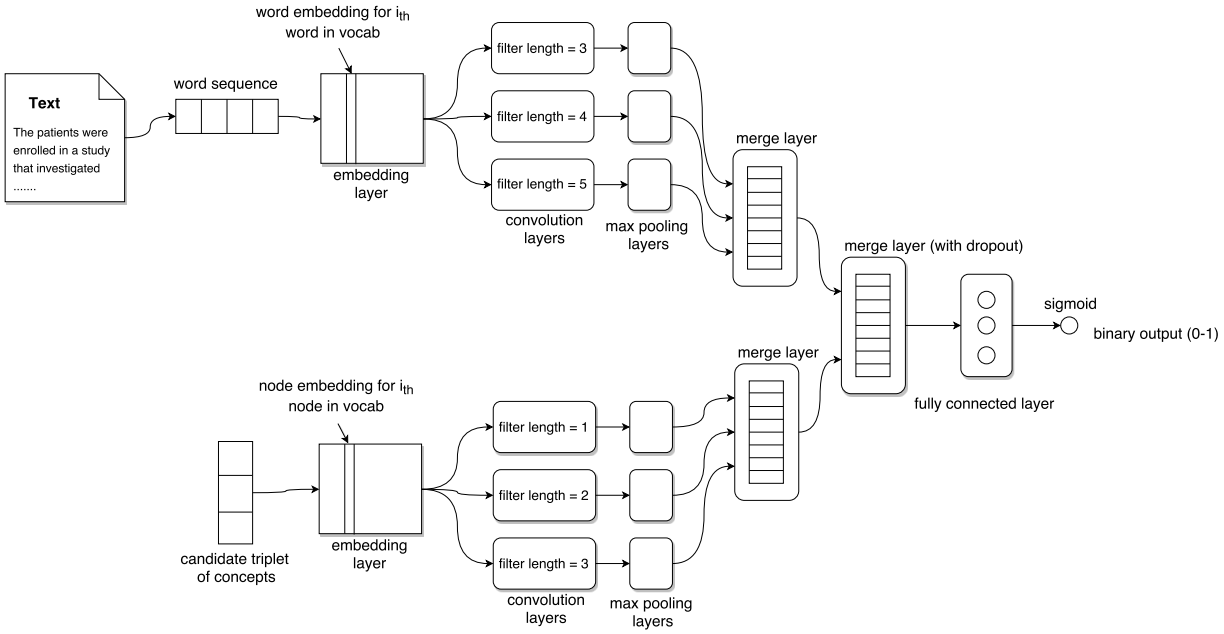


Figure 2: A schematic of our selector network variant *CS-joint*. This accepts as input the text snippets describing a study and a triplet of candidate concepts $(c_{\mathbf{P}}, c_{\mathbf{I/C}}, c_{\mathbf{O}})$, thus associating each candidate concept in the tuple with a particular PICO element. This induces a joint model that considers the likelihood of these three candidate concepts mapping to particular PICO concepts, given the input text. The output is a binary decision regarding the applicability of a candidate triplet.

CS-ind. The simplest variant of our model treats predictions regarding the designation of individual terms to respective PICO elements as independent, given the text. This model variant thus comprises three independent instances of the same model (i.e., with separate sets of parameters), one per PICO element. We concatenate the induced vector representations of the input text i and the (single) candidate concept under consideration (indexed by j) and estimate the probability of it being applicable to a given PICO element by running it through a logistic function σ:

$$P_e(\text{concept } j|\mathbf{x}_i^{(e)}, \mathbf{w}_o^{(e)}, \mathbf{C}^{(e)}, \mathbf{W}_h^{(e)}) = \sigma(\mathbf{w}_o^{(e)} \cdot \mathbf{v}_h)$$
$$\mathbf{v}_h = \lambda(\mathbf{W}_h^{(e)}[\mathbf{x}_i^{(e)} \oplus \mathbf{C}_j^{(e)}]) \quad (3)$$

Where e indexes PICO elements and hence models, making explicit the fact that the respective PICO model parameter sets are independent; $\mathbf{x}_i^{(e)}$ denotes a vector representation of input text i (induced via a CNN); $\mathbf{w}_o^{(e)}$ a weight vector parameterizing the output probability model; $\mathbf{C}^{(e)}$ the concept embeddings matrix; $\mathbf{W}_h^{(e)}$ a weight matrix for a hidden dense layer; and $\oplus$ denotes vector concatenation. Here $\lambda(\cdot)$ denotes an element-wise activation function (in our case, identity) and dropout regularization [27]. We reiterate that these predictions are made separately for each PICO element.

CS-cond. The patient population enrolled in a trial is not independent of the interventions and outcomes considered, as the former will clearly influence the latter. A first attempt to exploit such correlations is our *CS-cond* model, which starts by predicting which candidate terms describe the population, and then conditions the subsequent selection of terms corresponding to interventions on these. Finally, the selection of outcomes terms is explicitly conditioned on the preceding two sets of terms (i.e., the terms designated as describing the study population and interventions).

