

Automated tabulation of clinical trial results: A joint entity and relation extraction approach with transformer-based language representations

Jetsun Whitton^a, Anthony Hunter^a

^a*Department of Computer Science, University College London, Gower
Street, London, WC1E 6BT, UK*

Abstract

Evidence-based medicine, the practice in which healthcare professionals refer to the best available evidence when making decisions, forms the foundation of modern healthcare. However, it relies on labour-intensive systematic reviews, where domain specialists must aggregate and extract information from thousands of publications, primarily of randomised controlled trial (RCT) results, into evidence tables. This paper investigates automating evidence table generation by decomposing the problem across two language processing tasks: *named entity recognition*, which identifies key entities within text, such as drug names, and *relation extraction*, which maps their relationships for separating them into ordered tuples. We focus on the automatic tabulation of sentences from published RCT abstracts that report the results of the study outcomes. Two deep neural net models were developed as part of a joint extraction pipeline, using the principles of transfer learning and transformer-based language representations. To train and test these models, a new gold-standard corpus was developed, comprising over 550 result sentences from six disease areas. This approach demonstrated significant advantages, with our system performing well across multiple natural language processing tasks and disease areas, as well as in generalising to disease domains unseen during training. Furthermore, we show these results were achievable through training our models on as few as 170 example sentences. The final system is a proof of concept that the generation of evidence ta-

Email addresses: jetsun.whitton.20@alumni.ucl.ac.uk (Jetsun Whitton),
anthony.hunter@ucl.ac.uk (Anthony Hunter)

bles can be semi-automated, representing a step towards fully automating systematic reviews.

Keywords:

Natural Language Processing, Information Extraction, Systematic Review, Transformer, BERT, Randomised Controlled Trial, Evidence Table

1. Introduction

Over the last three decades, decision making in clinical practice has been driven by the systematic evaluation of healthcare evidence in what is known as evidence-based medicine (EBM). Defined as: “*the conscientious, explicit and judicious use of current best evidence in making decisions about the care of individual patients*”, EBM sets systematic evaluation standards to reduce bias and improve quality in clinical reports [1, 2]. Its goal is to combine high quality evidence with clinical experience and patient preference to achieve the best possible outcomes in care.

The gold standard for evidence in the healthcare domain is the randomised controlled trial (RCT) – a study where selected participants are randomly allocated into groups to test a specific drug, treatment or other intervention. These groups, also known as trial arms, are allocated either the study intervention (study arm[s]) or a comparator (control arm[s]) that could be another intervention or placebo. Both arms are then measured and compared over a period of time on a set of predefined outcomes, primarily efficacy and safety, to ascertain the effectiveness of the study intervention.

In the same time-frame that EBM has become the mainstay of modern medicine, the number of registered clinical trial studies has risen greatly [3]. Many disease areas are crowded with old and new treatments, each of which may be evidenced by several clinical trials that report varying risks and benefits in different patient groups. This makes it difficult or simply impossible for individual clinicians to keep abreast of the latest evidence through the traditional method of reading papers. Instead, they must turn to systematic literature reviews, which aggregate available evidence, predominantly from RCTs, for answering predefined clinical questions and making decision recommendations. As such, systematic reviews are extensively used by healthcare bodies for developing clinical guidance, including the National Institute for Health and Care Excellence (NICE) in the UK, the European

Outcome	Intervention	Control	Comparative	Quality
Mean change in intraocular pressure (IOP)	-6.3 mmHg	-0.2 mmHg	Not reported	High
Visual field progression at 24 months	23 (12.6%) patients	57 (27.4%) patients	Hazard ratio: 0.49 95% CI:0.21–0.67 p=0.037	Medium
Serious adverse events	12 events	7 events	Not reported	Low

Figure 1: An example of how an evidence table for an RCT investigation of a glaucoma treatment might look in NICE clinical guidelines [5]. CI: confidence interval.

Medicines Agency (EMA), the U.S. Food and Drug Administration (FDA) and the World Health Organisation (WHO).

Conducting systematic reviews is a labour intensive process: published RCT papers must be searched for on medical publication sites such as PubMed, Medline or UptoDate, screened for inclusion, and then read carefully to extract the relevant information into evidence tables (Figure 1). To provide an appropriate summary, this information needs to be relatively extensive and detailed, including data on the trial arms, patient populations and study results. The information extraction (IE) step is generally performed using a predefined framework, which provides a consistent approach to decomposing clinical questions, so that specific and recurring data elements can be retrieved to answer them, the most common being the **P**opulation, **I**ntervention, **C**ontrol, **O**utcome (PICO) framework [4].

While annotation software tools are available, much of the systematic review process still requires manual input from domain specialists that is both time-consuming and expensive [6, 7]. It has been estimated that the average yearly cost of systematic reviews is about 18 million dollars for each academic institution and 17 million dollars for each pharmaceutical company [7]. An automated system that can extract information from RCTs and automatically tabulate it into evidence tables is therefore highly desirable. However, the real value of such a system lies in its potential to overcome some of the key limitations of clinical guidelines [8]. New guidelines often take years to produce, and can quickly go out of date as new evidence becomes available. They also rarely account for the whole multitude of different patient characteristics and local regulations that a decision maker may face, restricted

by the prohibitive cost and complexity of conducting a single or multiple systematic reviews to cover every imaginable detail. As a tool for healthcare bodies or even decision makers themselves, automated evidence aggregation could keep up with the pace of new publications and reduce the cost barrier to developing personalised recommendations for specific patient characteristics and local requirements [8].

There has been some headway towards this goal; a number of natural language processing (NLP) studies have looked at rule-based [9, 10], statistical [9, 11–13] and, more recently, neural net (NN) models [14–16] for the automated extraction of information from RCTs, achieving varied results. Many of these investigations have focused on identifying relevant information at the sentence level, requiring further methods or human intervention to process the granular detail needed for an evidence table. Extraction of more detailed information, using techniques such as named entity recognition (NER) [16–21] and relation extraction (RE) [22–25], has progressed greatly over the last few years, largely thanks to improvements in the way language is represented by machines [26–28], transfer learning [27], and the release of large gold-standard corpora of clinical trial annotations [29]. However, to our knowledge no study to date has utilised advanced NER and RE techniques in combination to automatically extract and differentiate intervention, outcome, and outcome measure entities into the appropriate columns of an evidence table for a systematic review.

In the current study, we investigate automatic IE from RCTs for such a task, focusing on the tabulation of result sentences from the abstracts of published papers. To achieve this, we decompose the problem across an NLP pipeline with two transformer-based components – an NER model for extracting full-sequence entities and a RE model for identifying the relations between them – whose output we use to construct appropriate tuples for an evidence table (Figure 2). Specifically, we look to identify intervention, outcome and outcome measure entities, and use their respective relationships to sort them into tuples of the form (*outcome, arm 1, arm 2*).

In addition, we present a new gold-standard corpus used to train our models of over 550 result sentences from across six disease areas, with both named-entity and relational annotations. This corpus is made publicly available along with the system code in our study repository¹. Evaluation of our

¹<https://github.com/jetsunwhitton/RCT-ART.git>

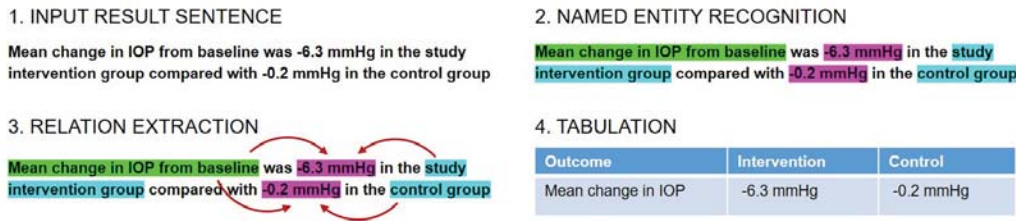


Figure 2: A simple example of how an input result sentence (1) can be processed through NER (2) and RE (3) and then tabulated (4) to form a segment of the example evidence table from Figure 1.

system explores performance differences gained by fine-tuning language representation models that were pre-trained on domain-specific corpora versus fine-tuning a model that was pre-trained on a general corpus. We demonstrate that the system performs well across multiple NLP tasks and in generalising to unseen disease area domains. We also show that it can achieve this performance with training on as few as 170 example sentences.

2. Background

2.1. Related work

Automating IE from healthcare publications has been the subject of considerable research, across a variety of NLP methods. PICO extraction is a key component of many of these studies, and is often a first step in more complex tasks such as argument mining and question answering [30–32]. However, there are significant differences across publications in terms of the PICO elements targeted for extraction, as well as their levels of detail, with investigations tending to focus on either full sentence classification or fine-grained, sub-sentence extraction of named entities.

2.1.1. NLP and clinical trials

NLP is becoming an increasingly valuable tool for clinical trial research, with its applications including trial design optimisation, patient recruitment and eligibility screening, and preparation for regulatory submissions [33].

Automated IE from unstructured data such as published articles – either through NER (see subsection 2.1.2) or classification of full sentences – is the first step for many of these tasks, and has undergone rapid advancement in techniques. In 2005, Demner-fushman et al. [9] classified published RCT

sentences containing study outcomes using an assortment of rule-based and statistical methods. Fifteen years later, a study by Zhang et al. [15] successfully demonstrated how long-term short-term memory and transformer-based architectures can be used to classify sentences across the full PICO framework.

Medical concept normalisation represents another important line of research that builds on IE of clinical trial data, [34–38] due to the wide variation in medical terminology that occurs even in structured repositories such as ClinicalTrials.gov [38]. By extracting different names for the same concept and mapping them to a unified term, normalisation can ensure trial data conforms to a standardised taxonomy for downstream applications [34–38]. Sophisticated NLP techniques are facilitating new breakthroughs in this task, such as in a study by Miftahutdinov et al. [38], who trained a transformer-based model with a triplet loss function to map distances between medical term mentions and positive and negative concept examples.

Automated patient matching to clinical trial eligibility criteria through searching health records is also being advanced greatly by modern NLP methods [39, 40]. In 2010, the first system to automatically retrieve trial eligibility information used simple pattern matching on surgical pathology reports [41]. A recent study by Hassanzadeh et al. [40] showcases how the subsequent decade of advancements in NLP can be applied to this field, combining concept normalisation with document vector embedding to semantically enrich patient records for eligibility classification by a NN.

2.1.2. NER of PICO elements

NER involves the labelling of sub-sentence lengths of unstructured text that describe named entities with predefined categories (e.g. labelling “aspirin” as “drug”). Around a decade ago, researchers using NER to extract PICO elements generally focused on identification of only the population, intervention and control categories. These studies tended to use statistical machine learning (ML) models such as conditional random fields (CRFs), support vector machines and naïve Bayes to classify noun-phrases and sentences, and then further processed these with hand-crafted rules and regular expression matching to extract PICO elements [17, 18]. Although these approaches have been shown to be effective to varying degrees, relying on hand-crafted patterns and rules can limit a model’s ability to generalise. To overcome this problem, Trenta et al. [10] restricted the use of rules and pattern-matching in their two-stage classification system that used a statistical ML model to

identify the syntactic heads of PICO entities (e.g. “patient” in “patient with glaucoma” or “corneal” in “corneal implant”) in RCT abstracts. This study included both outcomes and measure entities, and achieved encouraging results; however, classification of full-span entities was left to future research.

Another key limitation of PICO NER has been the lack of publicly available training data, with researchers such as Trenta et al. sourcing and annotating their own RCT publications. Along with presenting an expensive and time-consuming barrier to entry for this research area, the self-annotation of text creates a number of issues; the most salient being that variations in corpus annotation methods inhibits meaningful comparisons of system performance across different studies. However, steps to overcome this barrier were made in 2018, when Nye et al. [29] released the EBM-NLP corpus, which contains 4,993 PICO annotated abstracts of RCTs from the MedLine database, and was developed using a combination of crowdsourcing and expert review. This dataset has now been used by multiple studies to train current state of the art (SOA) models, most commonly with deep NN architectures. These include: a recurrent NN trained by Brockmeier et al. [19] for identifying PICO elements to score abstract relevancy against systematic review questions; a long short-term memory (LSTM)-CRF model trained by Nye et al. [21] for the PICO extraction component in their live and automated RCT classification system, Trialstreamer; and a LSTM-CRF model by Kang et al. [20], who further labelled a subset of studies from the EBM-NLP corpus with outcome measures to train one of the most comprehensive PICO NER systems to date.

2.1.3. RE between PICO elements

RE is a core IE task with a variety of approaches, seeking to identify the contextual relationships between sentences or entities, such as which outcome measure belongs to which study arm (e.g. identifying that the outcome measure, “39.3% of patients had unacceptable intraocular pressure”, belongs to the intervention arm that received the study drug, “latanoprost”). As the gateway to semantic reasoning, identifying the relations between PICO entities is a complex task, especially if entities are jointly extracted, and is less studied than NER alone, particularly prior to the development of contextualised embeddings [22]. Since this milestone, however, PICO extraction studies with this objective have begun to trend.

Initially, studies of RE with contextualised language representations focused on identifying general bio-medical entity-pair relations, rather than

direct PICO elements. For example, studies by Lim and Kang [23] and Joël et al. [42] sought to jointly extract gene–disease and drug–disease entity pairs, respectively, and classify their relations, using LSTM architectures to first extract entity pairs, and then relations via a grammar dependency mechanism. More recently, Nye et al. [24] and DeYoung et al. [25] used transformer-based models to extract ICO entities (ignoring population in PICO) and their relations, using this information to construct ICO triplets, where an intervention, control and outcome description are matched with a comparative outcome description (e.g. intervention *reduced* outcome compared with control).

In this study, we build on the concept of the ICO triplet, by looking beyond comparative outcome descriptions. Instead, we seek to create triplet tuples where outcome descriptions, the interventions for each study arm and their individual outcome measures are divided into respective tuple positions (i.e. columns of a table).

2.2. Transformer-based language representations

Precise understanding of context is an important challenge in NLP, particularly when it comes to the complexity of medical terminology and concepts. Transformers [26] and transformer-based encoding architectures [27, 28] are a relatively recent breakthrough that have greatly advanced contextualised language representation, using attention mechanisms to embed and encode word tokens and their contextual information into a feature vector space. Bidirectional Encoder Representations from Transformers (BERT) [27] is one such architecture, with its encoded representations considering contextual words from both left-to-right and right-to-left of the target word token. This is particularly important for NLP tasks such as NER and RE, which rely on context from both directions.

Extensive research has now built on the original BERT system. Some studies have sought to optimise performance through adapting BERT’s architecture and training procedure, such as with the general model RoBERTa [43]. Others have extended pre-training with enormous domain specific corpora: SciBERT [44] and BioBERT [45] being two such systems in the biomedical domain. Using the weights of the original BERT model as a base, BioBERT is pre-trained on abstracts from PubMed and full text articles from the PubMed Central archive, accounting for a sum-total of 14 billion additional words, while SciBERT is pre-trained from scratch on 1.14 million

papers from Semantic Scholar (over three billions words). Both models outperform the original BERT model on NLP tasks in the biomedical domain. In the development of our NLP pipeline, we explore both of these domain-specific language representations, as well as the general RoBERTa model for comparison.

3. Dataset creation

In this section, we discuss the development of our dataset used to train and test our RCT result tabulation system. The novel data included in our corpus was created through annotating sentences from structured RCT abstracts. We collected these abstracts from two sources, the Trenta et al. [10] study dataset of glaucoma RCT abstracts and the EBM-NLP corpus [29], which were preprocessed and cleaned before independent annotation by three experts for the given task at hand. Our final gold-standard dataset included over 558 annotated sentences from six disease-area domains.

3.1. Data collection

As the primary evidence in EBM systematic reviews, we used RCTs as the source of data for our study. In particular, we decided to focus on the abstracts of published RCT papers, as these are both freely available and offer a structured summary of the trial, which should conform to the CONSORT policies published in 2010 [46]. These guidelines outline how RCT publications should be constructed, and include a checklist to ensure all key trial information and results are reported within the abstract, within the following labelled sections: background, objective, method, results and conclusion. In addition to ensuring they are a reliable RCT overview, these minimum requirements for information mean abstracts from different studies are comparable in terms of included PICO entities, even across different disease-area domains.

In line with our focus of extracting respective relationships between study arms, outcomes and measures, we made two further key restrictions to data collection, adapted from those used by Trenta et al [10]. First, we only included abstracts from RCTs with a two-group study design. This was done to simplify our task and limit the study’s scope, as sentence complexity of reported results, as well as the number of entities and relationships to track, increases in line with the addition of study arms beyond two. Second, we limited our dataset to abstract result sentences that include at least one

outcome and/or study arm and a clear, numerical measure with respective relationships between these entities. Examples of such sentences can be seen in subsection 3.3. These result sentences were annotated to form the novel training and test data used for framing the NER and RE tasks as supervised-learning classification problems.

3.2. Abstract sources

3.2.1. *Trenta et al. glaucoma*

Our dataset was initially composed from that of the Trenta et al. [10] study, which similarly investigated the extraction of PICO information with respect to study arms, albeit with a different approach that focused on statistical-based methods. This dataset comprises RCT study abstracts in the disease area of glaucoma, collated from PubMed with three search strategies: 1) titles and abstracts including “glaucoma” and specifying the study as an RCT; 2) titles including at least one prescription drug for glaucoma or ocular hypertension from a predefined list, and specifying the study as an RCT; 3) titles including at least one surgical procedure for glaucoma or ocular hypertension from a predefined list, and specifying the study as an RCT. These queries retrieved 176 abstracts, with this set filtered using inclusion criteria similar to those outlined in the previous section. We have included all 99 of the resulting abstracts in our dataset, with 211 result sentences included in our corpus.

