Journal of
Biomedical Semantics

RESEARCH Open Access

Comparing generative and extractive approaches to information extraction from abstracts describing randomized clinical trials

Christian Witte[1†], David M. Schmidt[1*†] and Philipp Cimiano[1]

Abstract

Background Systematic reviews of Randomized Controlled Trials (RCTs) are an important part of the evidence-based medicine paradigm. However, the creation of such systematic reviews by clinical experts is costly as well as time-consuming, and results can get quickly outdated after publication. Most RCTs are structured based on the Patient, Intervention, Comparison, Outcomes (PICO) framework and there exist many approaches which aim to extract PICO elements automatically. The automatic extraction of PICO information from RCTs has the potential to significantly speed up the creation process of systematic reviews and this way also benefit the field of evidence-based medicine.

Results Previous work has addressed the extraction of PICO elements as the task of identifying relevant text spans or sentences, but without populating a structured representation of a trial. In contrast, in this work, we treat PICO elements as structured templates with slots to do justice to the complex nature of the information they represent. We present two different approaches to extract this structured information from the abstracts of RCTs. The first approach is an extractive approach based on our previous work that is extended to capture full document representations as well as by a clustering step to infer the number of instances of each template type. The second approach is a generative approach based on a seq2seq model that encodes the abstract describing the RCT and uses a decoder to infer a structured representation of a trial including its arms, treatments, endpoints and outcomes. Both approaches are evaluated with different base models on a manually annotated dataset consisting of RCT abstracts on an existing dataset comprising 211 annotated clinical trial abstracts for Type 2 Diabetes and Glaucoma. For both diseases, the extractive approach (with `flan-t5-base`) reached the best F_1 score, i.e. 0.547 (±0.006) for type 2 diabetes and 0.636 (±0.006) for glaucoma. Generally, the F_1 scores were higher for glaucoma than for type 2 diabetes and the standard deviation was higher for the generative approach.

Conclusion In our experiments, both approaches show promising performance extracting structured PICO information from RCTs, especially considering that most related work focuses on the far easier task of predicting less structured objects. In our experimental results, the extractive approach performs best in both cases, although the lead is greater for glaucoma than for type 2 diabetes. For future work, it remains to be investigated how the base model size affects the performance of both approaches in comparison. Although the extractive approach currently leaves more room for direct improvements, the generative approach might benefit from larger models.

[†] Christian Witte and David M. Schmidt contributed equally to this work.

*Correspondence:
David M. Schmidt
david.schmidt@uni-bielefeld.de
Full list of author information is available at the end of the article

Keywords Information Extraction, Deep Learning, Clinical Trials, Evidence-Based Medicine, PICO

Introduction

The number of publications describing Randomized Controlled Trials has been increasing at an exponential pace for decades [1], thus making it more and more challenging to appropriately summarize the existing clinical evidence by way of systematic reviews. Yet, the ability to summarize the current clinical evidence is a core process to support evidence-based medical decision making [2]. Indeed, the creation of systematic reviews is costly and time consuming as it is done manually by clinical experts with the result that systematic reviews and guidelines quickly become outdated after publication or are even outdated at the time of publication [3–6]. Due to the effort associated with the creation of systematic reviews, there has been significant interest on the question how to automate their creation [7–9]. Recently, approaches to automatically summarize clinical evidence by way of argumentative structures have been proposed [10]. The bottleneck for such approaches is the missing availability of a database of semantically described clinical trials that comprise of structured representations of the key outcomes of each study. As argued by Sánchez-Graillet et al. [10], information extraction approaches have the potential to support the extraction of key information about the design and results of clinical trials from publications. These structured representations of the results of a trial in turn could support the process of systematic review creation or at least considerably reduce the effort to do so.

Most RCTs follow the PICO (**P**atient, **I**ntervention, **C**omparison, **O**utcomes) framework for structuring the presentation of clinical research findings. As a result, early IE approaches in the clinical domain classify full sentences of RCTs [11, 12] or smaller text spans [13] into the elements of the PICO framework. However, treating the PICO elements as flat objects represented as a collection of text spans does not reflect the complex information presented in RCTs for the following reasons: 1) the description of a single PICO element could be spread across several sentences and 2) the relationship between different PICO elements is not modelled (e.g. which outcomes belong to the intervention group and which ones belong to the comparison group).