More formally, we use *CS-ind* to select terms $\hat{\mathcal{Y}}_{\mathrm{P}} \subseteq C$. We then use a modified architecture for the models that select intervention and outcomes terms. In particular, the model for predicting interventions accepts a third input matrix comprising the stacked embeddings corresponding to the terms in $\hat{\mathcal{Y}}_{\mathrm{P}}$.[2] Because the order of these terms is arbitrary, we pass only length 1 convolutional filters over this matrix (such filters consider a single concept at a time). We again apply max-pooling over these to induce a vector representations of the population concepts selected by the model in the preceding step, which we designate by $\mathbf{z}_i^{(P)}$.

$$P_{\mathrm{I/C}}(\text{concept } j|\mathbf{x}_i^{(\mathrm{I/C})}, \mathbf{w}_o^{(\mathrm{I/C})}, \mathbf{C}^{(\mathrm{I/C})}, \mathbf{W}_h^{(\mathrm{I/C})}, \mathbf{z}_i^{(P)}) = \sigma(\mathbf{w}_o^{(\mathrm{I/C})} \cdot \mathbf{v}_h)$$
$$\mathbf{v}_h = \lambda(\mathbf{W}_h^{(\mathrm{I/C})}[\mathbf{x}_i^{(\mathrm{I/C})} \oplus \mathbf{z}_i^{(P)} \oplus \mathbf{C}_j^{(\mathrm{I/C})}]) \quad (4)$$

The model for outcomes is analogous, except that it takes as an additional input a matrix comprising the embeddings for the terms selected both for populations and interventions/comparators, i.e., in addition to merging $\mathbf{z}_i^{(P)}$ to the model input we concatenate $\mathbf{z}_i^{(I/C)}$ before passing through the network. Thus the selection of outcomes terms is conditioned jointly on the inferred population and intervention descriptors.

[2] Operationally, we impose an upper-bound k on the number of terms that can be selected for a given element; thus the input matrix here is $k \times d$, where d is the embedding dimension.

CS-joint. Our final variant is a fully joint approach to selecting P, I/C and O candidate terms. This model consumes structured triplets as input (i.e., one candidate concept per PICO element) and estimates the conditional probability that these jointly apply to the text under consideration. The model is depicted schematically in Figure 2. In brief, we create an input matrix comprising the embeddings of the candidates in a given triplet, and run convolutional filters of lengths ranging from 1 to 3 over this input; this induces a vector representation of the triplet of candidate concepts which is then concatenated with the inferred representation of the input text to form a penultimate representation used to make a joint prediction concerning the applicability of the structured triplet of terms.

This model is attractive in that it affords a truly joint estimate regarding assignment of terms to PICO elements. However, it does mean that at test time we have to generate permutations of candidate concepts to make predictions for possible triplets in turn.

2.2 Candidate Generation

Having presented our approaches for candidate selection s, we now turn our attention to *generating* candidates provided an input text, i.e., specification of g. Broadly, we consider two approaches here, the outputs of which we compose: in the first we use a separate, pre-existing system called MetaMap to generate an unstructured set of candidate terms. We also adopt a data-driven *learned* approach to candidate generation. We describe these in turn below.

2.2.1 MetaMap. MetaMap [1] is a tool developed by the National Library of Medicine (NLM) that assigns concepts from Unified Medical Language System (UMLS) vocabularies to free-texts. Note, however, that it does not attempt to categorize these assigned concepts into PICO elements. The UMLS is a meta-ontology, incorporating ~200 standardised medical vocabularies. Synonymous terms are linked across vocabularies by unique semantic identifiers. MetaMap provides rich semantic information for biomedical informatics, but for our purposes it suffices to know that it implements a service which provides UMLS terms that match a given input text. We thus use MetaMap to generate an initial list of unstructured candidate concepts. A schematic of this process is shown in Figure 3. In general, under the settings used here, we found the candidate set generated by MetaMap to be high recall but relatively low precision.

2.2.2 Learning to Generate Candidates. In addition to MetaMap, we consider the approach of directly predicting UMLS concepts corresponding to the respective PICO elements from free-text. This model is one of the baseline approaches to which we compare our proposed Candidate-Selector models. Learning to map directly from free-text to structured UMLS terms has the advantage of allowing recognition of concepts not identified by MetaMap (the recall of the generated candidate set is an upper bound on the recall the selector model will be able to achieve). However, the disadvantage of this approach is that the output space is vast: there are hundreds of thousands of concepts; learning to predict directly into this space is thus challenging, especially given our limited training data. Additionally, as we discuss further below, this approach precludes the possibility of identifying concepts that were not encountered during model training.

To directly predict candidate terms for input texts we adopt a convolutional neural *multitask* [5] architecture, depicted in Figure 4.[3] In brief, we run input text through a CNN to induce a vector representation, as described in the preceding section. This learned representation is shared across the classification tasks corresponding to the respective PICO elements, thus affording transfer learning across tasks, insofar as the model learns parameters that induce a representation useful for recognizing terms descriptive of the respective PICO elements. Output layers, however, are treated as conditionally independent, given the shared input representation. Thus, e.g., the output layer corresponding to population comprises $|\mathcal{V}|$ binary output nodes (with associated weight vectors) corresponding to concepts in the vocabulary $\mathcal{V}$. Here, $|\mathcal{V}|$ = 366,772. This was prohibitively large, and so as a practical matter we restricted the output size to 150,000 terms (the same 150,000 for each PICO element). These terms include (1) all that appear in the available training sample for any given run, augmented with, (2) terms randomly (IID) sampled from the vocabulary.

2.3 Candidate set sampling details

We use the above two methods to generate candidates at test time. Here we describe the training and testing processes related to candidate sampling in greater detail.