3.2.2. *The EBM-NLP corpus*

To improve the generalisation capacity of our trained system, we looked to extend our dataset beyond one disease-area domain with the recent EBM-NLP corpus [29] for PICO extraction. The corpus is comprised of 4,993 RCT abstracts sourced from the MedLine with a general focus on the domain areas of cardiovascular disease, autism and cancer, covering a range of common conditions. Unlike the glaucoma dataset of Trenta et al. [10], the EBM-NLP corpus was developed with no particular data extraction method in mind, and as such, had no inclusion criteria beyond requiring abstracts to describe an RCT. Because of this, and the lack of separation between domain data, it was necessary to screen the data for collection from this corpus, which we accomplished through a mixture of automated techniques and manual screening at the annotation stage. We retrieved datasets for five disease-area domains, including autism (208 abstracts), blood cancer (74 abstracts), solid tumour cancer (200 abstracts), diabetes (97 abstracts) and cardiovascular

disease (159 abstracts), from which we included 45, 15, 129, 46 and 112 sentences in our study corpus, respectively.

3.3. Annotation

Our gold-standard data was created with a two-stage annotation process, entity labelling and relationship labelling, using the Prodigy Python package by Explosion [47]. Prodigy is an annotation tool for local-server, browser-based text labelling across a variety of NLP tasks, and was made available to us through a free research licence. As highlighted in the data collection section, only result sentences with at least one numerical measure of an outcome with respect to a treatment arm were included for annotation. Comparative measures (e.g. an odds ratio comparison of two risk measures) are omitted.

Three medical writers, each with over five years' experience, were recruited as our expert annotators. Before each annotation task, the annotators were trained on annotation guidelines, which we make available in the supplementary Appendix. These guidelines include instructions on using Prodigy and on each of the annotations tasks, outlining their requirements in full, as well as any specific considerations for each of the label types, such as rules for entity boundaries. During the training phase, the annotators also had the opportunity to feedback on the guidelines, which were then refined before the full independent annotation tasks were commenced. After independent annotation, inter-annotator agreement (IAA) between the datasets was calculated (see subsection 3.4) and a reconciliation phase, where annotators resolved disagreements (see subsection 3.5), was conducted to define the gold label corpus. A final pass over the merged dataset was conducted to ensure that it was consistent with any new annotation rules defined to resolve disagreements. These rules were added to the annotation guidance, both for the final pass and for future studies that wish to follow this corpus development methodology.

3.3.1. Entity annotation

Three types of entity (Figure 3) were labelled during this stage of the annotation process: interventions (INTV), outcomes (OC) and measures (MEAS). A brief overview of the annotation guidance for each of these labels is as follows:

- **INTV:** label study treatments and comparators that patients have been randomised to receive

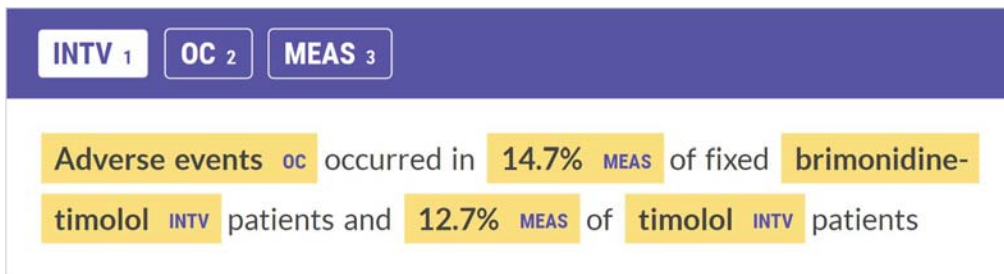


Figure 3: NER annotations in Prodigy.

- **OC**: label any description of the study results, most commonly describing measures of effectiveness and safety
- **MEAS**: label any clear, non-comparative, numeric measure that can be related back to an outcome and/or intervention within the sentence

A decision was made early in the study not to distinguish between the entity labels of interventions and their comparators, as well as their respective outcome measures (i.e. separate labels for the intervention measure and comparator measure). While identifying these entities would make the task of dividing data into the study-arm columns of an evidence table relatively trivial, the similarly fine-detailed labels of the EBM-NLP corpus were shown to significantly reduce model performance (F_1 score reduction of from 0.68 to 0.46 vs broad labels) [29]. Moreover, outcome measures need to be related back to their respective outcomes, which would necessitate either even more specific entity labels or overlapping entity spans, the latter being highly non-trivial for NER models [48]. Instead, we sought to decompose the complexity of the task by sub-dividing it between the NER and RE components of our system. Masking the intervention-comparator division allows our NER component to focus on classifying a simpler set of entity labels, which are then passed to our RE component, where entities are further distinguished through classifying the relations between entity pairs.

3.3.2. Relationship annotation

Entity relations were labelled between relevant entity pairs (Figure 4) in each sentence, with three types of relationship labels: relations between measures and their **RES**pective outcome description (**OC_RES**), and relations between measures and their **RES**pective intervention arms, which we have

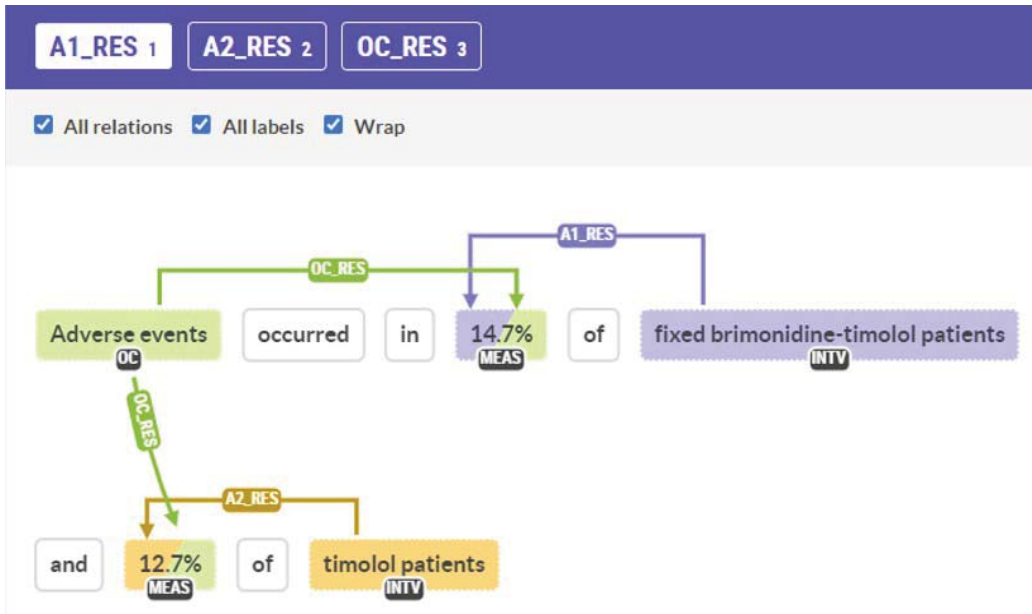


Figure 4: RE annotations in Prodigy.

	C1	C2	C3
1	Outcomes	Arm 1	Arm 2
2	intervention	brimonidine-timolol	timolol
3	Adverse events	14.7%	12.7%

Figure 5: Output CSV of a gold-standard table, generated by our tabulation component from the gold-standard NER and RE annotations.

limited to two in the current investigation (A1_RES, A2_RES). Relationship labels were also directional, highlighting a parent-to-child dependency, with the INTV and OC being parent entities and MEAS being the child entity.

Although further relations could have been annotated, such as the comparative relationship between study arms, we decided to limit complexity through restricting these labels to the minimum needed for our tabulation component to construct a result sentence into an evidence table (Figure 5), with a column for each arm and row for an outcome description.

3.4. Inter-Annotator Agreement

All three expert annotators independently labelled the full study corpus for both annotation tasks. IAA between the three independent datasets was calculated after each of the tasks to validate and assess reliability of the annotations. Fleiss’ kappa, a generalisation of Scott’s pi [49], was chosen as the statistical measure to calculate IAA. This was based on its suitability for finite nominal-scale data and flexibility in being able to calculate agreement between more than two annotators, unlike Scott’s pi or Cohen’s Kappa [49]. Fleiss’ Kappa assesses the observed agreement corrected for the agreement expected by chance [49].

3.4.1. Entity annotation task agreement

IAA for this task was calculated at the token level to account for disagreement in the annotation of entity boundaries, and covered all result sentences that were included in the study and annotated. The Fleiss Kappa calculation included four categories – the three label types (*OC*, *INTV*, *MEAS*) and *no label given*. The *no label given* category led to two approaches being taken with regard to token inclusion in the calculation, based on previous research findings that missing label categories tend to introduce substantial bias to the Kappa coefficient [50]. The first of these approaches includes all tokens, regardless of rating category, and resulted in a Fleiss Kappa of 0.82, which can be interpreted as almost perfect IAA. The second approach omits tokens with *no label given* by any of the three annotators. The goal of this approach was to reduce the bias introduced by this category through removing tokens that all annotators agreed were not part of any entity, being the most common unanimous clarification agreement by far and disproportionately weighting the calculation. The Fleiss Kappa for this approach was 0.71, indicating substantial agreement between the annotators.

3.4.2. Relation annotation task agreement

The IAA for the relation labels was calculated for all possible entity pairs in each annotated sentence. Similarly to entity IAA assessment, the Fleiss Kappa calculation included four categories, comprising the three label types (*A1_RES*, *A2_RES*, *OC_RES*) and *no label given*. Again, two calculations with different approaches were performed for the same reasons outlined previously, aiming to reduce the significant bias introduced by the *no label given* class. The first of these included all entity pairs, regardless of category and number of annotators, and resulted in a Fleiss Kappa of 0.94, almost perfect

IAA. The second approach omitted all entity pairs where all three annotators did not map a relation (*no label given*), again being by far the most common unanimous agreement and skewing the calculation. The Fleiss Kappa for this approach was 0.84, indicating that the agreement remained almost perfect.

3.5. Disagreement

We sought to resolve disagreement between the independently annotated datasets during the reconciliation phase. This phase involved comparing the sentences without unanimous annotation agreement, first resolving any differences that were due to erroneous annotation or deviation from the guidelines, with any remaining conflicts reviewed and discussed by all three annotators. Once all disagreements in a sentence were resolved, it was added to the merged dataset of sentences with unanimous annotation agreement.

Disagreements that warranted discussion were borne out of differences in interpretation of the annotation guidelines and in expert opinion on what information is necessary for constructing an evidence table. To focus these debates on the objective of the corpus and limit annotation complexity, the two-part question in italics below was developed as a discussion reference.

What are the minimum label requirements for each task needed to:

- *Accurately represent each instance of the named entity categories?*
- *Divide the named entity instances into their correct respective positions of an evidence table?*

Resolutions for common and complex disagreements were defined as new rules, which we report in the next two sections, and were added to the annotation guidance (see Appendix). The updated guidance was then used for a final pass review of the merged dataset to produce the gold corpus.

3.5.1. Entity annotation disagreements

There were two common entity annotation disagreements, both related to the outcome category. The first of these was whether to include mentions of the time frame in which the outcome was to be measured as part of the outcome entity (see Figure 6). One argument was to omit timings and focus only on text describing the outcome in relation to its associated measure, limiting the length and complexity of this entity. In addition, outcome time frames are often mentioned separately from the main contiguous body of the

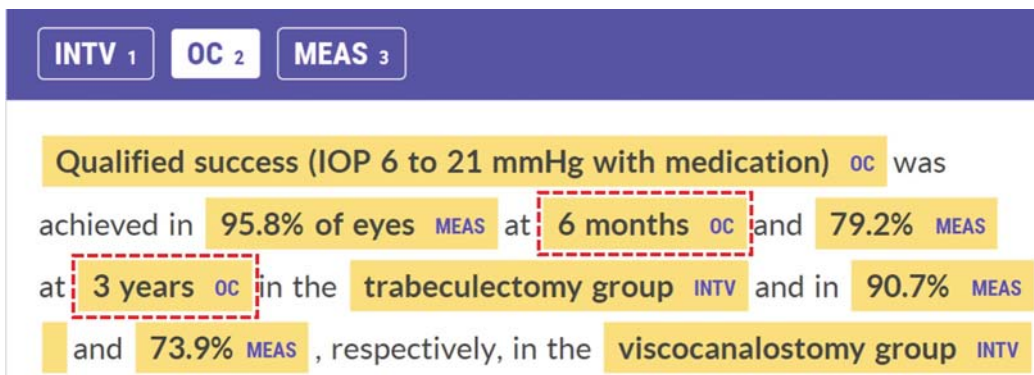


Figure 6: An annotated result sentence with two time frames reported (dotted red lines) for the assessed outcome, each of which have respective measures.

outcome description, adding complexity to the downstream relation mapping task. From a clinical evidence view, however, outcome time frames are essential to properly understand the action of a treatment (e.g. how a cancer treatment affects 5-year survival). Furthermore, there are common examples (Figure 6) throughout the corpus where measures are reported separately across multiple time frames (e.g. IOP reduction at 3 month and 6 months), making it impossible to separate these measures into a table without highlighting each time frame individually. Therefore, a decision was made to highlight outcome time frames in addition to their main description body, with both highlighted entities being mapped to the same respective measure where necessary.

The second common disagreement was with regard to phrases referring back to, or expanding on, separate outcome descriptions (see Figure 7). The annotation guidance instructed the annotators to highlight these referential phrases in sentences where the outcome description body was not present, but there was a lack of consensus over sentences where these references and their respective outcome body are both present. One argument, in line with the annotation guidance, was to highlight only the span of text that is used to semantically identify the measures related to the outcome (e.g. only highlighting *success* in “*success* was achieved by 20 patients when defined as an IOP lowering of 3 mmHg”).

This was decided against, under the rationale that referential phrases and outcome description are often equally important in identifying respective measures and putting what they are measuring into context. It was therefore

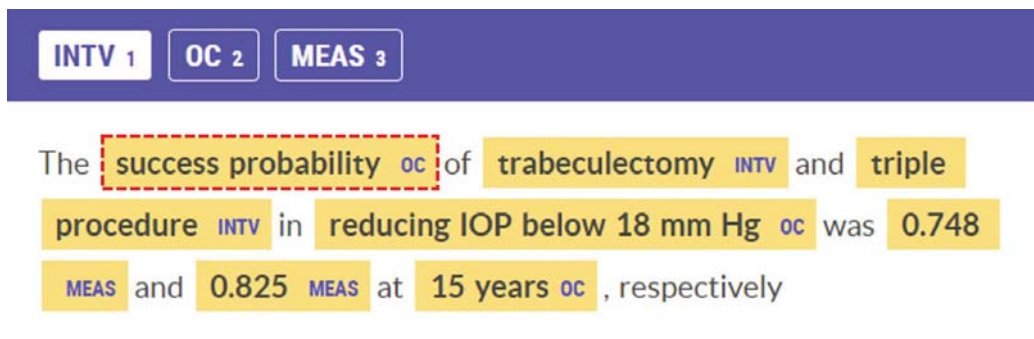


Figure 7: An annotated result sentence with a referential phrase (dotted red lines) referring back to an outcome description that occurs in the same sentence.

decided to highlight both referential phrases and their respective outcome description, with the two separately labelled entities each being mapped to any jointly respective measures.

3.5.2. Relation annotation disagreements

There was just one recurring disagreement during the reconciliation phase of the relation annotation task. This was with regard to the somewhat uncommon occurrence of both study arms achieving the same measurement for the same outcome (see Figure 8). This set of cases was overlooked in the guidance, and the concern was that mapping separate study arms to the same outcome measure would cause tabulation issues. It was noted, however, that the measure entity should just be repeated in each respective arm column of the table, resulting in the decision to map both arms to the single measure entity in these cases.

3.6. Dataset statistics

The total annotation statistics of the gold corpus can be seen in Table 1. The total size of the gold corpus was 558 sentences, comprising 3,541 entity annotations and 3,182 relation annotations.

Making up 41% of the entity labels, MEAS was the common annotation for this task, followed by INTV (31%) and OC (28%). This slight imbalance in favour of measurements is likely due to them being the only mandatory entity for a sentence to be included in the corpus, with most sentences including at least two MEAS labels.

The OC_RES label was the most common relation annotation by a significant margin at 58%, occurring at over twice the rate of the A1_RES (23%)

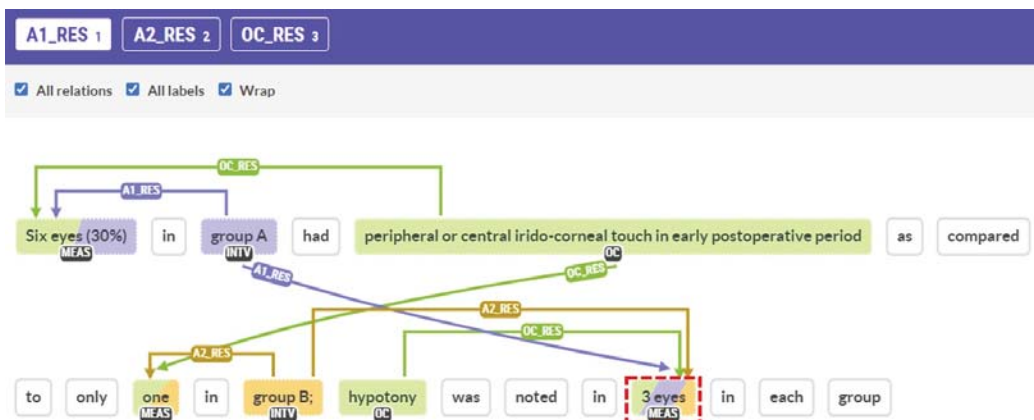


Figure 8: An annotated result sentence where both arms achieved the same measure (dotted red line) for the outcome *hypotony*.