Witte and Cimiano [14] have proposed an extractive information extraction approach that captures the design and key results of an RCT by way of 10 different templates that capture the PICO elements in a structured way, modelling dependencies and relations between them. These templates are based on the C-TrO Ontology that has been designed to support use cases related to the aggregation of evidence from multiple clinical trials [15]. Those templates are instantiated with information from a given abstract describing the trial. For instance, a template `Medication` with slots `DrugName`, `DoseValue` and `DoseUnit` could be used to describe medications of intervention arms mentioned in a RCT. However, Witte and Cimiano [14] assume that the number of template instances (e.g. number of outcomes) is provided a-priori, which hinders the application of their approach in real world settings. Further, the approach of Witte and Cimiano [14] chunks the text into smaller segments and then combines the templates instantiated for each segment. This makes it difficult to capture relations that are mentioned across chunks.

In this paper, we build on the approach of Witte and Cimiano [14] and extend it in two directions. First, we rely on Longformers [16] and Flan-T5 [17] in order to encode the complete abstract, inferring template instances and slots jointly for the complete text. Second, overcoming the key assumption that the number of template instances are known a priori, we extend the approach by a clustering step that induces the number of template instances in an unsupervised manner.

Beyond the extractive approach, we also present a generative approach that is inspired in recent seq2seq architectures such as REBEL [18] or GenIE [19]. These approaches rely on an encoder-decoder architecture by which the text is encoded and certain output structures are generated. We apply this idea to directly decode a complex nested template structure representing the design and key results of a study. As main novelties, we propose a decoding approach that relies on a grammar to guide decoding, ensuring that only valid structures are generated. Second, we present an approach to linearize the structure to be predicted such that it can be encoded as a sequence to be predicted by the generative approach. Our grammar-constrained decoding approach is inspired by Lu et al. [20], who also prune/mask the vocabulary to consist only of elements which comply with the desired output format. The decoding mechanism presented in this work generalizes the output format specification to arbitrary right-linear context-free grammars.

We evaluate and compare both approaches on the dataset provided by Sanchez-Graillet et al. [21] and used in previous work [14], which consists of predicting 10 templates. The dataset comprises a total of 211 documents for two diseases: type 2 diabetes (104) and glaucoma (107). Our results show that the improved extractive approach using Flan-T5 as a base model performs best for both diseases in the dataset, achieving a mean F_1 score of 0.547 (±0.006) for type 2 diabetes and 0.636 (±0.006) for glaucoma. However, both approaches have different strengths and weaknesses and are not yet suitable to fully automate the process of systematic review creation, but still have the potential to reduce the necessary effort a lot.

Additional data and evaluations (Appendix 2, 4 and 5) as well as the used grammar (Appendix 1) and a case study (Appendix 3) can be found in the appendix.

In summary, our contributions are the following:

- We present an extension of the approach proposed by Witte and Cimiano [14] in two directions: i) relying on Longformers [16] and Flan-T5 [17] to encode the complete abstract and infer templates and slots for the complete document jointly, and ii) using a clustering step to cluster the extracted template instances to infer the number of instances for each template type.
- We present a novel generative information extraction approach that relies on a grammar to guide decoding, and propose a novel serialization of the nested template structure such that the problem can be casted as a seq2seq inference problem.
- We evaluate both approaches on the dataset by Sanchez-Graillet et al. [21] and show that our improved extractive approach using Flan-T5 [17] as a base model performs best for both diseases.

Related work

In recent years, a number of information extraction approaches have been developed, targeting tasks such as event extraction (e.g., Lu et al. [22], Hsu et al. [23], Yang et al. [20]), relation extraction (e.g., Giorgi et al. [24]) or role/slot/template filling (e.g. Du et al. [25, 26]). With respect to biomedical information extraction, there are also several approaches which aim to solve different tasks specifically for the domain of biomedical texts, e.g. scientific articles or clinical trials. Application domains range from event extraction (e.g., Wang et al. [27], Ramponi et al. [28], Zhu and Zheng [29], Huang et al. [30], Trieu et al. [31]) over relation extraction (e.g., Jiang and Kavuluru [32, 33]) and question answering (e.g., Wang et al. [27]) through to named entity recognition (e.g., Stylianou et al. [34]).

The set of methods and tools used to solve these problems is quite diverse, comprising joint end-to-end transformer models (e.g., Ramponi et al. [28], Trieu et al. [31], Jiang and Kavuluru [32], Stylianou et al. [34]) as well as support vector machines (e.g., Kim and Meystre [33]), conditional random fields (e.g., Stylianou et al. [35], Farnsworth et al. [34], Tseo et al. [36]), hybrid deep neural networks (e.g., Zhu and Zheng [29]) and Long Short-Term Memory networks (LSTMs, e.g., Jiang and Kavuluru [32], Kim and Meystre [33], Farnsworth et al. [35]).