During **training**, we draw positive triplets using the ground truth annotations. For example, if we have a set of ground truth annotations C_{P}, $C_{\text{I/C}}$ and C_{O} for an instance $\mathbf{x}$ then we construct positive triplets $(c_{\text{P}}, c_{\text{I/C}}, c_{\text{O}})$ by randomly and independently sampling one concept each from C_{P}, $C_{\text{I/C}}$ and C_{O}.

We also need to construct negative examples to be fed to the model during training. For this we use a 'negative sampling' approach in which we draw one or two concepts from the ground truth set, and the remaining concept(s) from the set of all concepts $\mathcal{V}$. We draw five negative triplets for every positive triplet, and pass these as input to the model in Figure 2. In addition to constructing triplets using concepts from the standard vocabulary, we assume the presence of a universal concept "_" in all annotations. This induces triplets of the form of $\{(c_{\text{P}}, \_, \_), (c_{\text{P}}, c_{\text{I/C}}, \_)\}$, in addition to fully specified triplets $(c_{\text{P}}, c_{\text{I/C}}, c_{\text{O}})$. Our CS-Joint model is defined directly over triplets in order to learn the joint distribution of concepts contained in different distinct sets. The introduction of underspecified triplets such as $(c_{\text{P}}, \_, \_)$ effectively allows the model to also learn marginal probabilities of concepts for a particular element, given an input text. We later empirically show the benefit of this approach.

During **testing**, we use the models described in the preceding subsections to generate candidate sets. Specifically, for a given input text, we use MetaMap to generate an unstructured list of candidate terms. We also use the multitask model described above (trained on the available training data) to make predictions based on the text, thereby inducing a supplementary, structured candidate set of terms, i.e., these are explicitly associated with individual PICO elements. We then exhaustively construct input candidate tuples by placing the MetaMap candidates into arbitrary slots, combining these with candidates assigned to specific PICO elements by the MT model. In this way, we construct every possible triplet $(c_{\text{P}}, c_{\text{I/C}}, c_{\text{O}})$

[3] This is similar to the multitask model used in [10].

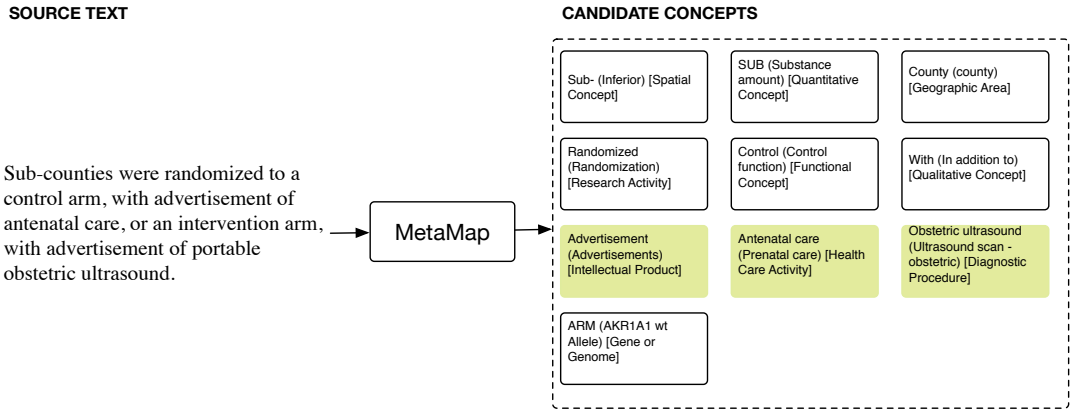


Figure 3: Schematic illustration of the use of MetaMap to generate a high recall set of candidate concepts. The target subset of concepts (here being those describing the interventions studied) are highlighted. Note that MetaMap output includes two types of noise: 1) An ambiguous string being assigned to the incorrect concept (e.g. 'Sub' being mapped to 'substance amount') and 2) the concept being correctly mapped from text but not describing our aspect of interest.

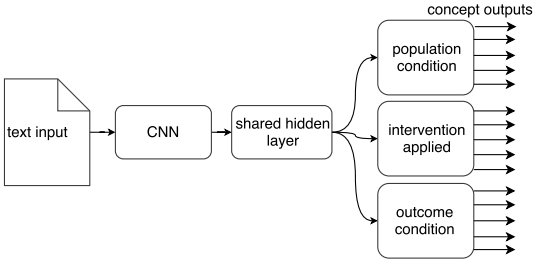


Figure 4: The multitask neural architecture we use to directly predict structured vocabulary terms from free texts.

samples (clinical trials)	4306
distinct population concepts	875
distinct intervention concepts	1115
distinct outcome concepts	1731
population concepts	9387
intervention concepts	5458
outcome concepts	13800

Table 1: Dataset statistics.

that can be derived from the candidate sets; this includes all possible incomplete specifications of the form $(\_, c_{I/C}, \_)$.

3 EXPERIMENTAL SETUP

We begin this section by providing details regarding the dataset used for experiments. We then describe the baseline models to which we compare our proposed approaches. Finally, we outline the evaluation setup we adopt and the metrics we use to assess performance.