	Entity labels			Relation labels		
	INTV	MEAS	OC	A1_RES	A2_RES	OC_RES
Count	990	1096	1455	717	630	1835
Proportion	0.28	0.31	0.41	0.23	0.20	0.58

Table 1: Total number and proportions of entity and relation annotations in the gold corpus.

and A2.RES (20%) labels. The primary reason for this imbalance is due to each outcome usually including at least two mappings to respective measures for each study arm, with the number of mappings doubling if the outcome time frame was annotated separately from its main description body.

4. System design and implementation

We discuss the implementation and design of our NLP pipeline for RCT result sentence tabulation in this section. Our system was designed to house two key components – a model for NER of INTV, OC and MEAS elements and a model for extracting the relations between them – both built on contextual BERT-based language representations. The output of these models is

then processed by a tabulation module, which returns result sentence tables in CSV format. In addition, we present Python functions and classes used in data collection from the EBM-NLP corpus, preprocessing and for adapting inbuilt spaCy components for our pipeline.

4.1. System architecture design and overview

Our extraction system was developed in Python 3.9.6 with the open-source NLP library spaCy (version 3.1). spaCy includes a variety of NLP tools for tasks ranging from rule-based sentence segmentation to BERT-based NER and was developed for building custom language processing pipelines [51, 52]. It also connects to the HuggingFace transformer library, which allowed us to import different BERT-based language representations for use in our models, including SciBERT, BioBERT and RoBERTa.

An overview of the full architecture of our study system can be seen in Figure 9, and consists of five key components. The first is a data collection module for retrieving abstracts from the EBM-NLP corpus and Trenta et al. study dataset, and processing them for annotation with Prodigy. The second component is a preprocessing module that prepares the annotated data for training our spaCy pipeline models. Performing the key tasks of our system, these models form our third and fourth components and are for NER and RE, respectively. The final, fifth component tabulates result sentences, applying both the NER and RE models sequentially to input text, outputting these tables as CSV files.

4.2. The data collection and preprocessing components

Built as a Python module, our data collection component has two parts: a class for screening and sorting abstracts from the EBM-NLP corpus as outlined in subsection 3.2.2; and a collection of functions for segmenting retrieved abstracts into sentences and processing them for annotation with Prodigy. The preprocessing component was also built as a Python module, and was used to process our gold-standard dataset after annotation into training and test input for our models.

Before preprocessing began, single domain subsets of the gold annotated dataset were created for cross-domain testing. The full gold corpus and individual domain datasets were then converted from the JSONL format output by Prodigy to the spaCy Doc format – the primary data-structure used by our models. This format natively comprises full sentence texts and their tokens, as well as labelled entities, but had to be extended with a custom

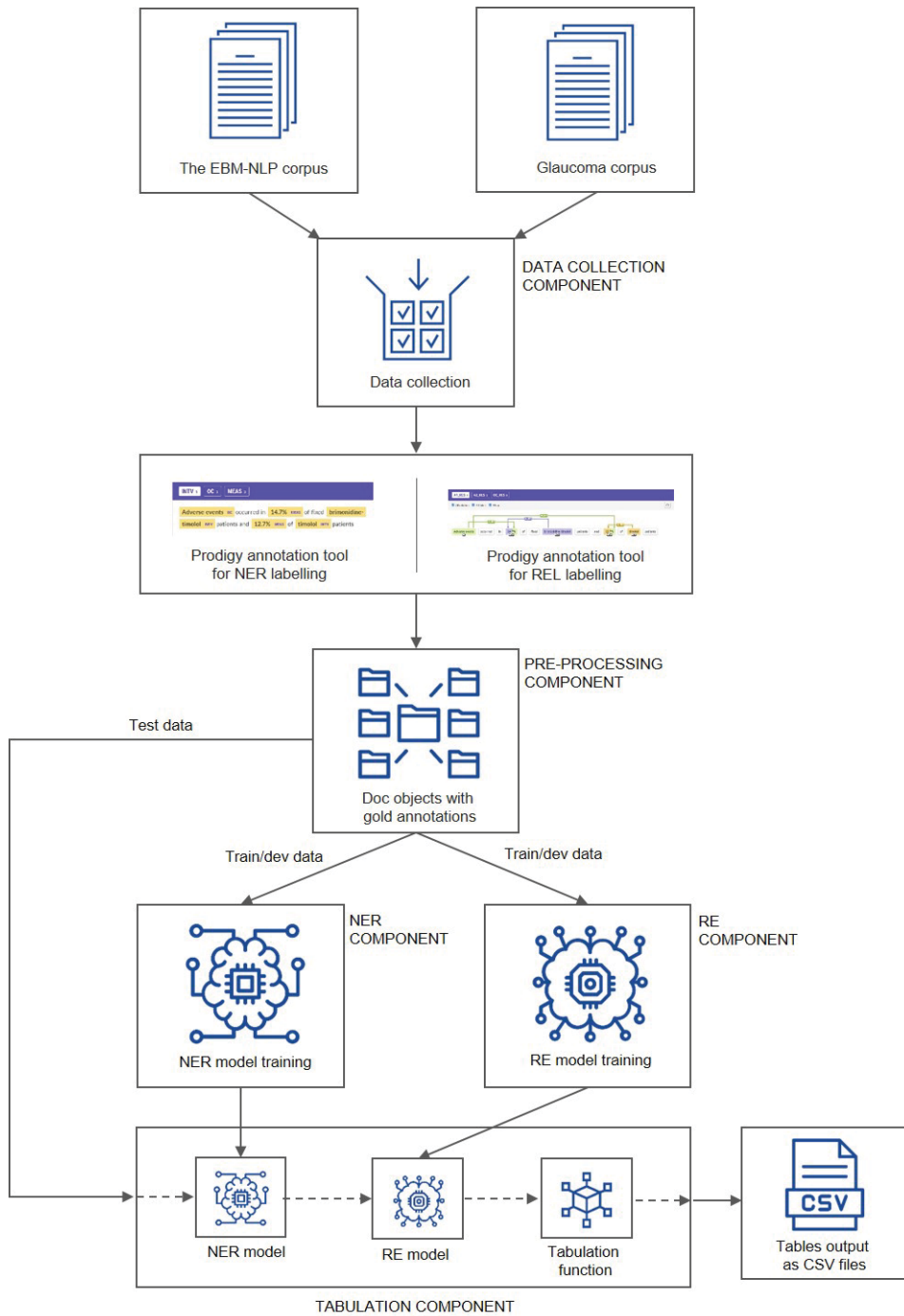


Figure 9: The architecture of our full study system.

attribute for relation labels. Before the training phase, all datasets were split into training (train), development (dev) and test sets, with sentence order randomised before the split.

4.3. The NER component

Our NER component was developed as a spaCy pipeline, with a language representation model feeding into the NER prediction model provided by the library. Importantly, this NER extractor works with interchangeable language representations, including pre-trained transformer-based models.

4.3.1. Design

The spaCy NER model has a transition-based parser architecture, inspired by the chunking model from Lample et al. [53], and uses the BILOU tagging scheme, which reportedly promotes better system performance than its simpler BIO counterpart [54]. In the Lample et al. model, a sequence of tokens is incrementally passed from a buffer to an entity stack for labelling or straight to the output list, with a greedy algorithm choosing the optimal action to take. The spaCy adaptation of this architecture has the following BILUO-directed actions to choose from:

Begin: Begin a new entity by adding token to stack

In: Continue the current entity by adding token to stack

Last: Entity label stack with current token as last word, move stack to output

Unit: Label current token as a single-word entity and move to output

Out: Move current token straight to output without marking as entity

We describe an optimal set of actions with an example sentence from the glaucoma domain in Figure 10. The scores for each possible action are calculated at each time step by feeding a representation of the current state of the stack and buffer to a multilayer perceptron, with the best action chosen until the algorithm reaches a termination state. The state representation is derived through combining the word embeddings of the tokens that make up the entity stack and the buffer, which are passed to the transition-based parser layer from an upstream language representation model, in our case a pre-trained transformer model.

Actions	Output	Stack	Buffer	Entity
	[]	[]	[interocular, pressure, in, the, latanoprost, arm]	
begin	[]	[interocular]	[pressure, in, the, latanoprost, arm]	
last	[(interocular pressure)-OC]	[]	[in, the, latanoprost, arm]	
out	[(interocular pressure)-OC, in]	[]	[the, latanoprost, arm]	(interocular pressure)-OC
out	[(interocular pressure)-OC, in, the]	[]	[latanoprost, arm]	
unit	[(interocular pressure)-OC, in, the, (latanoprost)-INTV]	[]	[arm]	(latanoprost)-INTV
out	[(interocular pressure)-OC, in, the, (latanoprost)-INTV, arm]	[]	[]	

Figure 10: Example of optimal actions taken as an input sequence is passed into the dependency parser.

Components in a spaCy pipeline are built with “listener” layers that allow them to receive word embeddings from layers made of interchangeable language models. An overview of the layers of this model design is outlined in Figure 11. The listener layer of the NER model is also used for back-propagation, passing the error gradients used to adjust the model weights back upstream to fine-tune the language representation layers. Gradients are calculated from a loss function that scores entity prediction errors per action against the target gold annotations, with all layers of the NER component being updated during the target task.

4.3.2. Implementation

The pipeline for our NER model was implemented as a spaCy config file, a feature of the library that facilitates and centralises pipeline design and modularity, where each layer of the model is defined, along with their hyper-parameters for training and the random seed for weight initialisation. There are a large number of variables that can be adjusted within the spaCy config system for customising models. Here, we describe the main config choices made in the implementation of our system, and refer the reader to the spaCy library documentation for settings not discussed [55].

For our transformer-based language representation layers, we experiment with three pre-trained BERT models: SciBERT and BioBERT, both extensively trained on medical study abstracts, and the generally trained model RoBERTa, which was included for comparison. We chose the “cased” version of all these models, where the capitalisation of words is considered, as these have been shown to perform better for NER, with evidence for this in the medical domain [56]. A separate config was created for each pre-trained model.

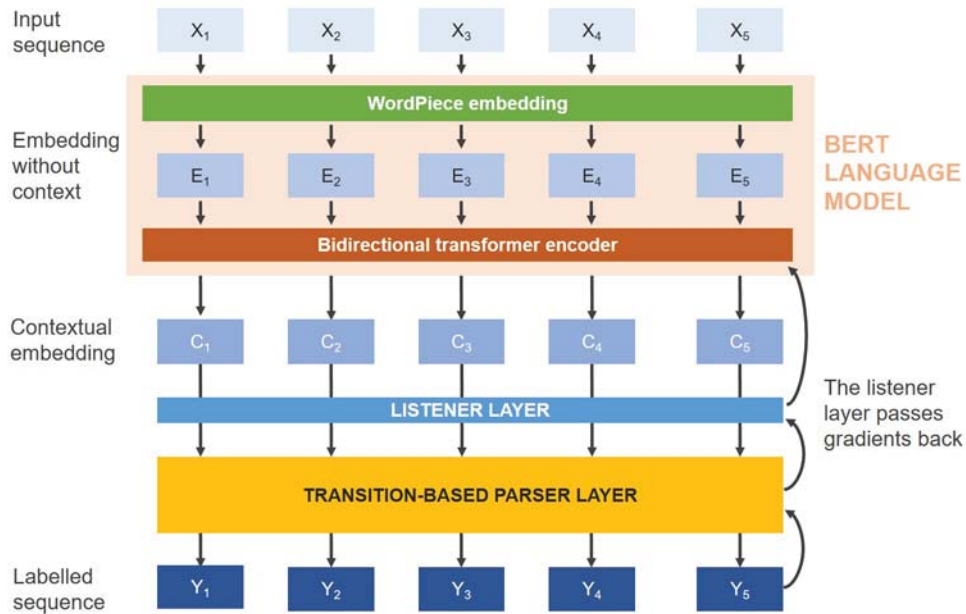


Figure 11: Schematic of the NER component architecture.

Our hyper-parameters for training were initially set to match those used by Devlin et al. [27] for fine-tuning the original BERT model. These included a training example batch size of 32, a dropout rate of 0.1 and a learning rate of $5e-5$ with Adam optimisation [57] used to fine-tune the system. After experimenting with a limited number of alternative hyper-parameter values (batches: 16, 32, 64, 128; drops: 0.1, 0.2, 0.3; learning rates: $5e-5$, $3e-5$, $2e-5$), in our final model configuration we adjusted the batch size to 64 and the dropout rate to 0.2, both gaining us incremental performance increases on our test data.

Models were trained with early stopping, activated by a patience parameter of 1,000 steps with no performance gains on the dev set, with a limit of 20,000 steps and no cap on epochs (the spaCy framework is uncommon in using steps rather than epochs for patience). A single machine was used to train all models, operating with a GeForce RTX 3080 GPU (10GB video RAM) and 16GB of RAM. Training time for fine-tuning on the full study corpus train set was around 6 minutes for each of the base models.

4.4. The RE component

Similar to our named-entity extractor, the RE component of our system was built as a spaCy pipeline, again using BERT-based language representations. However, as the library does not include an inbuilt RE model, we needed to implement it as a custom component of the pipeline, which we adapted from a spaCy project template [58].

4.4.1. Design

Designed with multiple layers, our RE model is built with a multi-label classification objective, where it scores a probability for each of our relation labels between entity pairs within the input sequence. As with our NER model, a listener layer passes word embeddings from a downstream transformer. In this case, however, two extra layers are required before classification. The first extracts word vectors for entities (checking the entity span labels of the Doc object), with the vectors of multi-token entities being “pooled” into a single vector, by taking their mean. The second pairs potential relation instances of entity pairs, in both directions to assess parent (i.e. subject) and child (i.e. object) status, outputting these instances as a tensor of the paired entity vectors. For classification, the output tensors are forwarded to a linear layer, and then to a sigmoid activation function for multi-label classification. The final output is a probability matrix for each entity pair across all of the defined relation labels, and both possible parent–child directions of the pair. A full overview of this architecture can be reviewed in Figure 12.

The relation-type label of entity pairs is identified by the highest probability in the matrix, with a hyper-parameter probability threshold value being set for the existence of any relation at all (e.g. if no probability in the matrix is above $P(0.5)$ then no relation is classed). This binary classification is performed on the matrix output of the model architecture above by a downstream component (see next section) or at system evaluation.

For training, the loss function is calculated as mean square error, taking the difference between predicted probabilities of the output matrix and gold-standard annotations, where probabilities are set to one for existing relationships and zero for non-existing relations. Again, back-propagation is used to train each layer of the pipeline, with the listener layer passing gradients back upstream to fine-tune the transformer language representation.

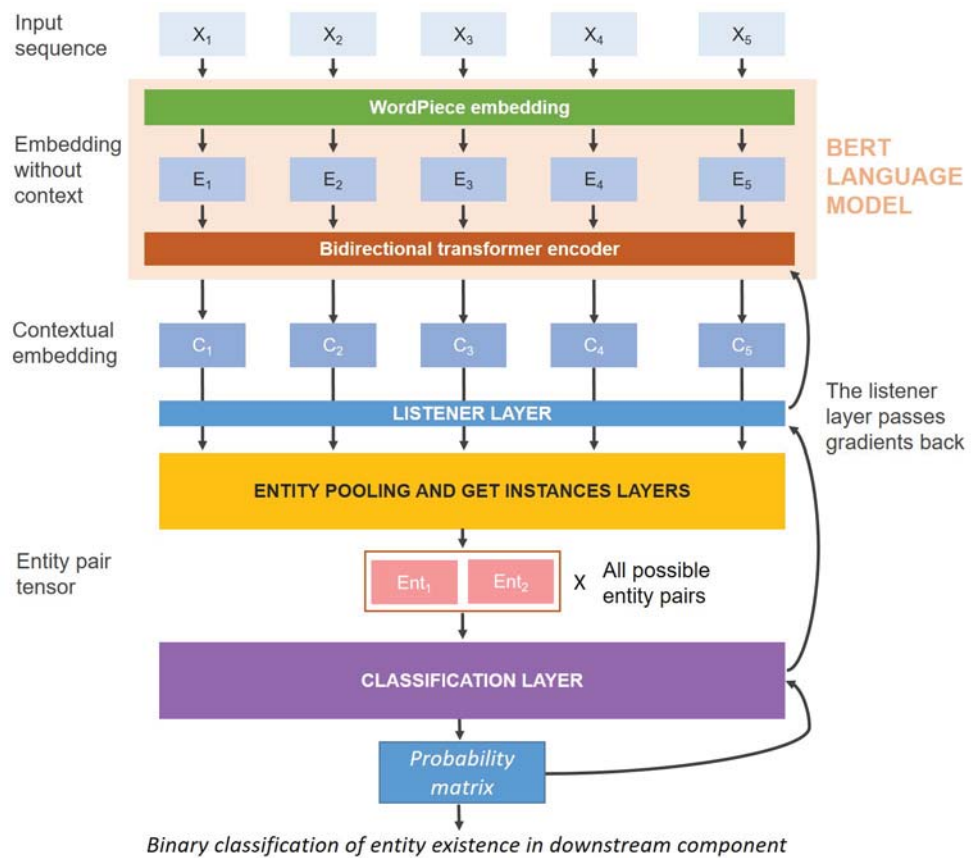


Figure 12: Schematic of the RE component architecture.

4.4.2. Implementation

Two Python modules underlie the RE component architecture outlined in the previous section: a module for the functional layers of the RE model, built using the Thinc library (an ML library cousin of spaCy, built on Pytorch), and a pipe module that integrates these layers for use in a spaCy pipeline. Functions within these modules are made accessible as model layers to the spaCy config system, which as with the NER component, was used to define and train the RE component.

We again created three separate configs for testing different language representation layers – one for each of the same pre-trained BERT models used in the NER component. For the layers of the RE models itself, we experimented with different criteria for selecting an entity pair instance for classification, first looking at a more restrictive approach that only selected pairs of interest ([OC and MEAS] or [INTV and MEAS]). However, this was found to reduce model recall performance by around 10% without improving precision. As a result of this exploration, the final model had a relaxed function for entity pair instances, retrieving all possible combinations of separate entities within a certain distance of 100 tokens of each other.

Hyper-parameter tuning was explored with the methodology we used for the NER component, which resulted in the selection of the same parameters for training (batch size: 64; dropout rate: 0.2; learning rate: 5e-5 [Adam Optimisation [57]]). The RE models were also trained using early stopping with the same step and epoch parameters, and on the same machine. Training time for fine-tuning on the full study corpus train set was around 5 minutes for each of the base models.

4.5. The tabulation component

The final component structures the full pipeline for the task of tabulating result sentences, using the joint predictions of our NER and RE components. Developed as a Python module, it processes batches of input sentences in Doc format over three stages. The initial two stages load the trained models to sequentially extract the entities and relations of the full input batch, adding these predictions to the Docs. These are then forwarded to the tabulation function, where the entity and relation labels are used to parse the sentence content into a structured table, which is outputted as a CSV file.

The tabulation function is where the probability threshold hyper-parameter for a detected entity–pair relation is set. As would be expected, a lower

threshold improves recall performance, while a higher one improves precision. We found that the optimum threshold as measured by F_1 score was 0.5, which was selected for the final system.

5. Evaluation

In this section, we report the results of our system evaluation across five IE tasks: NER, RE on gold-annotated entities, joint NER + RE, tabulation with exact tuple matching and tabulation with relaxed tuple matching. Test data performance with different BERT-based language representations is assessed for the full dataset across all domains. We also investigated our system’s ability to execute these tasks when trained with a varying number of examples and at generalising to unseen disease area domains. The section is closed with an error analysis of the system models.

5.1. Evaluation methodology

The system was evaluated against gold-standard labels in terms of precision (P), recall (R) and their harmonic mean, F -score – the latter being calculated with no emphasis on either of the former (F_1 score). For our multi-label classification tasks (NER, RE and joint NER + RE), due to our labels being slightly imbalanced (see subsection 3.6) and no priority differences between classes, we used micro-averaging to obtain F_1 scores, with true positives (tp), false positives (fp) and false negatives (fn) being summed globally across classes. Only NER could be evaluated using inbuilt spaCy library tools, which meant custom evaluation functions had to be developed for assessing the other system tasks. Here, we give a brief outline of how each of these was measured.

5.1.1. NER evaluation

The inbuilt spaCy NER evaluation function assesses tp , fp and fn on a per-entity basis with exact match. That is, it does not count partial matches of multi-token entities as tp . For the NER task scores, all of the exactly matched entities across classes are marked as tp , with the set of predicted not in gold counted as fp , and the set of gold not in predicted counted as fn .

5.1.2. RE evaluation on gold entities

To investigate the performance of its underlying model, the RE task was assessed on test data with gold-standard entity annotations. Predicted entity-pair relation labels (above probability threshold of 0.5) are counted as *tp* if they matched the gold relation label of the same entity pair tuple (ordered tuples: (a, b) and (b, a) represent different parent-child relationships). Predicted class labels not matching the class of the reference annotation are marked as *fp*. Those that do not breach the classification threshold, but were in the list of gold relation annotations are marked *fn*.

5.1.3. Joint NER + RE evaluation

The joint NER + RE task involves first predicting named-entities within an input sequence and then the relations between these predicted entity pairs. Evaluation of this task was somewhat more complex than the prior two, and was achieved through extending the RE evaluation function. First, all entity pairs are checked to see if they have relation annotations within the gold dataset. Those that do, are passed through to the RE evaluation function and assessed in the same way as previously described. Entity pairs not within the gold dataset are checked to see if they have relations above the prediction threshold, and classed as *fp* if they do. Predicted entities with no relation and not part of a gold annotated entity-pair (not all gold-labelled entities have relations) are evaluated with the prior NER evaluation methodology described.

5.1.4. Tabulation strict tuple matching

To assess the overall performance of the full system in tabulating RCT result sentence, tuples (order: *outcome*, *arm 1*, *arm 2*) from the predicted output CSV files were matched against tuples from corresponding (same input sentence) gold-standard CSV files, with two matching criteria. The strict criteria required for tuples to exactly match, both in order and entries, for the prediction to be marked as *tp*. Predicted tables without an exact match were counted as *fp*. The system outputs a CSV for every input sentence, so the number of predicted and gold CSVs always match, but their number of tuples (rows) could differ. Predicted tuples that were not within the gold CSV were counted as *fp*, while tuples in the gold CSV that were not in the predicted CSV were counted as *fn*.

5.1.5. Tabulation relaxed tuple matching

The second tuple-matching criterion was implemented after inspection of output CSVs found that, while not matching exactly, many predicted entities overlapped with gold CSV entities (see subsection 5.6). Inspired by the Type Matching criterion used by a number of biomedical entity studies [52, 59, 60], relaxed tuple matching marks a predicted tuple as *tp* if the entities have some token overlap with those of the corresponding gold tuple, and match its order.

5.2. Experimental settings

Investigation-specific partitions of the study corpus were made to train and test models for our experiments. The size, composition, and “train, dev, test” split of these partitions are outlined in the description of their respective experiment in the following sections.

The system architecture, model training implementations and hardware outlined section 4 remained the same for all investigations. Each experiment was conducted over ten runs, with model weights initialised with a different random seed for each run², implemented in independent spaCy config files, which we make available in our study repository. Mean performance scores for each experiment were calculated by averaging model performance scores across the ten runs and are reported with standard deviations (SD).

Repeated measures ANOVA was used to compare the mean F_1 score performance of the different base language models, BioBERT, SciBERT, and RoBERTa. For tasks with a significant ANOVA result, post-hoc pairwise comparisons were conducted to test for significance between each model pair using the Bonferroni correction to account for multiple comparisons. Statistical significance was set at an alpha threshold of 0.05.

5.3. System performance on the all-domain dataset with different language representations

Results across the five tasks for each BERT-based version of our system, trained and tested on the all-domains dataset (70:10:20 train, dev, test split), can be found in Table 2, and are given in terms of mean P , R and F_1 scores.

The highest mean F_1 scores for the independent tasks of NER and RE on gold entities were 0.86 ± 0.01 (BioBERT) and 0.80 ± 0.01 (BioBERT), respectively. For the dependent tasks, the highest scores were 0.73 ± 0.01 (BioBERT)

²Random seeds 3, 18, 23, 34, 67, 76, 89, 234, 263 and 452.

EMBEDDING MODEL	NER			RE (GOLD ENTITIES)			JOINT NER + RE			TABULATION (STRICT)			TABULATION (RELAXED)		
	<i>P</i>	<i>R</i>	<i>F</i> ₁	<i>P</i>	<i>R</i>	<i>F</i> ₁	<i>P</i>	<i>R</i>	<i>F</i> ₁	<i>P</i>	<i>R</i>	<i>F</i> ₁	<i>P</i>	<i>R</i>	<i>F</i> ₁
BioBERT	0.86 ± 0.01	0.87 ± 0.01	0.86 ± 0.01	0.78 ± 0.04	0.83 ± 0.04	0.80 ± 0.01	0.70 ± 0.03	0.78 ± 0.03	0.73 ± 0.01	0.56 ± 0.02	0.82 ± 0.02	0.67 ± 0.02	0.76 ± 0.02	0.86 ± 0.02	0.81 ± 0.02
SciBERT	0.85 ± 0.01	0.86 ± 0.01	0.85 ± 0.01	0.78 ± 0.02	0.80 ± 0.03	0.79 ± 0.01	0.69 ± 0.02	0.75 ± 0.02	0.72 ± 0.01	0.53 ± 0.02	0.79 ± 0.03	0.63 ± 0.02	0.72 ± 0.03	0.84 ± 0.02	0.78 ± 0.02
RoBERTa	0.86 ± 0.01	0.85 ± 0.01	0.86 ± 0.01	0.77 ± 0.03	0.80 ± 0.04	0.79 ± 0.01	0.70 ± 0.03	0.76 ± 0.03	0.73 ± 0.01	0.52 ± 0.03	0.80 ± 0.02	0.63 ± 0.02	0.72 ± 0.04	0.85 ± 0.02	0.78 ± 0.02

Values in bold indicate the language model that scored the highest on a task-specific metric.

Table 2: System performance results of the five IE tasks for each BERT-based language representation trained on the all-domains test set. We report mean and SD values from 10 runs.

for joint NER + RE and 0.67 ± 0.02 (BioBERT) for tabulation with strict entity matching. As these latter two tasks are downstream in the pipeline, the first dependent on the NER component and the second dependent on joint NER + RE, their respective performance decreases are to be expected. The highest F_1 score for relaxed tabulation, however, was 0.81 ± 0.02 (BioBERT), likely reflecting the impact of loosening performance dependency on the upstream task of NER.

Repeated measures ANOVA revealed significant differences between the mean F_1 scores of the base language models across four of the five tasks: RE on gold entities [$F(2,18)=9.97$, $p=0.0012$], joint NER + RE [$F(2,18)=4.18$, $p=0.0322$], strict tabulation [$F(2,18)=17.26$, $p=0.0001$] and relaxed tabulation [$F(2,18)=10.36$, $p=0.0010$].

Post-hoc pairwise comparison across these tasks showed that the mean F_1 scores for BioBERT reported above were significantly higher for RE on gold entities than those of SciBERT [0.79 ± 0.01 , $p=0.0075$] and RoBERTa [0.79 ± 0.01 , $p=0.0043$]. BioBERT also achieved significantly higher mean scores for strict tabulation than both SciBERT [0.63 ± 0.02 , $p=0.0007$] and RoBERTa [0.63 ± 0.02 , $p=0.0002$], as well as for relaxed tabulation [SciBERT: 0.78 ± 0.02 , $p=0.0041$; RoBERTa: 0.78 ± 0.02 , $p=0.0004$]. However, no significant pairwise results were found between any of the model architectures for the joint NER + RE task, despite a significant ANOVA reading, potentially due to the conservative nature of the Bonferroni correction. There were no significant differences between SciBERT and RoBERTa on any tasks.

These results only partially follow our expectations set by the litera-

ture, where domain-specific BERT models tend to perform better on in-domain tasks [44, 45], with SciBERT’s performance being almost identical to RoBERTa’s across all tasks. One explanation could be that RoBERTa has an improved training procedure that has been shown to improve its performance versus the original BERT system [43], potentially offsetting SciBERT’s domain-specific performance gains. Another reason, and perhaps more likely, is that the pre-training dataset of RoBERTa is around ten times the size of that of SciBERT, which has been demonstrated to be a key hyperparameter for downstream tasks [43]. BioBERT has a pre-training corpora around five times larger than that of SciBERT, which may be why it frequently outperformed the other models, balancing domain-specificity with pre-training dataset size. Finding an optimum threshold between these two factors may be an interesting area for future exploration.

The performance differences between the base language models on the independent classification tasks were marginal, with mean F_1 score differences not exceeding 0.1 points. These results were only significant for the RE on gold entities task in favour of BioBERT in both pairwise comparisons, revealing that this base model’s advantage may lie in its ability to work with relational context between medical terminology. For the tabulation tasks, the performance gap between BioBERT and the other two architectures widens, with the F_1 scores for BioBERT 0.04 points higher than those of both SciBERT and RoBERTa for strict tabulation and 0.03 points higher for relaxed. These findings indicate that small differences in performance at the independent task stages are amplified downstream in our system.

5.3.1. NER performance on individual class labels

System performance at classifying the individual NER class labels of the all-domains test set can be viewed in Table 3.

The INTV label was the highest performing classification with a mean F_1 score of 0.93 ± 0.01 (BioBERT), followed by MEAS with 0.89 ± 0.01 (SciBERT) and, lastly, OC with 0.76 ± 0.02 (BioBERT). These differences are similar to those in the literature, with outcomes tending to be one of the harder PICO elements to classify, likely due to high variation in entity length and poor inter-annotator agreement on their boundaries [29]. Furthermore, INTV and MEAS entities tended to be shorter and follow more consistent lexical forms.

Comparing individual NER class label performance across the base model architectures, significance was only found for the OC label [$F(2,18)=5.51$,

EMBEDDING MODEL	OC			INTV			MEAS		
	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>
BioBERT	0.76 ±0.02	0.75 ±0.02	0.76 ±0.02	0.93 ±0.01	0.92 ±0.01	0.93 ±0.01	0.88 ±0.02	0.90 ±0.02	0.89 ±0.02
SciBERT	0.72 ±0.02	0.73 ±0.03	0.72 ±0.02	0.92 ±0.02	0.91 ±0.02	0.92 ±0.01	0.88 ±0.02	0.90 ±0.01	0.89 ±0.01
RoBERTa	0.75 ±0.03	0.73 ±0.03	0.74 ±0.02	0.94 ±0.01	0.91 ±0.02	0.92 ±0.01	0.87 ±0.02	0.89 ±0.02	0.88 ±0.02

Values in bold indicate the language model that scored the highest on a task-specific metric.

Table 3: Performance of the NER component for individual entity labels on the all-domains test. We report mean and SD values from 10 runs.

$p=0.0136$] by repeated measures ANOVA. This result went against our prediction that RoBERTa as the general model would struggle with the INTV and MEAS labels, due to the former being composed of highly domain-specific tokens, such as generic drug names, and the latter being domain-specific constructions of numbers and units.

Post-hoc pairwise comparison between the models for the OC label found a significant difference between the performance of BioBERT versus SciBERT [0.72 ± 0.02 , $p=0.0049$], with the former having a 0.04 point higher mean F_1 score than the latter. Again, this may be due to its higher pre-training dataset size. No significance was found for the other two pairwise comparisons.

5.3.2. RE performance on individual class labels

We present system performance at classifying the individual RE class labels when given gold-standard entities on the all-domains test set in Table 4.

Best classification performance was the same for both the A1_RES and A2_RES labels (interventions mapped to their respective measures by arm) at 0.90 ± 0.02 mean F_1 score (both BioBERT), dropping to 0.73 ± 0.01 (BioBERT) for the OC_RES label (outcome mapped to respective measure). As gold-standard entities are provided to the RE component for evaluating this task, it is unclear whether the performance drop for the OC_RES label is due to the outcome entity itself or the context linking it to its measure (or, perhaps more likely, a combination of the two).

Significant differences between mean F_1 scores were found between the

EMBEDDING MODEL	A1_RES			A2_RES			OC_RES		
	<i>P</i>	<i>R</i>	<i>F</i> ₁	<i>P</i>	<i>R</i>	<i>F</i> ₁	<i>P</i>	<i>R</i>	<i>F</i> ₁
BioBERT	0.87 ±0.02	0.92 ±0.03	0.90 ±0.02	0.86 ±0.04	0.96 ±0.02	0.90 ±0.02	0.72 ±0.06	0.75 ±0.07	0.73 ±0.01
SciBERT	0.86 ±0.04	0.87 ±0.03	0.86 ±0.02	0.84 ±0.05	0.91 ±0.03	0.87 ±0.02	0.73 ±0.03	0.72 ±0.05	0.72 ±0.02
RoBERTa	0.88 ±0.04	0.86 ±0.03	0.87 ±0.01	0.86 ±0.03	0.91 ±0.03	0.88 ±0.01	0.70 ±0.05	0.75 ±0.06	0.72 ±0.01

Values in bold indicate the language model that scored the highest on a task-specific metric.

Table 4: Performance of the RE component for individual relation labels on the all-domains test set. We report mean and SD values from 10 runs.

three architectures for two of the three RE labels, A1_RES [$F(2,18)=10.24$, $p=0.0011$] and A2_RES [$F(2,18)=4.73$, $p=0.0224$]. Comparing the language representations on these two labels with pairwise comparisons, BioBERT performed significantly better on the A1_RES label than both SciBERT [0.86 ± 0.02 , $p=0.0049$] and RoBERTa [0.87 ± 0.01 , $p=0.0068$] by 0.04 and 0.03 mean F_1 score points, respectively. BioBERT also performed significantly better than SciBERT on the A2_RES label [0.87 ± 0.02 , $p=0.012$] with a 0.03 point higher mean F_1 score, but was not significantly better than RoBERTa. There was no significant difference between SciBERT and RoBERTa on either of the labels.

We hypothesise that BioBERT’s superiority on A1_RES and A2_RES labels may be due to RE placing a greater emphasis on the contextual words around named entities, such as verbs and prepositions, which hold syntactic information such as subject–object ownership (e.g. [INTV] achieved [OC] of [MEAS]), and are generally not domain-specific. Therefore, being pre-trained on the original general BERT corpora in addition to medical literature potentially positions BioBERT more favourably to identify relations between medical entities. This would also explain the comparatively poor performance of BioBERT in mapping outcome relations, where the proportion of domain-specific tokens in outcome entities (generally formed of descriptive phrases that mix medical and non-medical words) is lower than in intervention entities (primarily formed of specific drug names).