3.1 Dataset

We use a real-world dataset provided by the Cochrane Collaboration,[4] which comprises manual annotations applied to biomedical publications. Specifically, aligning with the task we have outlined throughout this paper, trained annotators have applied tags from a subset of the Unified Medical Language System (UMLS) to free text summaries of biomedical articles, corresponding to the PICO elements. Recall that PICO stands for Population, Intervention/Comparator and Outcomes. These are defined briefly as follows. Population concerns the characteristics of or clinical problem shared by trial participants (e.g., diabetic males). Interventions are the active treatments being studied (e.g., aspirin); Comparators are baseline or alternative treatments to which these are compared (e.g., placebo) – the distinction is arbitrary, and hence we collapse I and C. The outcomes are the variables measured to assess the efficacy of different treatments (e.g., headache severity).

Trained annotators attach concept (UMLS) terms for each PICO element to individual free-text summaries of articles. These summaries comprise fields pertaining to each PICO element for every study. For this work, we merge them into single texts that span all PICO elements; this represents a more typical setup. All collected annotations undergo a rigorous quality assurance process; every annotation is subsequently checked by a domain expert.

3.2 Baselines

Two straightforward ways of performing the task under consideration are: (1) simply use MetaMap output, and, (2) train a model that learns to predict UMLS terms for each PICO element directly from the input text.

MetaMap. In the case of using MetaMap, it is not clear how best to assign the unstructured list of terms it provides for a piece of text to the respective PICO elements. Therefore, to make this baseline as competitive as possible, we 'cheat' in its favor by using text explicitly corresponding to different PICO elements. In particular, recall from above that in addition to attaching terms to abstracts,

[4]Cochrane is an international organization that focusses on improving healthcare decisions through evidence: http://www.cochrane.org/.

annotators also highlight the text corresponding to each PICO element. Therefore, we know which subspans correspond, e.g., to the population description in a given text. To induce P terms using MetaMap, we then pass *only* this population-specific text to MetaMap and retrieve the corresponding terms that it provides. We emphasize that *only* this baseline model has access to the span-level annotations at test time, which would not generally be available. Therefore, this represents an upper-bound on the performance we can expect to realize using MetaMap alone.

Multitask neural model. As a second baseline, we use the output candidate generation model introduced in Section 2.2.2 (and depicted in Figure 4). Recall that this is a multitask CNN that directly predicts terms for each PICO element, given the input text.

3.3 Evaluation Details

We divided the data into 60/40 for train/test split. We had ground truth annotations for all instances and for all three PICO elements, i.e., all texts have been annotated by domain experts with structured UMLS terms. The texts here are themselves summaries of each element written for previous reviews; we therefore concatenated these together, forming contiguous texts for each instance comprising spans relevant to the respective elements. We used only the 'Cochrane subset' of the UMLS. This is because the annotations we have (performed by Cochrane) contain only terms from this set. The Cochrane vocabulary comprises 366,772 concepts.

All hyper-parameter tuning was performed via nested validation (i.e., within train set). In particular, we used 30% of the training data for hyperparameter tuning. This included iteratively experimenting with and improving the structure of the network. The dropout rate [27] was tuned over a range of 10 equidistant values in the interval $[0, 1]$. The threshold for binary classification for each term (i.e., the threshold above which a term will be assigned) was tuned over the same range and interval. During hyperparameter search we optimized for average F1-score outputs. We trained for 100 epochs, caching and ultimately using the parameters that performed best on a nested validation set.

As mentioned previously, word embeddings were initialized to pre-trained vectors fit by running word2vec over all biomedical abstracts indexed on MEDLINE.

3.4 Metrics

We evaluated the performance of our approach using three standard metrics: precision, recall, and their harmonic mean (i.e., F1 score). We calculated these metrics for each instance and category (i.e., for each PICO element) separately, and aggregated over all instances for the respective categories to obtain MicroPrecision, MicroRecall and MicroF1 scores.

These metrics are strict because they require exact matches between predicted and true concepts. Results will thus be pessimistic in the sense that the model will be heavily penalized for predicting a concept that is semantically similar to (i.e., nearby in the ontology) — but not an *exact* match to — a target concept. As a simple means of relaxing match criteria, we therefore additionally report precision and recall at '2-hops' distance between annotations. Briefly, this counts a predicted term as a match to a target term if the former can reach the latter by taking two hops or fewer. More generally, we also report precision and recall at k hops for varying values of k in Figure 6.

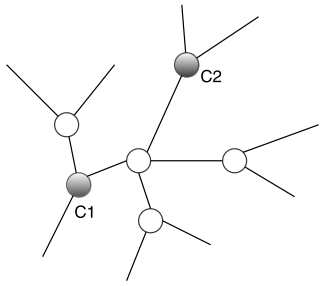


Figure 5: We consider two nodes at a distance of less than r hops as an 'r-hop match'; with this we compute the precision@r-hops and recall@r-hops metrics.

4 RESULTS

4.1 Quantitative Results

We report results for all models in Table 2. When reading the results here, which are low in absolute terms, it is important to keep in mind two key points. First, the output space is vast, which makes the task inherently quite difficult. And second, as mentioned above, the metrics are pessimistic here because they are very strict in requiring exact (or near-exact, in the case of the 2-hop metrics) matches.

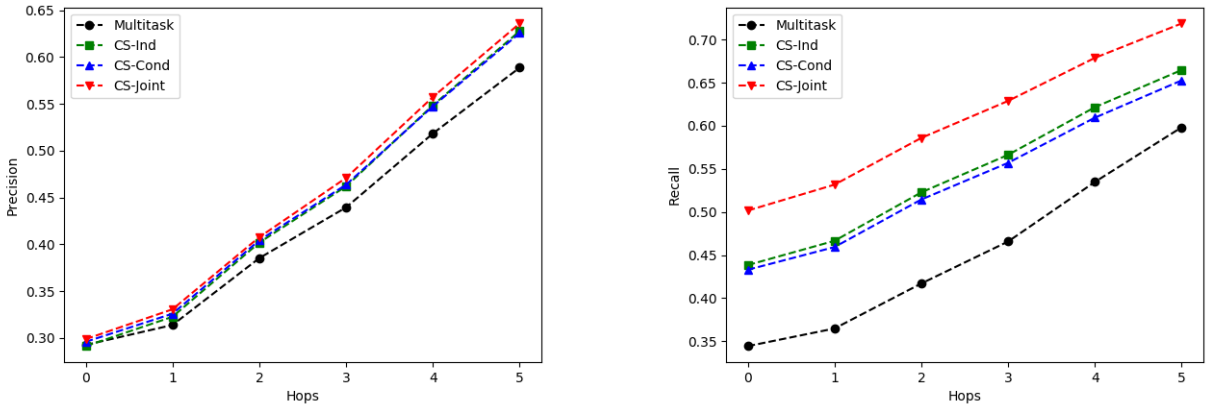
The methods prefixed with 'CS-' (below the dotted lines) are the three instantiations of the Candidate-Selector framework we introduced in Section 2; these are compared to the two baselines described in Section 3.2. A few observations: CS- approaches uniformly best baseline strategies, and the gains are considerable: we realize a 7-15 point absolute boost in F1-score, compared to the multitask neural model baseline. We also observe that the CS-Joint approach (Figure 2) yields the best performance for both precision and recall (and so also F1) for interventions and outcomes categories, and remains competitive with respect to population predictions (achieving the best recall at a modest cost in precision). This demonstrates the advantage of exploiting correlations between the PICO elements.

Figure 6 shows mean r-precisions and r-recalls (mean taken over the three PICO elements) achieved, as a function of r. Thus these plots show the results achieved under increasingly relaxed definitions of concept matches. Note that we omit the MetaMap baseline from these plots because it performed very poorly, to the extent that it rendered the plots difficult to read. The salient observation here is that the CS- models dominate the multitask CNN baseline, and the CS-joint model is consistently the best performing. In other words, the results just reported are robust to more relaxed definitions of concept matches.

4.1.1 Unseen Concepts. As mentioned at the outset of this paper, a challenge in healthcare applications of machine learning is limited training data. In our case, this is compounded by the very large output (label) space. As a consequence, the test data often contains concepts (i.e., labels) that were never seen in the training data.

Category	Model	Precision	Recall	F1-score	Pr-2hops	Re-2hops	F1-2hops
Population	MetaMap	0.134	0.280	0.181	0.262	0.489	0.341
	Multitask	0.358	0.383	0.370	0.501	0.502	0.501
	CS-Ind	**0.385**	0.529	0.446	**0.557**	0.636	**0.594**
	CS-Cond	0.384	0.535	**0.447**	0.553	0.640	0.593
	CS-Joint	0.318	**0.594**	0.415	0.485	**0.709**	0.576
Interventions/Comparator	MetaMap	0.108	0.288	0.157	0.163	0.387	0.230
	Multitask	0.248	0.245	0.246	0.264	0.262	0.263
	CS-Ind	0.226	0.272	0.247	0.274	0.322	0.296
	CS-Cond	0.225	0.282	0.250	0.275	0.331	0.300
	CS-Joint	**0.265**	**0.421**	**0.326**	**0.314**	**0.473**	**0.378**
Outcomes	MetaMap	0.209	0.391	0.273	0.314	0.518	0.391
	Multitask	0.198	0.211	0.204	0.283	0.290	0.286
	CS-Ind	0.272	0.497	0.352	0.380	0.593	0.464
	CS-Cond	0.268	0.497	0.348	0.378	0.591	0.461
	CS-Joint	**0.279**	**0.503**	**0.359**	**0.38**	**0.595**	**0.468**

Table 2: Precisions, recalls and f1 measures realized by different models on the respective PICO elements. Best result for each element and metric are bolded. Models with prefix 'CS' (below the dotted lines) are variants of the Candidate-Selector approach we have proposed in this work. We should mention that r-hop refers to the case when we consider a match between two concepts that are at a distance of $\leq r$ hops.



(a) Precision **(b) Recall**

Figure 6: Average (over PICO elements) r-precisions (a) and recalls (b) for each method as a function of r (i.e., using increasingly relaxed metrics; r-precision) counts a predicted concept as matching the truth concept when it is $\leq r$ hops away.

Approaches that learn to directly map from texts to predicted concepts would be generally incapable of predicting unseen concepts, by construction. Thus, e.g., our multitask CNN cannot predict a concept it has never seen in the training data, as there is no means of training the weights parameterizing the node corresponding to the unseen concept. However, because our Candidate-Selector architecture takes as inputs (embeddings of) candidate concepts, these can indeed be completely novel from the models perspective. Our use of MetaMap – and external candidate generator, effectively – means that it is entirely possible to select previously unseen terms. We show this in Table 6.

4.1.2 Pre-trained vs. Randomly Initialized Concept Embeddings. Recall that we use pre-trained distributed representations of medical concepts, induced via *DeepWalk* [25] performed over the UMLS graph. Here we explore the benefit (if any) of initializing embeddings to pre-trained vectors, as compared to randomly initializing them. In Table 4 we report results using these two initialization strategies. In general, using pretrained embeddings for initialization perhaps provides a slight edge, but the differences are not consistent.