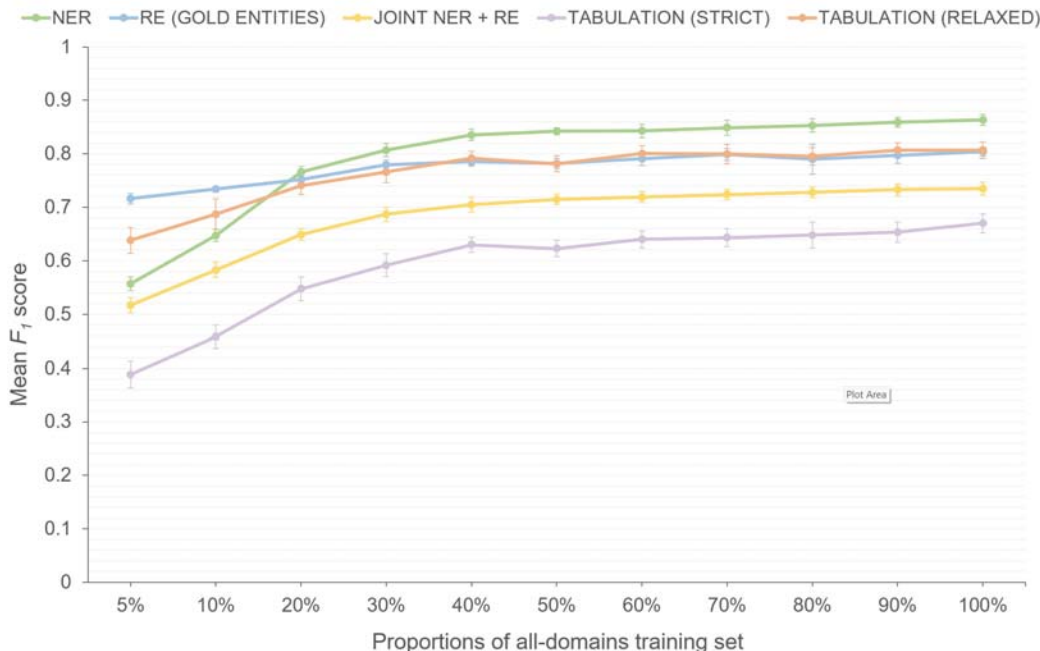


Figure 13: System (BioBERT language model) performance results across the five IE tasks after training on varying proportions of the all-domains training set. We report mean values from 10 runs and SD as error bars.

5.4. All-domain performance with respect to the number of training examples

Figure 13 shows the impact of varying the number of training examples on the mean F_1 scores of the five evaluation tasks.

This investigation was conducted by taking proportions of the all-domains training set from 5% to 10% and then in increments of 10% to maximum, with an NER and RE component trained on each size stratification. As the base language model that performed best on all five tasks, particularly the strict tabulation task, we selected the BioBERT model for this experiment, as well as out-of-domain testing and domain comparison testing in the next sections.

Across NER and all of the dependent tasks, mean F_1 score improved as the proportion of the original training set was increased, rapidly from 5% of the set to around 30%, where performance gains began to plateau with diminished returns from 40% to 100%. For the independent RE on gold entities task, performance at 5% was still surprisingly high with a mean F_1 score of 0.72 ± 0.01 , demonstrating that relatively high performance for this

task can be achieved by fine-tuning BERT-based models with as few as 27 example sentences.

The NER, joint NER + RE and strict tabulation tasks were the most sensitive to size adjustments in training samples, with the greatest performances dips at 5% and the steepest gradients of ascent as more training examples were introduced. This is unsurprising, particularly for joint NER + RE and strict tabulation, which are downstream of the independent tasks. Upstream errors propagate further errors as they are passed down the pipeline, reducing the performance of each downstream task (e.g. one missed entity \rightarrow two related relations missed \rightarrow three related tuples missed). This may explain the drop in performance across all downstream components at 50% training set proportion, with the slight decrease in RE performance at this stratification being amplified in each of the tabulation tasks.

From 20% of the all-domains training set, the relative performance differences between tasks begin to resemble those of the full set. Interestingly, relaxed tabulation closely tracks the performance of the RE on gold entities task from this point to the full training set, perhaps due to being less strictly dependent on the performance of the NER component.

5.5. Generalisation performance across domains

In this section we explore generalisation of our system across disease areas, redistributing the all-domains dataset into train and test sets that are separated by domain (with the dev set including the same domains as the train set), and retraining our BERT-based (BioBERT) NER and RE components for each experiment.

5.5.1. One unseen disease area domain

Figure 14 shows mean F_1 score performance on each of the domain areas when removed from the all-domains dataset and used as an unseen test set, with the models being trained on a randomised set of all other domains (e.g. *train set domains*: blood cancer, cardiovascular disease, diabetes, glaucoma; *test set domain*: autism). We report relatively comparable performance across domains on all five tasks, with the unseen cardiovascular disease test set having the greatest number of lowest scoring tasks (4 tasks). This result is somewhat unexpected, as cardiovascular disease is the broadest domain and has significant overlap with the other domains in terminology, particularly diabetes. It may be that this domain gives the system a performance advantage on narrower, cardiovascular-related disease domains, while examples

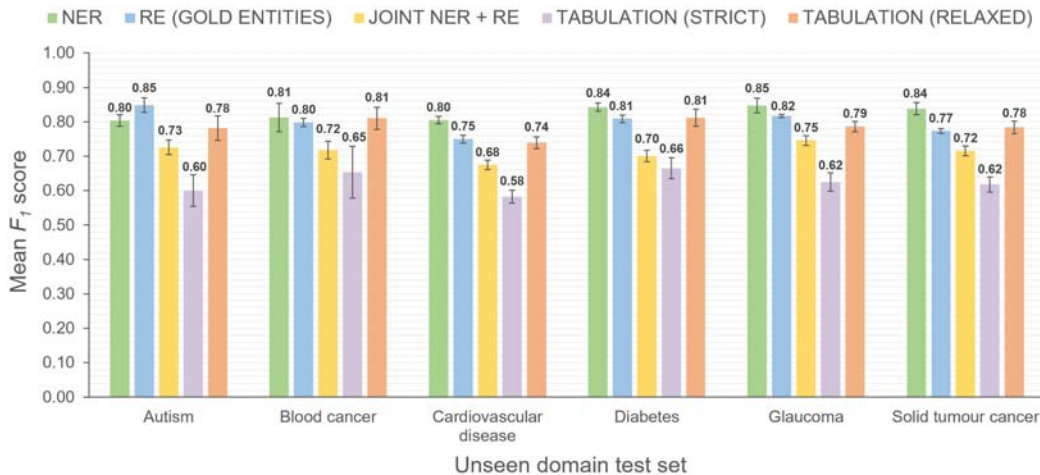


Figure 14: System (BioBERT language model) performance results of the five IE tasks on different unseen disease-area domain test sets. We report mean values from 10 runs and SD as error bars.

from these more specific domains do not impart enough information about the broad disease area as a whole.

Interestingly, autism was the only disease area where performance on the RE on gold entities task was higher than NER. This may be due to all five of the other disease area being more widely researched than autism, with a wider pool of studies across a wider range of patient groups. Sentences from these unseen domain test sets may therefore have greater variance in the way they report results, with lower consistency in word dependencies impacting on RE performance.

5.5.2. Varying the number of training domains

To explore the importance of training with a domain variety, we trained our system on datasets composed of abstract sentences from a varying number of disease areas.

Three training datasets were composed for this investigation: glaucoma alone (210 randomised examples); glaucoma and cardiovascular disease (105:105 randomised examples); and glaucoma, cardiovascular disease and solid tumour cancer (70:70:70 randomised examples). These domains were selected based on their size, allowing for training sets large enough ($\geq 30\%$ of all-domain set) to offset the effect of the number of examples on performance.

We show the results of training on these varying domain sets in Figure

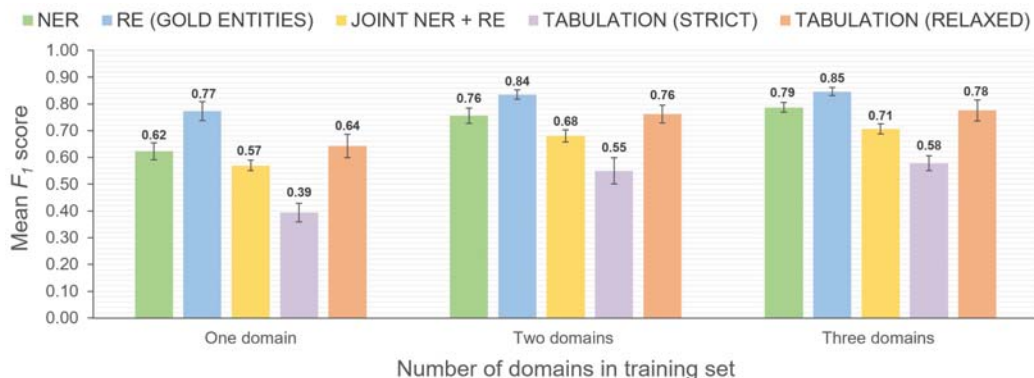


Figure 15: System (BioBERT language model) performance on the autism test set as training domain variety increases. We report mean values from 10 runs and SD as error bars.

15, where the autism test set was chosen as the least-related disease area. From one to two domains, all five system tasks see substantial mean F_1 score increases, including a 0.16 point increase for strict tabulation. From two to three domains, we see only moderate performance increases across the tasks, with just 0.03 point improvement on strict tabulation. This indicates that only two domains may be enough for reasonable out-of-domain system performance, with returns diminishing as domains are added beyond this point. More data is needed across more domains, however, to confirm this in a future study.

5.5.3. Comparison of single domain performance

We compare the performance of our system when trained and tested on individual domain disease areas in Figure 16.

We again chose the largest three domains for comparison, for the same reasons as outlined in the varying domains section, with each capped to 112 examples (70:10:20 train, dev, test split) for meaningful comparison.

Glaucoma was the strongest performing dataset out of the three domains, with mean F_1 scores similar or exceeding (RE on gold entities task) those achieved by the system on the all-domains dataset.

Cardiovascular disease performed relatively poorly on the NER (0.72 ± 0.03) and RE on gold entity tasks (0.61 ± 0.03) when compared with the other two disease areas, as well as on joint NER + RE extraction (0.61 ± 0.04) although by a lower margin. As discussed in subsection 5.3.1, cardiovascular dis-

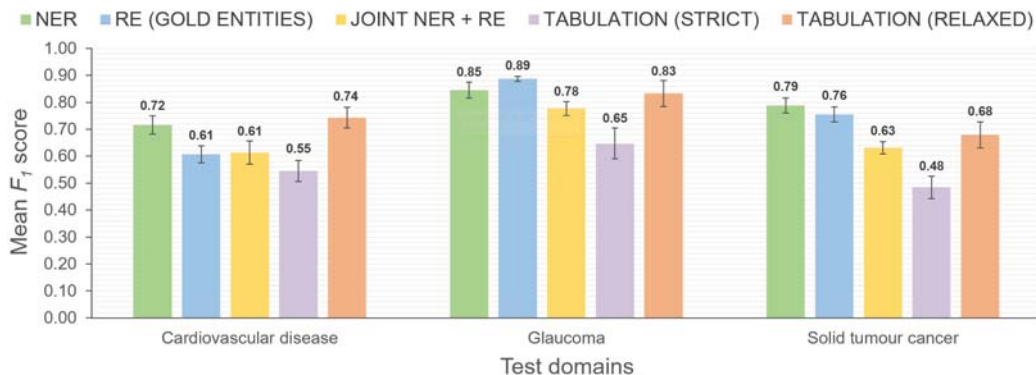


Figure 16: Comparison of system (BioBERT language model) performance after training on different disease areas with the same number of examples, tested on an unseen test set from same domain. We report mean values from 10 runs and SD as error bars.

ease is the broadest domain, grouping a number of different diseases, likely giving it higher variance in result reporting and sentence construction. This could account for the lower RE performance, with less consistent word dependencies between the entity pairs for the model to learn from.

Solid tumour cancer achieved higher NER and RE on gold entity performance scores than cardiovascular disease, but had the lowest scores for the tabulation tasks out of all three domains. Again, this makes sense when considering the disease area. In terms of NER, RCT objectives in oncology are often limited to overall survival, progression-free survival, response rate and safety, while many interventions, particularly chemotherapy, have been in use for years and feature repeatedly across studies. However, study structures in oncology are complex, with different combinations of interventions given and tested across cycles, making it difficult for the system to extract all of the correct information into correctly ordered tuples.

5.6. Error analysis

In this section, we take a closer look at the common errors occurring within the two main IE models in our system.

5.6.1. NER errors

We present a normalised confusion matrix for our discussion of NER errors in Figure 17, which displays token level classifications of the BioBERT-based component on the all-domains dataset.

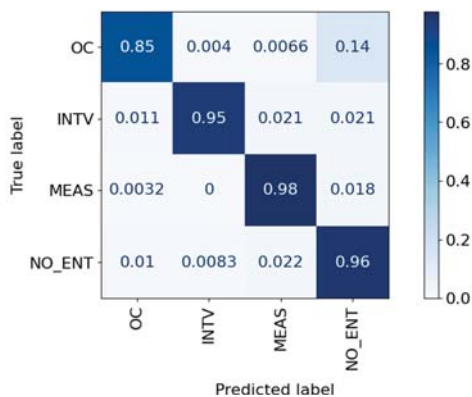


Figure 17: Normalised confusion matrix of token-level NER predictions on the all-domains test set.

Overall , 58 % of the 36 patients assigned to AMB successfully completed prophylaxis compared with 80 % of the 41 patients assigned to FLU (< 0.05)

OC

Figure 18: Incomplete OC label error. Gold-standard OC labels are in green, while the predicated label is in red.

Incorrect negative classifications are the most common type of misclassification error, particularly affecting the OC label. This is reflected in the commonly observed issue of the model incompletely identifying the tokens in these entities (see Figure 18), likely due to the reasons discussed in subsubsection 5.3.1, with outcome entities variable both in length and annotator boundary agreement.

Another commonly observed error, reflected in Figure 18, is the misclassification of numeric tokens with the MEAS label, primarily occurring with intervention entities which includes concentration values (see Figure 19), or non-entity numbers. For the former issue, a post-processing rule that looks one token ahead of interventions for concentration values may be worth investigating.

5.6.2. RE errors

A normalised confusion matrix is also presented for RE errors in Figure 20, which displays the entity-pair RE classifications of the BioBERT-based component on the all-domains dataset.

INTV MEAS
Latanoprost 0.005% once daily reduced IOP (+/- SEM) more effectively than latanoprost 0.0015%
twice daily (9.8 +/- 0.9 mm Hg and 6.7 +/- 0.9 mm Hg, respectively)

Figure 19: MEAS misclassification error. Gold-standard INTV labels are in green, while the predicated label is in red.

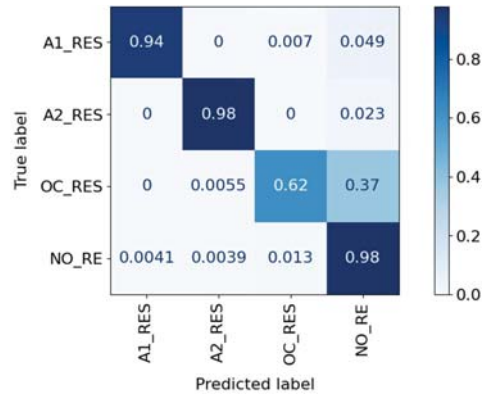


Figure 20: Normalised confusion matrix of entity-pair RE predictions on the all-domains test set.

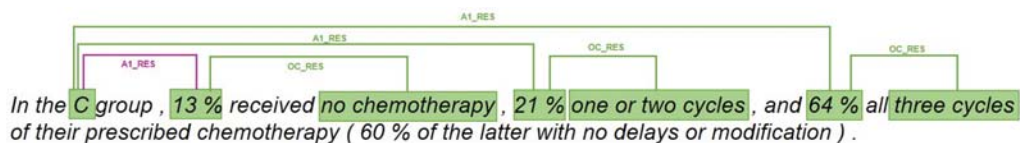


Figure 21: Negative relation classification error. The model only successfully recognises one relation (pink), and misses five (green).

Negative classification of existing relations is the most common type of error, and again occurs most frequently in outcomes and their respective measures. As discussed previously, in disease areas like solid tumour cancer, trial structure can make mapping relations between outcomes and their measures particularly difficult. In Figure 21 we can see a case of this, where the system fails to categorise a relation between the number of chemotherapy cycles and the proportion of patients who received them.

While reducing the probability threshold of relation classification would reduce the *fn* error rate, during the implementation stage it was found to impact too greatly on tabulation precision. Presumably, this is because the process of sorting entities through their relations is more sensitive to increases in *fp* errors, where one incorrect entity results in a *fp* for the whole tuple.

At the current threshold, positive relationship classes tend not to be misclassified as other positive relationships, with A1_RES or A2_RES classes very occasionally being classified as OC_RES, and vice versa. Surprisingly, respective arm-1- and arm-2-measure relationships had zero instances of being miscast as each other, considering the similarity of these relations.