Category	Model	Precision	Recall	F1-score
Population	MetaMap	0.190	0.274	0.224
	Multitask	0.355	0.562	0.435
	CS-Ind	0.413	0.758	0.534
	CS-Cond	**0.490**	0.731	**0.587**
	CS-Joint	0.413	**0.772**	0.539
Interventions	MetaMap	0.119	0.296	0.170
	Multitask	**0.298**	0.371	**0.331**
	CS-Ind	0.162	0.230	0.191
	CS-Cond	0.196	0.250	0.219
	CS-Joint	0.234	**0.420**	0.300
Outcomes	MetaMap	0.270	0.397	0.321
	Multitask	0.339	0.319	0.328
	CS-Ind	0.352	0.560	0.432
	CS-Cond	**0.356**	0.601	0.447
	CS-Joint	0.355	**0.633**	**0.455**

Table 3: Results on completely held out data (reference annotations were collected during model development).

4.1.3 Marginal vs. 'Complete' CS-Joint variant. Recall (Section 2) that the proposed CS-Joint model accepts as input triplets of candidate concepts, each assigned to a particular PICO element. This allows the model to exploit correlations between, e.g., populations and corresponding interventions. However, we would like to also enable the model to consider marginal probabilities of individual terms, conditioned on the input text). The model should be able to select these when appropriate, regardless of the other PICO term designations. To this end, in Section 2.3 we introduced the trick of including partially specified triplets, e.g., $(c_P, c_{I/C}, \_)\}$. Such partially specified triplets are also considered at test time during our exhaustive consideration of candidate triplets. The alternative would be to use *only* fully specified PICO triplets. To validate the 'marginals' approach adopted, we therefore compared these two strategies. We report results in Table 5. Using the partially specified (marginal) triplets clearly and uniformly improves model performance.

4.1.4 Results on final heldout data. Finally, we report results achieved by the final models (trained on the entire dataset explored thus far) on a completely new/heldout set of data, collected while we developed the model. This dataset comprises 88 instances, annotated in total with 76, 87, and 139 unique concepts corresponding to population, intervention/comparator and outcomes, respectively.

Results on this dataset are reported in Table 3. Here we report only one-hop measures for brevity, although results with respect to two-hop metrics are comparable. We can see that the proposed CS- models again generally best baselines, and that on average CS-Joint model performs the best of these, achieving a mean F1 across elements of 0.43, versus 0.42 for CS-Cond and 0.37 for the multitask model.

4.2 Qualitative Analysis

In addition to the quantitative results reported above, we performed a modest qualitative analysis. In particular, a selection of the model output on the test set was assessed qualitatively by an author who is clinically trained, and by an external annotation quality expert. We provide an illustrative example of model output in Figure 7. Qualitatively, the output was deemed usable for information retrieval purposes, and the majority of fields examined were populated with correct concepts. Missing concepts appeared to be the most common error type (e.g. 'Third Trimester Pregnancy' was correctly detected in Figure 7, but 'Second Trimester Pregnancy' was not); these typically appeared to be caused by a concept not being present in the candidate set generated via MetaMap. Some source texts were short and lacking in detail (particularly those describing outcomes), resulting in missed annotations.

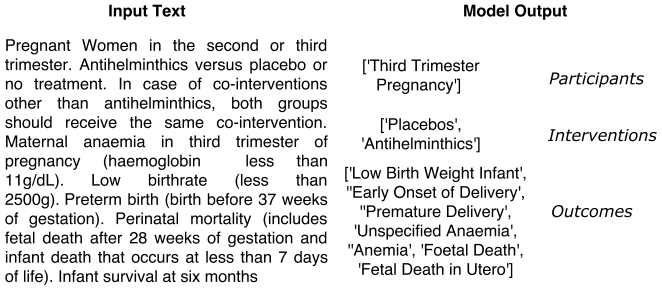


Figure 7: Illustrative example of model output.

Perhaps unsurprisingly, longer and more descriptive source texts appeared to result in better quality output from MetaMap. Our system currently does not make use of negation information; so, e.g., characteristics of excluded populations would be assigned a positive concept. Overall, the annotations appeared more useful qualitatively than the quantitative results might suggest (given the low absolute values, which we discussed in brief above).

5 RELATED WORK

We briefly review two threads of work related to our present effort: research on automated biomedical text annotation (Section 5.1) and then approaches to structured and multilabel classification. (Section 5.2).

5.1 Biomedical Text Annotation

Biomedical natural language processing is a broad, active field [12, 34]. Here we briefly review work relevant to our specific task of annotating text with structured PICO element concepts. One early system developed to extract clinical trial characteristics from free-texts is ExaCT [18], which aimed to identify and extract data elements from free texts describing clinical trials necessary for evidence synthesis. ExaCT used a hybrid of statistical and rule-based approaches. A similar system was developed by Summerscales [28]. His system attempted to automatically calculate summary statistics reported in an abstract by first identifying treatment group and outcome mentions and then processing numerical quantities in the text with reference to these.

Related work has attempted to identify spans or sentences of texts describing trials that correspond to the PICO elements. For example, Boudin et al. described ensemble methods for identifying sentences in abstracts corresponding to each PICO element [3]; they demonstrated that automatic PICO tagging can improve clinical IR [4]. More recently, Wallace and colleagues developed a model

Category	Model	Precision	Recall	F1-score	Pr-2hops	Re-2hops	F1-2hops
Population	CS-Joint random	**0.268**	**0.251**	**0.259**	0.386	0.382	0.384
	CS-Joint pre-trained	0.264	0.250	0.257	**0.392**	**0.392**	**0.392**
Interventions/Comparator	CS-Joint random	0.219	0.248	0.233	0.272	**0.294**	**0.283**
	CS-Joint pre-trained	**0.233**	**0.257**	**0.244**	**0.273**	0.293	0.282
Outcomes	CS-Joint random	0.315	0.302	0.308	0.412	0.404	0.408
	CS-Joint pre-trained	**0.341**	**0.356**	**0.348**	**0.440**	**0.449**	**0.445**

Table 4: The performance of the CS-Joint model when using randomly initialized versus pre-trained embeddings. Recall from above that the pre-trained embeddings for words were learned using *word2vec* [23] on MEDLINE abstracts, while the concept embeddings were learned using *DeepWalk* [25] over the medical concept vocabulary graph.

Category	Model	Precision	Recall	F1-score	Pr-2hops	Re-2hops	F1-2hops
Population	CS-Joint Complete	0.197	0.145	0.167	0.267	0.216	0.239
	CS-Joint +Marginals	**0.264**	**0.250**	**0.257**	**0.392**	**0.392**	**0.392**
Interventions/Comparator	CS-Joint Complete	0.156	0.149	0.153	0.180	0.168	0.174
	CS-Joint +Marginals	**0.233**	**0.257**	**0.244**	**0.273**	**0.293**	**0.282**
Outcomes	CS-Joint Complete	0.182	0.138	0.157	0.224	0.182	0.201
	CS-Joint +Marginals	**0.341**	**0.356**	**0.348**	**0.440**	**0.449**	**0.445**

Table 5: The performance of the CS-Joint model trained using only completely specified candidate triplets of the form $(c_{\mathbf{P}}, c_{\mathbf{I/C}}, c_{\mathbf{O}})$ (referred to as CS-Joint Complete) versus a variant that accepts partially specified frames like $(\_, \_, c_{\mathbf{O}})$ or $(c_{\mathbf{P}}, \_, c_{\mathbf{O}})$ and marginalizes over missing elements; we refer to the latter approach as CS-Joint +Marginals. As can be seen, the marginals approach yields consistently better predictive results, which is intuitive because it is less restricted, but still exploits correlations. This is the CS-Joint variant that we use.

Category	Unseen concepts	Correctly classified
Population	193	24
Intervention	326	54
Outcome	423	77

Table 6: The number of unseen concepts identified correctly by the proposed CS-Joint model. The proposed model can identify such unseen concepts due to the use of MetaMap to generate candidate concepts, which may be novel from the perspective of the model. However, our use of pre-trained concept embeddings means that even when previously unseen, the model is sometimes able to correctly select such concepts. Models that explicitly learn to map input texts to concepts will in general be incapable of recognizing concepts not present in the training data.

of extracting PICO sentences from full-texts, by exploiting a novel form of distant supervision [31].

Work has also been done on automatically assessing the 'risks of bias' in clinical trials, e.g., due to improper randomization, based on the text in the articles describing them. This work has entailed jointly extracting the sentences supporting these assessments [20, 21, 24, 32].

As far as we are aware, the present work is the first to consider the task of mapping from free-texts to structured concepts explicitly corresponding to the respective PICO elements.

5.2 Structured Multilabel Classification

The task we have considered may be viewed as an instance of structured multilabel classification. There is of course a rich body of work on general multilabel classification (e.g., [13, 14, 26]). It is challenging to learn an accurate and effective multilabel classifier in domains with many labels [29, 30]. Label space reduction methods provide one means of mitigating the problem of large label spaces [2, 7, 15].

More specific to the current application, multilabel classification for text has also received a fair amount of attention [16, 22]. A classic approach for multilabel text classification is to posit a generative mixture model wherein documents are associated with a set of labels that are in turn ascribed partial responsibility for generating the words comprising a given document [22]. It is not clear how to generalize this approach to our setting, however, because: (1) Labels are *grouped* as PICO elements which implies a correlation between these label sets, i.e., documents are not associated unstructured bags of labels; (2) Our output space (defined by a medical ontology) is vast, and thus a mixture model would require an unwieldy number of latent components.

Another sub-area of machine learning research relevant to our setting is multitask learning [5]. In particular, the PICO elements (and associated multilabel sets) may be viewed as distinctive 'tasks'; thus we find ourselves in effectively a multitask multilabel setting. Standard multitask learning has been studied at length in general, and in the context of natural language processing in particular [10, 11]. Indeed, we build upon the basic neural multitask architecture in [10] as a component in our approach.

To the best of our knowledge, this is the first work to explicitly consider the problem of jointly annotating texts with ontological labels for multiple, correlated aspects.

6 CONCLUSIONS

We developed a new model for structured clinical text annotation that can work effectively with limited training data. In particular, our model learns to infer terms from the UMLS metathesaurus that describe the individual PICO elements relevant to a given study, as described in an input free-text. This is an important practical task for biomedical natural language processing. Our model defines a novel Candidate-Selector architecture composed of two parts: candidate generation and then (possibly joint) selection and assignment of these candidates to constituent PICO elements. In our CS-Joint model the selection model is a Convolutional Neural Network jointly conditioned on a triplet of structured PICO UMLS terms and the free-text to be annotated, thus realizing a fully joint approach. This model achieved consistently strong empirical results, besting alternative approaches.