6. Conclusion

In this study, we address a key problem in automating systematic reviews with a system that can tabulate result sentences from published RCT abstracts. Our NLP pipeline achieves this task in three stages: extracting interventions, outcomes and their measures as named entities; identifying the respective relations between them; and using this relational data to sort entities into the appropriate *outcome*, *arm 1* and *arm 2* columns of an evidence table.

For the two NLP tasks central to our system, NER and RE, we took a transfer learning approach. Through fine-tuning BERT-based transformer models, pre-trained on billions of domain-specific tokens, our system embeds

and encodes input sentences into context-rich language representations for these classification tasks. We have also developed an extensive corpus of over 550 RCT result sentences across six disease areas for training these models and testing them, as well as our whole system.

6.1. Results discussion

In its primary task of tabulation, our system (BioBERT-based models) achieved a mean F_1 score of 0.67 ± 0.02 with strict entity matching and 0.81 ± 0.02 with relaxed. If we consider the general NLP tasks of the system against the literature, our mean F_1 scores are relatively high for domain-specific NER (0.86 ± 0.01), RE (0.80 ± 0.01) and joint NER + RE (0.73 ± 0.01), on a test set that includes six different disease area domains; however, the data inclusion criteria of our study must be considered when making such comparisons. We also found that our system generalised well when tested on disease area domains unseen during training, with as few as two domains needed within the training set to achieve this performance. Furthermore, our overall results obtained on the all-domains dataset can be achieved by fine-tuning the layers of our models with relatively small training sets of around 170 example sentences.

In the context of the Trenta et al. [10] study, from which we derived these criteria and our glaucoma dataset, we see how much the field has progressed thanks to the recent innovations in contextual language representation. Our pipeline not only differentiates entities that their *one hot vector*-based classification system performed poorly on, such as outcome measures by study arm, but also extracts full entity spans rather than just single-token syntactic heads.

A more contemporary study by Mutinda et al. [16] has also investigated automation of the systematic review process through tabulation and achieved promising results with BERT-based language models, albeit with wide differences in approach to system architecture. Unlike our study, they utilised NER alone, training a single model on a corpus of full abstracts from a single disease area (breast cancer) with a comprehensive set of PICO entity labels, including differentiation of interventions into study and control treatments, as well as outcomes measures into specific categories such as mean, absolute and percentage values. A rule-based method was then used to separate measure and intervention groups by outcomes into structured tables, with the system limited at this stage to absolute measure values for calculating statistics. In comparison, our system has a far more limited NER stage, but uses

an RE model, rather than pre-determined rules, to map entities for the task of tabulation. Both systems scored comparable F_1 scores for NER; however, Mutinda et al. do not report F_1 scores for end-to-end tabulation performance, making it difficult to compare system performance for this task. A notable error highlighted in their study is the misclassification of entity classes, such as mixing study treatments with controls. Our system had limited misclassification errors between entity labels (see section 5.6), potentially highlighting the benefits of decomposing this objective across separate models for NER and RE. Indeed, Mutinda et al. highlight RE as an area for future exploration in their discussion to overcome the limitations of rule-based tabulation.

6.2. Limitations

In terms of the limitations of our pipeline, a number are worth discussing here. Firstly, our system has a narrow focus on measure entities, omitting the comparative statistics between arm respective values, which are an essential part of evidence tables. As they have been addressed by other studies, such as that of Kang et al. [20], we made a decision at the start of the study to focus only on entities that could be clearly divided into table columns. However, the system could be easily extended to include these measures, with a new comparative statistics entity class that could be related back to the specific outcome entity. Our system also had broad NER classification classes compared with studies such that of Mutinda et al. [16], in part due to our goal of decomposing our objective across two models. Extending their dataset with relation annotations to retrain and test our system may be an interesting area for further exploration.

Another limitation was the constraint of our dataset to the sentence level. Although interventions, outcomes and their respective measures occur frequently together in sentences, multi-sentence constructions are not uncommon, and represent a potential blind spot for our system. However, while we chose to operate at the sentence level, this is not an inbuilt limitation of our architecture. BERT language representations have a window of sequence length within which they operate best – a potential solution to this limitation may be to include as many sentences within this window as possible (without truncating the last inclusion). In addition, our corpus was restricted to result sentences from study abstracts, which rarely contain all of the information needed for a systematic review.

To limit the scope of our investigation to the goal of automated tabulation, the inclusion criteria for our corpus restricted it to clinical trials with

a two arm study design and abstracts with at least one result sentence that included at least one outcome and/or study arm, and a clear, numerical measure. Many of the abstracts for published clinical trials do not conform to these rules: trials with three or more arms are not uncommon and some abstracts, particularly for older publications, do not include clear, numerical measures. While this represents a significant gap between our current system and manual data extraction for systematic reviews, these restrictions are again not enforced by our system architecture. There is potential for future iterations of this system to be trained and tested on less restrictive datasets.

Long entities represented a challenge for our system, being harder to classify, particularly outcomes containing hierarchies of sub-entities, which were often predicted individually. A solution based on our existing approach may be to further decompose these entities into their constitute parts, and share the problem with the RE component. For example, for the outcome entity, “*reduction in intraocular pressure of at least 18 mm Hg*”, annotators could label “*reduction in intraocular pressure*” as the main entity, while “*at least 18 mm Hg*” could be tagged as a qualifier, with a relationship mapped between them for the RE component to resolve.

Training and run-time efficiency were not considered during the development of our system; however, it should be noted that transformers are relatively resource-intensive models. Transformers require high specification GPU hardware to run, and two of these models are included in our system. While using a single transformer for both classification models was explored, the observed reduction in performance (overall f_1 score drops of up to 0.2) was consider too great for this approach to be of value. Nevertheless, we would argue the hardware and energy costs related to our system would be outweighed by reductions in the human labour cost of systematic reviews.

6.3. Future research

Future research of our system could investigate expanding it to process inputs with with less restrictive inclusion criteria, such as including sentences from studies with more than two arms, as well as extending its extraction scope to PICO elements from all abstract sentences. If extended to full abstracts, it may be worth exploring the addition of a question–answer sentence pairing module to the system pipeline (the second task BERT language representations are trained on [27]) for linking sentences stating primary outcomes to their respective result sentences. The distinction between a study’s primary and secondary outcomes is an important one for a systematic review,

and is usually defined in the methodology section of an abstract. It would also be interesting to test the system beyond abstracts, which only represent the first stage of a systematic review. Once a study has been accepted for inclusion, information from the full published paper must be extracted to complete the evidence table. As our system accepts sentences as inputs, it is not inconceivable that it could work well with the sentences from the results section of a full paper.

Considering more advanced areas of research, our system or parts of it, could be used as a component in frameworks for the automation of evidence-based clinical recommendations. More specifically, tabulated result sentences could be used to synthesise claims of the logical arguments that underlie these recommendations. For more information on argumentation with results of clinical trials, see [8].

6.4. Concluding remarks

While a great amount of work still remains in automating systematic reviews of clinical evidence, our study has shown that a key barrier – differentiating interventions, outcomes and their measures into relevant categories – may be overcome with context-based language representations, and decomposing the classification problems across a pipeline approach. In the short term, this technology could be used to semi-automate construction of evidence tables, potentially as a first pass process that allows reviewers to start from a pre-filled baseline. Long term, as language representations evolve, and more innovative methods are developed to classify their outputs, it is conceivable that future systems could play an even greater role in automating the systematic review process, with medical domain experts needed only for oversight. This could potentially result in thousands of hours saved in labour costs across the healthcare industry, which could be redirected to achieve the ultimate goal of improving patient care.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] D. L. Sackett, W. M. C. Rosenberg, J. A. M. Gray, R. B. Haynes, W. S. Richardson, Evidence based medicine: what it is and what it isn't, *BMJ* 312 (7023) (1996) 71–72. doi:10.1136/bmj.312.7023.71.
- [2] D. L. Sackett, W. M. C. Rosenberg, On the need for evidence-based medicine, *Journal of Public Health* 17 (3) (1995) 330–334.
- [3] ClinicalTrials.gov, Trends, charts, and maps, <https://clinicaltrials.gov/ct2/resources/trends>, (Accessed on 09/17/2021) (2022).
- [4] J. P. Higgins, J. Thomas, J. Chandler, M. Cumpston, T. Li, M. J. Page, V. A. Welch, *Cochrane handbook for systematic reviews of interventions*, John Wiley & Sons, 2019.
- [5] NICE, National guideline centre. glaucoma: diagnosis and management, <https://www.nice.org.uk/guidance/ng81/evidence/full-guideline-pdf-4660991389>, (Accessed on 21/04/2023) (2017).
- [6] R. Borah, A. W. Brown, P. L. Capers, K. A. Kaiser, Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the prospero registry, *BMJ Open* 7 (2) (2017). doi:10.1136/bmjopen-2016-012545.
- [7] M. Michelson, K. Reuter, The significant cost of systematic reviews and meta-analyses: a call for greater involvement of machine learning to assess the promise of clinical trials, *Contemporary Clinical Trials Communications* 16 (2019) 100443.
- [8] A. Hunter, M. Williams, Aggregating evidence about the positive and negative effects of treatments, *Artificial Intelligence in Medicine* 56 (3) (2012) 173–190.
- [9] D. Demner-Fushman, J. Lin, Knowledge extraction for clinical question answering: Preliminary results, in: *Proceedings of the AAAI-05 Workshop on Question Answering in Restricted Domains*, AAAI Press (American Association for Artificial Intelligence) Pittsburgh, PA, 2005, pp. 9–13.

- [10] A. Trenta, A. Hunter, S. Riedel, Extraction of evidence tables from abstracts of randomized clinical trials using a maximum entropy classifier and global constraints (2015). [arXiv:1509.05209](https://arxiv.org/abs/1509.05209).
- [11] G. Y. Chung, Sentence retrieval for abstracts of randomized controlled trials, *BMC Medical Informatics and Decision Making* 9 (1) (2009) 1–13.
- [12] K. Hirohata, N. Okazaki, S. Ananiadou, M. Ishizuka, Identifying sections in scientific abstracts using conditional random fields, in: *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008, pp. 381–388.
- [13] K.-C. Huang, C. C.-H. Liu, S.-S. Yang, F. Xiao, J.-M. Wong, C.-C. Liao, I.-J. Chiang, Classification of PICO elements by text features systematically extracted from pubmed abstracts, in: *2011 IEEE International Conference on Granular Computing*, IEEE, 2011, pp. 279–283.
- [14] D. Jin, P. Szolovits, PICO element detection in medical text via long short-term memory neural networks, in: *Proceedings of the BioNLP 2018 workshop*, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 67–75. doi:10.18653/v1/W18-2308.
- [15] T. Zhang, Y. Yu, J. Mei, Z. Tang, X. Zhang, S. Li, Unlocking the power of deep PICO extraction: Step-wise medical NER identification, *ArXiv Preprint ArXiv:2005.06601* (2020).
- [16] F. W. Mutinda, K. Liew, S. Yada, S. Wakamiya, E. Aramaki, Automatic data extraction to support meta-analysis statistical analysis: a case study on breast cancer, *BMC Medical Informatics and Decision Making* 22 (1) (2022) 1–13.
- [17] S. Kiritchenko, B. De Bruijn, S. Carini, J. Martin, I. Sim, ExaCT: automatic extraction of clinical trial characteristics from journal publications, *BMC Medical Informatics and Decision Making* 10 (1) (2010) 1–17.
- [18] K. Hara, Y. Matsumoto, Extracting clinical trial design information from medline abstracts, *New Generation Computing* 25 (3) (2007) 263–275.

- [19] A. J. Brockmeier, M. Ju, P. Przybyła, S. Ananiadou, Improving reference prioritisation with PICO recognition, *BMC Medical Informatics and Decision Making* 19 (1) (2019) 1–14.
- [20] T. Kang, S. Zou, C. Weng, Pretraining to recognize PICO elements from randomized controlled trial literature, *Studies in Health Technology and Informatics* 264 (2019) 188.
- [21] B. E. Nye, A. Nenkova, I. J. Marshall, B. C. Wallace, TrialStreamer: mapping and browsing medical evidence in real-time, in: *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting, Vol. 2020, NIH Public Access, 2020*, p. 63.
- [22] S. R. Jonnalagadda, P. Goyal, M. D. Huffman, Automating data extraction in systematic reviews: a systematic review, *Systematic Reviews* 4 (1) (2015) 1–16.
- [23] S. Lim, J. Kang, Chemical–gene relation extraction using recursive neural network, *Database: The Journal of Biological Databases and Curation* 2018 (2018).
- [24] B. E. Nye, J. DeYoung, E. Lehman, A. Nenkova, I. J. Marshall, B. C. Wallace, Understanding clinical trial reports: Extracting medical entities and their relations, in: *AMIA Annual Symposium Proceedings, Vol. 2021, American Medical Informatics Association, 2021*, p. 485.
- [25] J. DeYoung, I. Beltagy, M. van Zuylen, B. Kuehl, L. L. Wang, MS2: Multi-document summarization of medical studies, *ArXiv Preprint ArXiv:2104.06486* (2021).
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in Neural Information Processing Systems, 2017*, pp. 5998–6008.
- [27] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *ArXiv Preprint ArXiv:1810.04805* (2018).
- [28] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).

- [29] B. Nye, J. J. Li, R. Patel, Y. Yang, I. Marshall, A. Nenkova, B. Wallace, A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 197–207. doi:10.18653/v1/P18-1019.
- [30] N. Stylianou, I. Vlahavas, Transformed: End-to-end transformers for evidence-based medicine and argument mining in medical literature, *Journal of Biomedical Informatics* 117 (2021) 103767.
- [31] T. Mayer, E. Cabrio, S. Villata, Transformer-based argument mining for healthcare applications, in: the 24th European Conference on Artificial Intelligence (ECAI) 2020, IOS Press, 2020, pp. 2108–2115.
- [32] L. Schmidt, J. Weeds, J. Higgins, Data mining in clinical trial text: Transformers for classification and question answering tasks, *ArXiv Preprint ArXiv:2001.11268* (2020).
- [33] R. Bhatnagar, S. Sardar, M. Beheshti, J. T. Podichetty, How can natural language processing help model informed drug development?: a review, *JAMIA open* 5 (2) (2022) ooac043.
- [34] D. Wright, NormCo: Deep disease normalization for biomedical knowledge base construction, University of California, San Diego, 2019.
- [35] D. Xu, Z. Zhang, S. Bethard, A generate-and-rank framework with semantic type regularization for biomedical concept normalization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8452–8464.
- [36] Z. Ji, Q. Wei, H. Xu, BERT-based ranking for biomedical entity normalization, *AMIA Summits on Translational Science Proceedings* 2020 (2020) 269.
- [37] R. Leaman, Z. Lu, TaggerOne: joint named entity recognition and normalization with semi-markov models, *Bioinformatics* 32 (18) (2016) 2839–2846.

- [38] Z. Miftahutdinov, A. Kadurin, R. Kudrin, E. Tutubalina, Medical concept normalization in clinical trials with drug and disease representation learning, *Bioinformatics* 37 (21) (2021) 3856–3864.
- [39] S. M. Meystre, P. M. Heider, A. Cates, G. Bastian, T. Pittman, S. Gentilin, T. J. Kelechi, Piloting an automated clinical trial eligibility surveillance and provider alert system based on artificial intelligence and standard data models, *BMC Medical Research Methodology* 23 (1) (2023) 1–11.
- [40] H. Hassanzadeh, S. Karimi, A. Nguyen, Matching patients to clinical trials using semantically enriched document representation, *Journal of Biomedical Informatics* 105 (2020) 103406.
- [41] L. Penberthy, R. Brown, F. Puma, B. Dahman, Automated matching software for clinical trials eligibility: measuring efficiency and flexibility, *Contemporary clinical trials* 31 (3) (2010) 207–217.
- [42] L. Joël, Y. Toussaint, C. Raïssi, A. Coulet, Cross-corpus training with treeLSTM for the extraction of biomedical relationships from text (2018).
URL <https://openreview.net/forum?id=S1LXVnxRb>
- [43] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, *ArXiv Preprint ArXiv:1907.11692* (2019).
- [44] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, *ArXiv Preprint ArXiv:1903.10676* (2019).
- [45] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, *Bioinformatics* 36 (4) (2020) 1234–1240.
- [46] D. Moher, S. Hopewell, K. F. Schulz, V. Montori, P. C. Gøtzsche, P. Devereaux, D. Elbourne, M. Egger, D. G. Altman, CONSORT 2010 explanation and elaboration: updated guidelines for reporting parallel group randomised trials, *International journal of surgery* 10 (1) (2012) 28–55.
- [47] Prodigy · an annotation tool for AI, machine learning & NLP, <https://prodi.gy/>, (Accessed on 09/19/2021) (2022).

- [48] A. O. Muis, W. Lu, Labeling gaps between words: Recognizing overlapping mentions with mention separators, ArXiv Preprint ArXiv:1810.09073 (2018).
- [49] A. Zapf, S. Castell, L. Morawietz, A. Karch, Measuring inter-rater reliability for nominal data— which coefficients and confidence intervals are appropriate?, BMC medical research methodology 16 (1) (2016) 1–10.
- [50] A. De Raadt, M. J. Warrens, R. J. Bosker, H. A. Kiers, Kappa coefficients for missing data, Educational and psychological measurement 79 (3) (2019) 558–576.
- [51] M. Neumann, D. King, I. Beltagy, W. Ammar, ScispaCy: Fast and robust models for biomedical natural language processing, ArXiv Preprint ArXiv:1902.07669 (2019).
- [52] N. Le Guillarme, W. Thuiller, TaxoNERD: deep neural models for the recognition of taxonomic entities in the ecological and evolutionary literature, Methods in Ecology and Evolution (2021). doi:10.1111/2041-210X.13778.
- [53] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, CoRR abs/1603.01360 (2016). arXiv:1603.01360.
- [54] L. Ratinov, D. Roth, Design challenges and misconceptions in named entity recognition, in: Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009), 2009, pp. 147–155.
- [55] Library architecture · spaCy API documentation, <https://spacy.io/api>, (Accessed on 09/19/2021) (2022).
- [56] M. Abadeer, Assessment of DistilBERT performance on named entity recognition task for the detection of protected health information and medical concepts, in: Proceedings of the 3rd Clinical Natural Language Processing Workshop, Association for Computational Linguistics, Online, 2020, pp. 158–167. doi:10.18653/v1/2020.clinicalnlp-1.18.
- [57] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

- [58] Explosion, spacy project: Example project of creating a novel nlp component to do relation extraction from scratch, https://github.com/explosion/projects/tree/v3/tutorials/rel_component, (Accessed on 09/19/2021) (2021).
- [59] I. Segura-Bedmar, P. Martínez, M. Herrero-Zazo, SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013), in: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 341–350.
- [60] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, *IEEE Transactions on Knowledge & Data Engineering* 34 (01) (2020) 50–70. doi:10.1109/TKDE.2020.2981314.

Figure 2: An example of how an evidence table for an RCT investigation of a glaucoma treatment might look in NICE clinical guidelines [\cite{nice_2017}}](#). CI: confidence interval.

Outcome	Intervention	Control	Comparative	Quality
Mean change in intraocular pressure (IOP)	-6.3 mmHg	-0.2 mmHg	Not reported	High
Visual field progression at 24 months	23 (12.6%) patients	57 (27.4%) patients	Hazard ratio: 0.49 95% CI:0.21–0.67 p=0.037	Medium
Serious adverse events	12 events	7 events	Not reported	Low

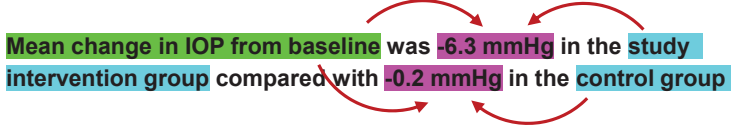
Figure 2: A simple example of how an input result sentence (1) can be processed through NER (2) and RE (3) and then tabulated (4) to form a segment of the example evidence table from Figure 1.

1. INPUT RESULT SENTENCE

Mean change in IOP from baseline was -6.3 mmHg in the study intervention group compared with -0.2 mmHg in the control group

3. RELATION EXTRACTION

Mean change in IOP from baseline was -6.3 mmHg in the study intervention group compared with -0.2 mmHg in the control group



The diagram illustrates relation extraction from the input sentence. Red arrows indicate relationships between entities: 'Mean change in IOP from baseline' is related to '-6.3 mmHg', which is related to 'study intervention group'. Similarly, '-0.2 mmHg' is related to 'control group'. There are also arrows between 'study intervention group' and 'control group', and between the two numerical values.

2. NAMED ENTITY RECOGNITION

Mean change in IOP from baseline was -6.3 mmHg in the study intervention group compared with -0.2 mmHg in the control group

4. TABULATION

Outcome	Intervention	Control
Mean change in IOP	-6.3 mmHg	-0.2 mmHg

Figure 9: The architecture of our full study system.

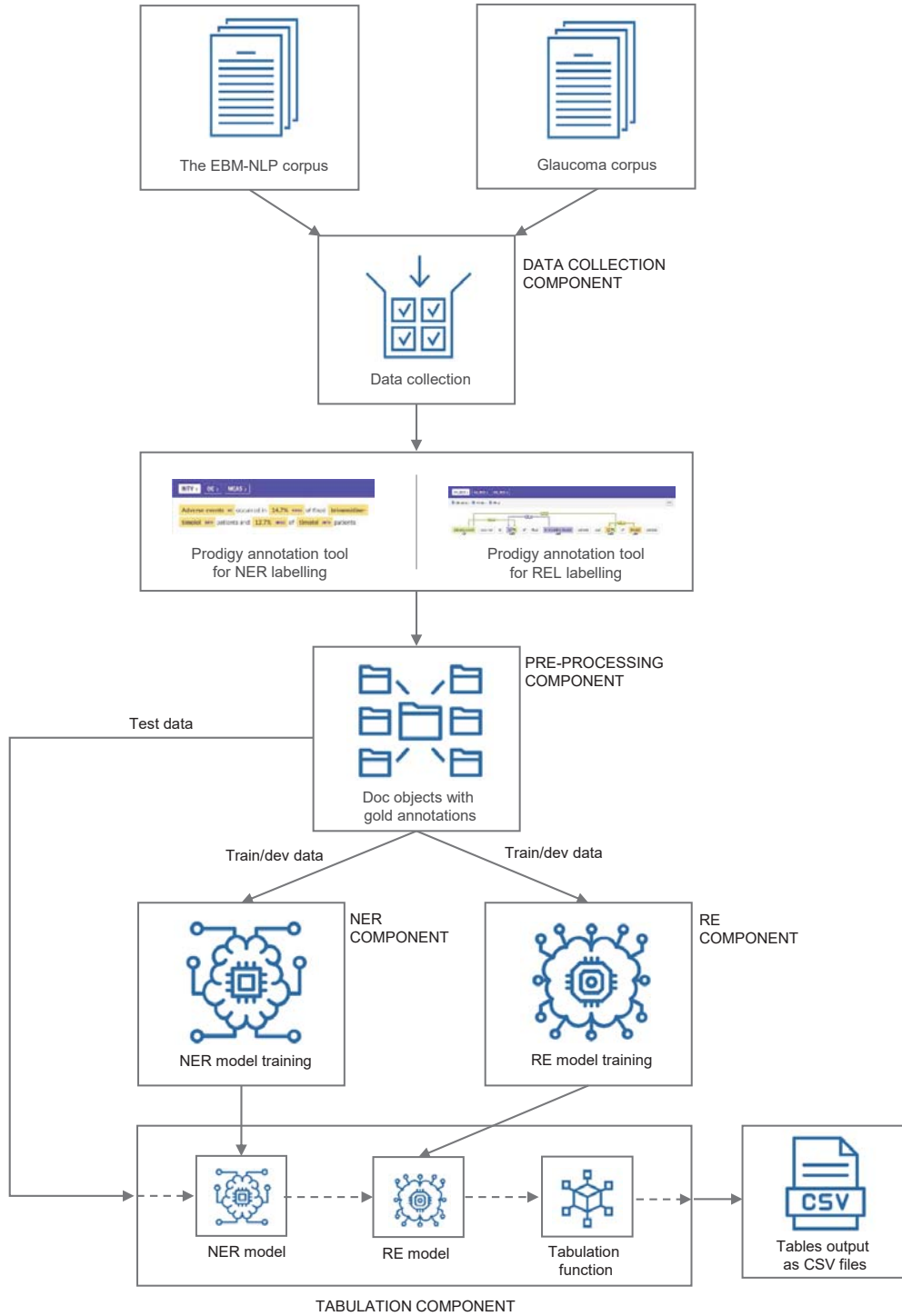


Figure 10: Example of optimal actions taken as an input sequence is passed into the dependency parser.

Actions	Output	Stack	Buffer	Entity
	[]	[]	[interocular, pressure, in, the, latanoprost, arm]	
begin	[]	[interocular]	[pressure, in, the, latanoprost, arm]	
last	[(interocular pressure)-OC]	[]	[in, the, latanoprost, arm]	
out	[(interocular pressure)-OC, in]	[]	[the, latanoprost, arm]	(interocular pressure)-OC
out	[(interocular pressure)-OC, in, the]	[]	[latanoprost, arm]	
unit	[(interocular pressure)-OC, in, the, (latanoprost)-INTV]	[]	[arm]	(latanoprost)-INTV
out	[(interocular pressure)-OC, in, the, (latanoprost)-INTV, arm]	[]	[]	

Figure 11: Schematic of the NER component architecture.

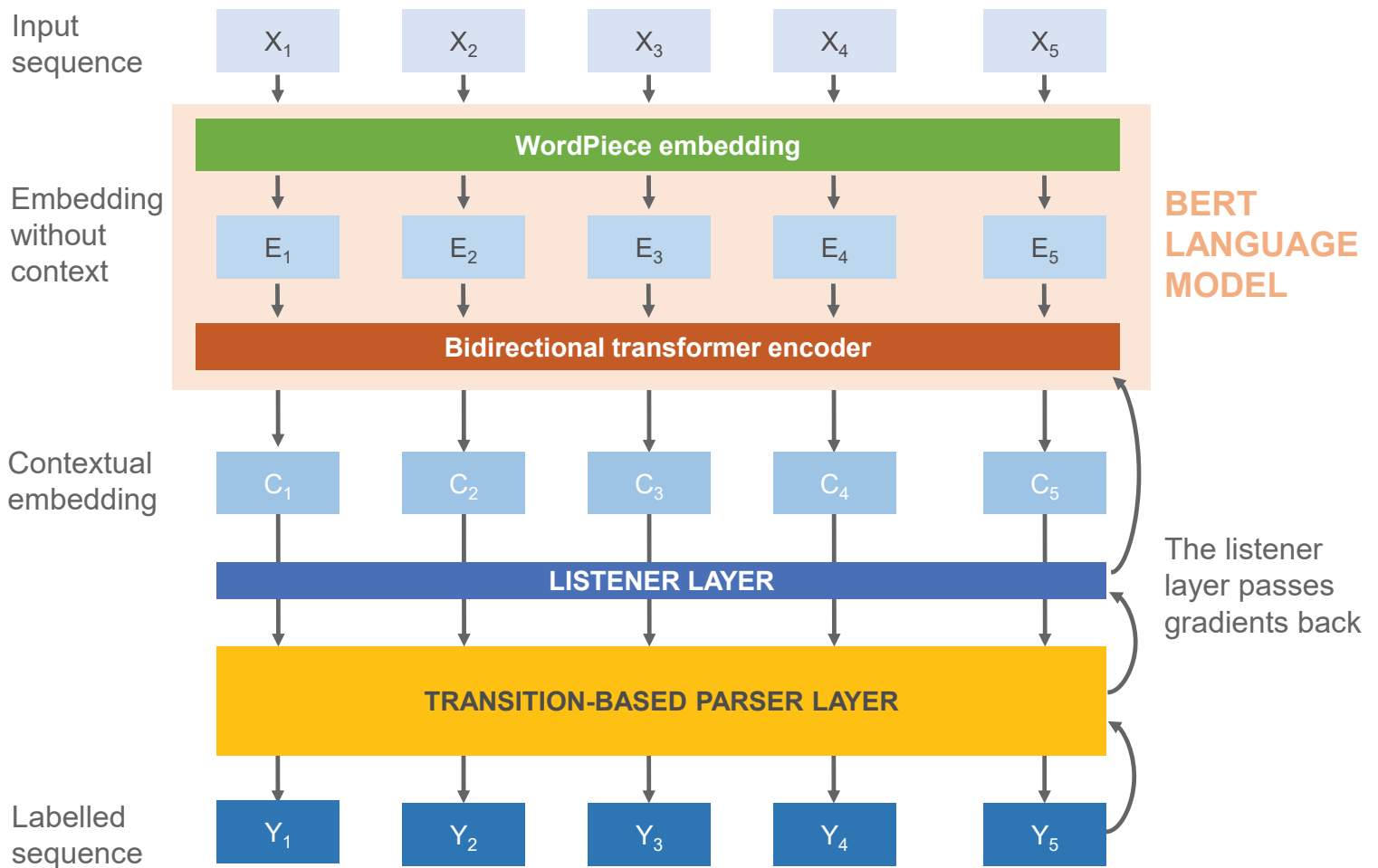


Figure 12: Schematic of the RE component architecture.

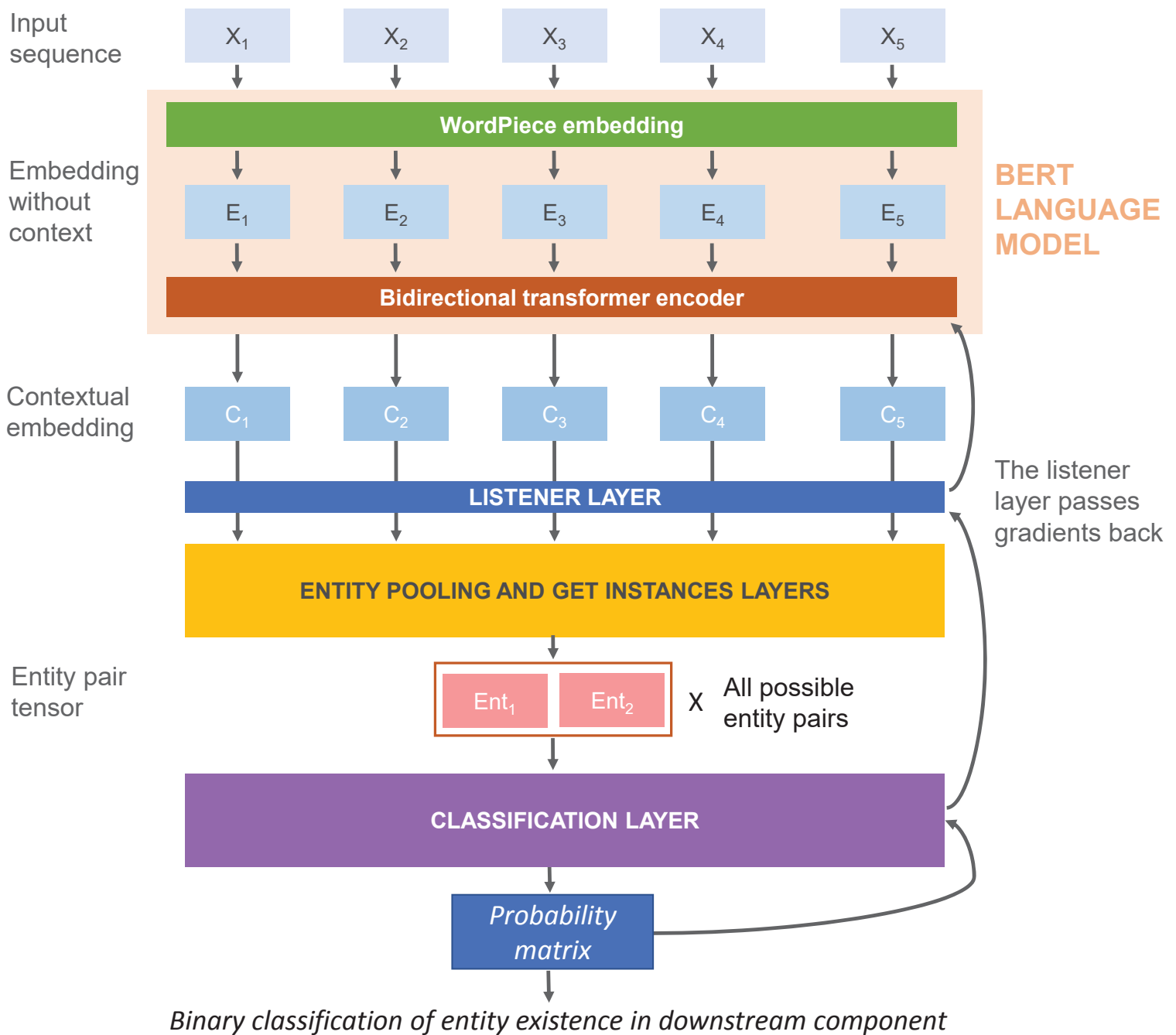


Figure 13: System (BioBERT language model) performance results across the five IE tasks after training on varying proportions of the all-domains training set. We report mean values from 10 runs and SD as error bars.