Moving forward, we believe we can further improve upon this model within the same framework, by better exploiting the ontological structure underlying UMLS. We also hope to focus efforts on improving the recognition of novel (unseen) terms, as this is important for the present task.

7 ACKNOWLEDGEMENTS

JT and GS acknowledge support from Cochrane via the Transform project. BCW was supported by the National Library of Medicine (NLM) of the National Institutes of Health (NIH), grant R01LM012086. IJM acknowledges support from the MRC (UK), through its grant MR/N015185/1.

REFERENCES

[1] Alan R Aronson and François-Michel Lang. 2010. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association* 17, 3 (2010), 229–236.
[2] Wei Bi and James Tin-Yau Kwok. 2013. Efficient Multi-label Classification with Many Labels.. In *ICML (3)*. 405–413.
[3] Florian Boudin, Jian-Yun Nie, Joan C Bartlett, Roland Grad, Pierre Pluye, and Martin Dawes. 2010. Combining classifiers for robust PICO element detection. *BMC medical informatics and decision making* 10, 1 (2010), 29.
[4] Florian Boudin, Lixin Shi, and Jian-Yun Nie. 2010. Improving medical information retrieval with PICO element detection. In *European Conference on Information Retrieval*. Springer, 50–61.
[5] Rich Caruana. 1998. Multitask learning. In *Learning to learn*. Springer, 95–133.
[6] Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. 2015. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 507–516.
[7] Yao-Nan Chen and Hsuan-Tien Lin. 2012. Feature-aware label space dimension reduction for multi-label classification. In *Advances in Neural Information Processing Systems*. 1529–1537.
[8] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*. 301–318.
[9] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 3504–3512.
[10] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.
[11] Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815* (2009).
[12] D Demner-Fushman, N Elhadad, et al. 2016. Aspiring to Unintended Consequences of Natural Language Processing: A Review of Recent Developments in Clinical and Consumer-Generated Text Processing. *IMIA Yearbook* (2016), 224–233.
[13] André Elisseeff, Jason Weston, et al. 2001. A kernel method for multi-labelled classification. In *NIPS*, Vol. 14. 681–687.
[14] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. 2008. Multilabel classification via calibrated label ranking. *Machine learning* 73, 2 (2008), 133–153.
[15] Shuiwang Ji and Jieping Ye. 2009. Linear Dimensionality Reduction for Multi-label Classification.. In *IJCAI*, Vol. 9. 1077–1082.
[16] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2008. Multilabel text classification for automated tag suggestion. *ECML PKDD discovery challenge* 75 (2008).
[17] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).
[18] Svetlana Kiritchenko, Berry de Bruijn, Simona Carini, Joel Martin, and Ida Sim. 2010. ExaCT: automatic extraction of clinical trial characteristics from journal publications. *BMC Medical Informatics and Decision Making* 10, 1 (2010), 56. https://doi.org/10.1186/1472-6947-10-56
[19] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzell. 2015. Learning to diagnose with LSTM recurrent neural networks. *arXiv preprint arXiv:1511.03677* (2015).
[20] Iain J Marshall, Joël Kuiper, and Byron C Wallace. 2014. Automating risk of bias assessment for clinical trials. In *Proceedings of the 5th ACM conference on Bioinformatics, computational biology, and health informatics*. ACM, 88–95.
[21] Iain J Marshall, Joël Kuiper, and Byron C Wallace. 2016. RobotReviewer: evaluation of a system for automatically assessing bias in clinical trials. *Journal of the American Medical Informatics Association* 23, 1 (2016), 193–201.
[22] Andrew McCallum. 1999. Multi-label text classification with a mixture model trained by EM. In *AAAI workshop on text learning*. 1–7.
[23] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[24] Louise AC Millard, Peter A Flach, and Julian Higgins. 2016. Machine learning to assist risk-of-bias assessments in systematic reviews. *International journal of epidemiology* 45, 1 (2016), 266–277.
[25] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.
[26] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2009. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases* (2009), 254–269.
[27] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
[28] Rodney L Summerscales, Shlomo Argamon, Shangda Bai, Jordan Hupert, and Alan Schwartz. 2011. Automatic summarization of results from clinical trials. In *Bioinformatics and Biomedicine (BIBM), 2011 IEEE International Conference on*. IEEE, 372–377.
[29] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 104.
[30] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2008. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD Workshop on Mining Multidimensional Data*. 30–44.
[31] Byron C Wallace, Joël Kuiper, Aakash Sharma, Mingxi Brian Zhu, and Iain J Marshall. 2016. Extracting PICO sentences from clinical trial reports using supervised distant supervision. *Journal of Machine Learning Research* 17, 132 (2016), 1–25.
[32] Ye Zhang, Iain J. Marshall, and Byron C. Wallace. 2016. Rationale-Augmented Convolutional Neural Networks for Text Classification. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*. Associaton for Computational Linguistics (ACL), 795–804.
[33] Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* (2015).
[34] Pierre Zweigenbaum, Dina Demner-Fushman, Hong Yu, and Kevin B Cohen. 2007. Frontiers of biomedical text mining: current progress. *Briefings in bioinformatics* 8, 5 (2007), 358–375.