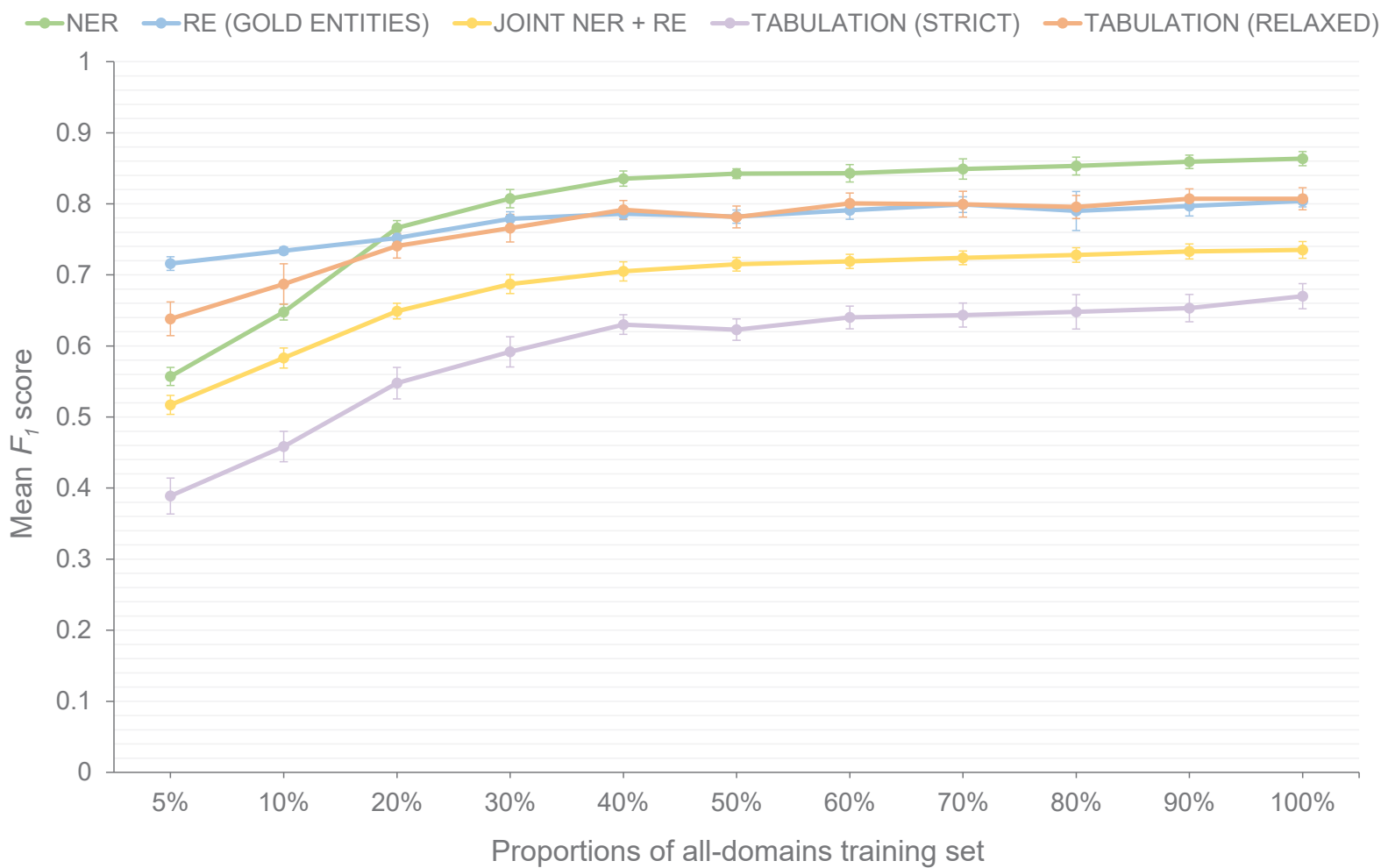


Figure 14: System (BioBERT language model) performance results of the five IE tasks on different unseen disease-area domain test sets. We report mean values from 10 runs and SD as error bars.

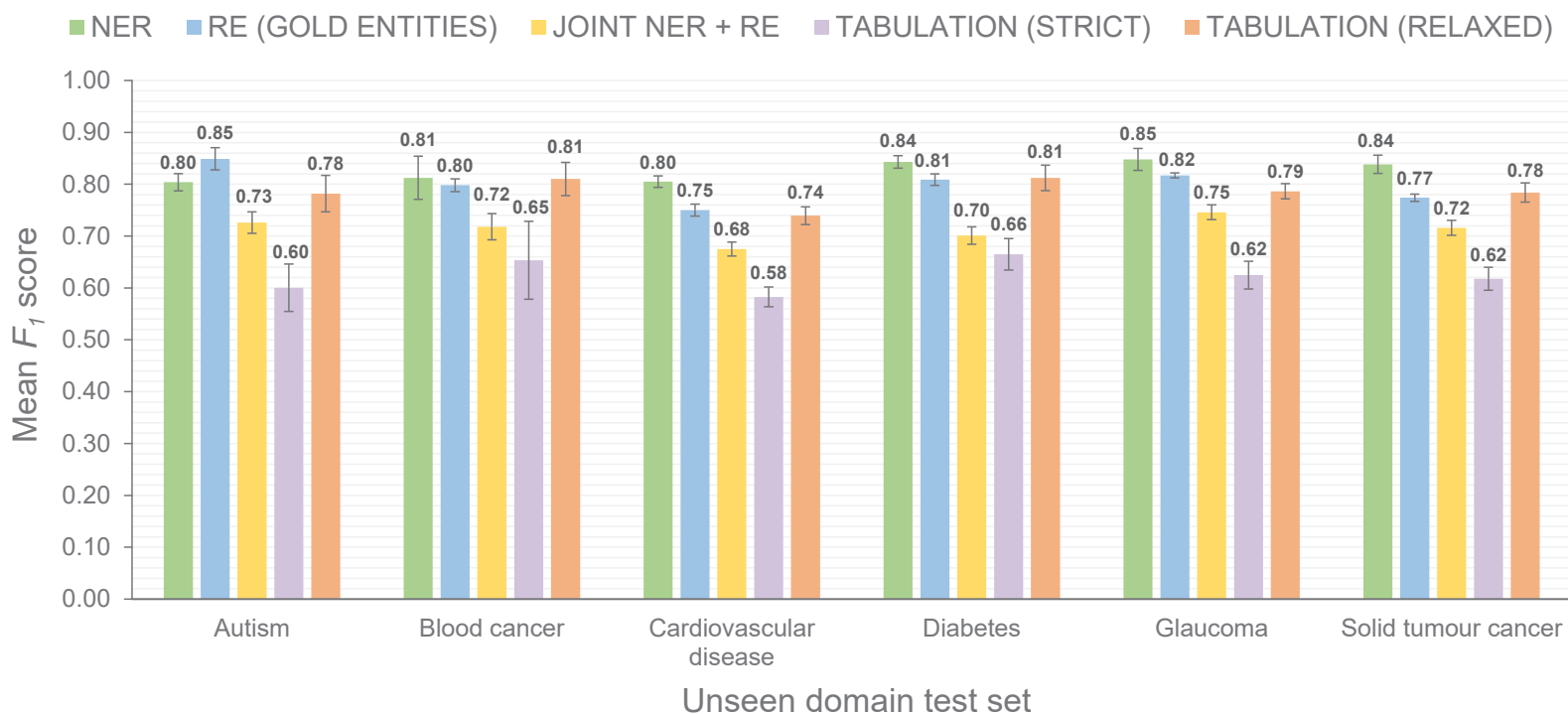


Figure 15: System (BioBERT language model) performance on the autism test set as training domain variety increases. We report mean values from 10 runs and SD as error bars.

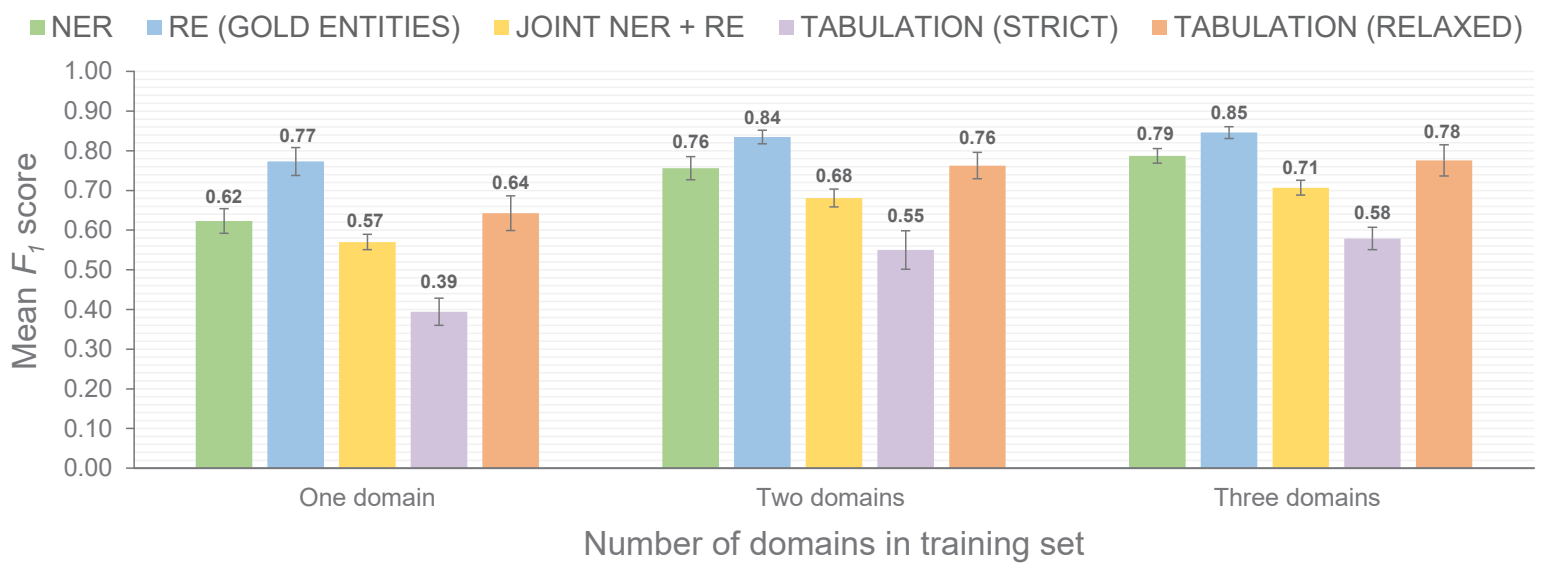


Figure 16: Comparison of system (BioBERT language model) performance after training on different disease areas with the same number of examples, tested on an unseen test set from same domain. We report mean values from 10 runs and SD as error bars.

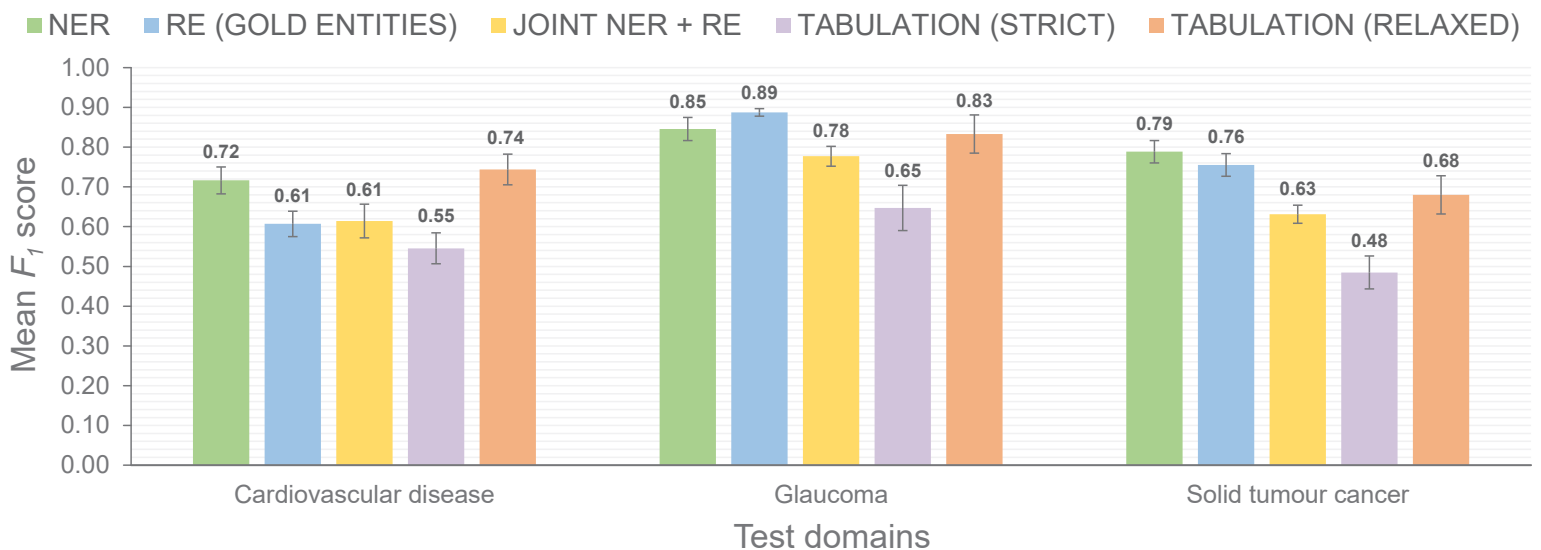


Figure 19: Incomplete OC label error. Gold-standard OC labels are in green, while the predicated label is in red.

OC

Overall , 58 % of the 36 patients assigned to AMB successfully completed prophylaxis compared with 80 % of the 41 patients assigned to FLU (< 0.05)

Figure 20: MEAS misclassification error. Gold-standard INTV labels are in green, while the predicated label is in red.

INTV **MEAS**
Latanoprost **0.005%** once daily reduced IOP (+/- SEM) more effectively than **latanoprost** **0.0015%**
twice daily (9.8 +/- 0.9 mm Hg and 6.7 +/- 0.9 mm Hg, respectively)

Figure 21: Negative relation classification error. The model only successfully recognises one relation (pink), and misses five (green).

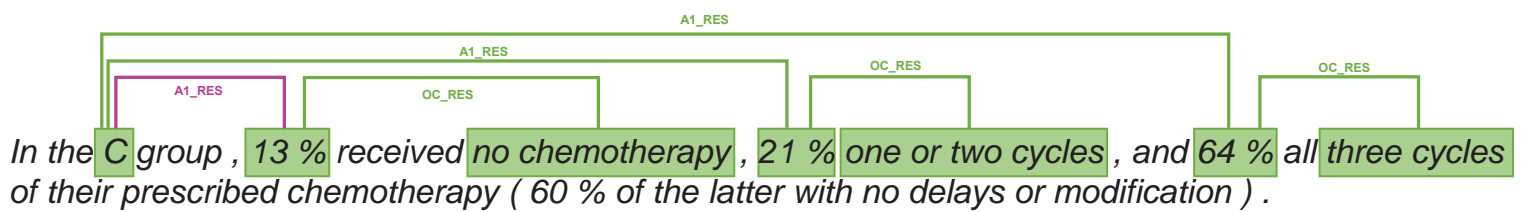


Table 1: Total number and proportions of entity and relation annotations in the gold corpus.

	Entity labels			Relation labels		
	INTV	MEAS	OC	A1_RES	A2_RES	OC_RES
Count	990	1096	1455	717	630	1835
Proportion	0.28	0.31	0.41	0.23	0.20	0.58

Table 2: System performance results of the five IE tasks for each BERT-based language representation trained on the all-domains test set. We report mean and SD values from 10 runs.

EMBEDDING MODEL	NER			RE (GOLD ENTITIES)			JOINT NER + RE			TABULATION (STRICT)			TABULATION (RELAXED)		
	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>
BioBERT	0.86	0.87	0.86	0.78	0.83	0.80	0.70	0.78	0.73	0.56	0.82	0.67	0.76	0.86	0.81
	±0.01	±0.01	±0.01	±0.04	±0.04	±0.01	±0.03	±0.03	±0.01	±0.02	±0.02	±0.02	±0.02	±0.02	±0.02
SciBERT	0.85	0.86	0.85	0.78	0.80	0.79	0.69	0.75	0.72	0.53	0.79	0.63	0.72	0.84	0.78
	±0.01	±0.01	±0.01	±0.02	±0.03	±0.01	±0.02	±0.02	±0.01	±0.02	±0.03	±0.02	±0.03	±0.02	±0.02
RoBERTa	0.86	0.85	0.86	0.77	0.80	0.79	0.70	0.76	0.73	0.52	0.80	0.63	0.72	0.85	0.78
	±0.01	±0.01	±0.01	±0.03	±0.04	±0.01	±0.04	±0.03	±0.01	±0.03	±0.02	±0.02	±0.04	±0.02	±0.02

Values in bold indicate the language model that scored the highest on a task-specific metric.

Table 3: Performance of the NER component for individual entity labels on the all-domains test set. We report mean and SD values from 10 runs.

EMBEDDING MODEL	OC			INTV			MEAS		
	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>
BioBERT	0.76 ± 0.02	0.75 ± 0.02	0.76 ± 0.02	0.93 ± 0.01	0.92 ± 0.01	0.93 ± 0.01	0.88 ± 0.02	0.90 ± 0.02	0.89 ± 0.02
SciBERT	0.72 ± 0.02	0.73 ± 0.03	0.72 ± 0.02	0.92 ± 0.02	0.91 ± 0.02	0.92 ± 0.01	0.88 ± 0.02	0.90 ± 0.01	0.89 ± 0.01
RoBERTa	0.75 ± 0.03	0.73 ± 0.03	0.74 ± 0.02	0.94 ± 0.01	0.91 ± 0.02	0.92 ± 0.01	0.87 ± 0.02	0.89 ± 0.02	0.88 ± 0.02

Values in bold indicate the language model that scored the highest on a task-specific metric.

Table 4: Performance of the RE component for individual relation labels on the all-domains test set. We report mean and SD values from 10 runs.

EMBEDDING MODEL	A1_RES			A2_RES			OC_RES		
	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>
BioBERT	0.87 ±0.02	0.92 ±0.03	0.90 ±0.02	0.86 ±0.04	0.96 ±0.02	0.90 ±0.02	0.72 ±0.06	0.75 ±0.07	0.73 ±0.01
SciBERT	0.86 ±0.04	0.87 ±0.03	0.86 ±0.02	0.84 ±0.05	0.91 ±0.03	0.87 ±0.02	0.73 ±0.03	0.72 ±0.05	0.72 ±0.02
RoBERTa	0.88 ±0.04	0.86 ±0.03	0.87 ±0.01	0.86 ±0.03	0.91 ±0.03	0.88 ±0.01	0.70 ±0.05	0.75 ±0.06	0.72 ±0.01

Values in bold indicate the language model that scored the highest on a task-specific metric.

Conflicts of interest

We have no conflicts of interest to disclose.



Click here to access/download
Supplementary Material
supplementary_appendix_v1.pdf