Some related work also deals with detecting clinical trial outcomes, outcome spans (e.g., Abaho et al. [37–39], Ganguly et al. [40]) or slot fillers (e.g., Papanikolaou et al. [41]) in (randomized) clinical trial abstracts. However, they lack the specific structure and dependencies of PICO templates and slots, which are used in this paper. These approaches mostly use transformer architectures, sometimes in combination with, e.g., LSTMs to detect the outcomes/slot fillers.

The PICO framework is frequently used to describe the results of RCTs in a structured way. This structure comprises of a number of templates and corresponding slots (which are uniquely assigned to a single template type). However, a RCT can contain multiple instances of a template, imposing the problem of matching recognized slot fillers with their corresponding template instance.

Some efforts in this area focus on the problem that larger amounts of training data are missing or at least expensive to create due to the need for clinical experts as annotators. These approaches therefore utilize distant or weak supervision for training on noisy label data (e.g., Dhrangadhariya and Müller [42], Nye et al. [43], Wallace et al. [44], Liu et al. [45]). In contrast, the approach presented in this paper relies on the availability of sufficient classical supervised training data.

Other methods work with Conditional Random Fields (CRFs) in combination with (Bi-)LSTMs (e.g., Jin and Szolovits [46], Kang et al. [47]) or rule-based methods (e.g., Chabou and Iglewski [48]).

While most recent work relies on transformer architectures, there are also diverse other approaches which utilize different machine learning techniques like support vector machines (e.g., Yuan et al. [49]), convolutional neural networks (e.g., Stylianou et al. [50]), LSTMs (e.g., Jin and Szolovits [51]) or other deep learning-based approaches (e.g., Afzal et al. [52]).

Several recent approaches use transformer models like BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. [53]) for PICO recognition, but focus on different architectual and task-related details.

However, some approaches refer to PICO elements as flat classes, i.e. parts of sentences are just labeled, e.g., P or I, whereas our approach considers PICO elements to be nested structures, i.e. templates with slots that have to be filled with some portion of text. Examples for this simplified view on PICO elements are listed in the following:

Schmidt et al. [54] treat the PICO recognition task as a sentence classification/question answering task and thus, in contrast to the approach presented in this paper, do not work on the level of whole documents/abstracts or PICO elements which span multiple sentences. Therefore, Schmidt et al. [54] do not benefit from contextualized representations utilizing the whole abstract as a context. Moreover, the problem of mapping found PICO elements to unique template instances is not dealt with.

Zhang et al. [55] propose a multi-step approach that first identifies P, I/C and O elements in the text using either Convolutional Neural Networks (CNNs) or Bi-LSTMs. After that, a Diseases Named Entity Recognition model is used to extract disease-related entities in the PICO-labeled sentences. Various different models, like, e.g., BERT-based or LSTM-based models, are compared in this category. Finally, a mapping model resolves some ambiguities, like intersections of recognition results for P and O. Again, different models (including both BERT and Bi-LSTMs) are evaluated for this task. Although this approach makes some efforts to create more structured results than flat sentence classification, it still ignores some aspects of the more complex structure of PICO elements.

Whitton and Hunter [56] propose a more structured view on PICO elements, e.g., by differentiating between two arms of a RCT. This is achieved in two steps by first applying a named-entity recognition model, recognizing three general types of entities (interventions, outcomes and measures). In a second step, they are then related to each other using a relation extraction model which also differentiates between the (up to) two arms of the considered RCTs. However, they focus on evidence tables, which are different from the nested template structure we work with in this paper. Moreover, the other approach does not work in a sequence-to-sequence manner with constrained decoding like the generative approach described in this paper.

Dhrangadhariya et al. [57] implement PICO recognition for more fine-grained entities, which - similarly to our approach - also consider more detailed information about participants, interventions and outcomes, like sample size, age, mortality, drugs or surgical interventions. Nevertheless, it is still less detailed than the template structure used in this paper, which consists of 10 templates comprising overall 85 slots (see Witte and Cimiano [14]). Moreover, by using BERT as an encoder and Bi-LSTM, self-attention as well as CRF and linear layers for classification, it does not work in a sequence-to-sequence manner like the generative approach we present in this work.

Methods

In this work, we address the problem of extracting a set of template instances from unstructured text. We tackle this problem from two different perspectives and present two approaches solving the same problem: 1) an extractive approach and 2) a generative approach. An illustration of both approaches can be found in Fig. 1.

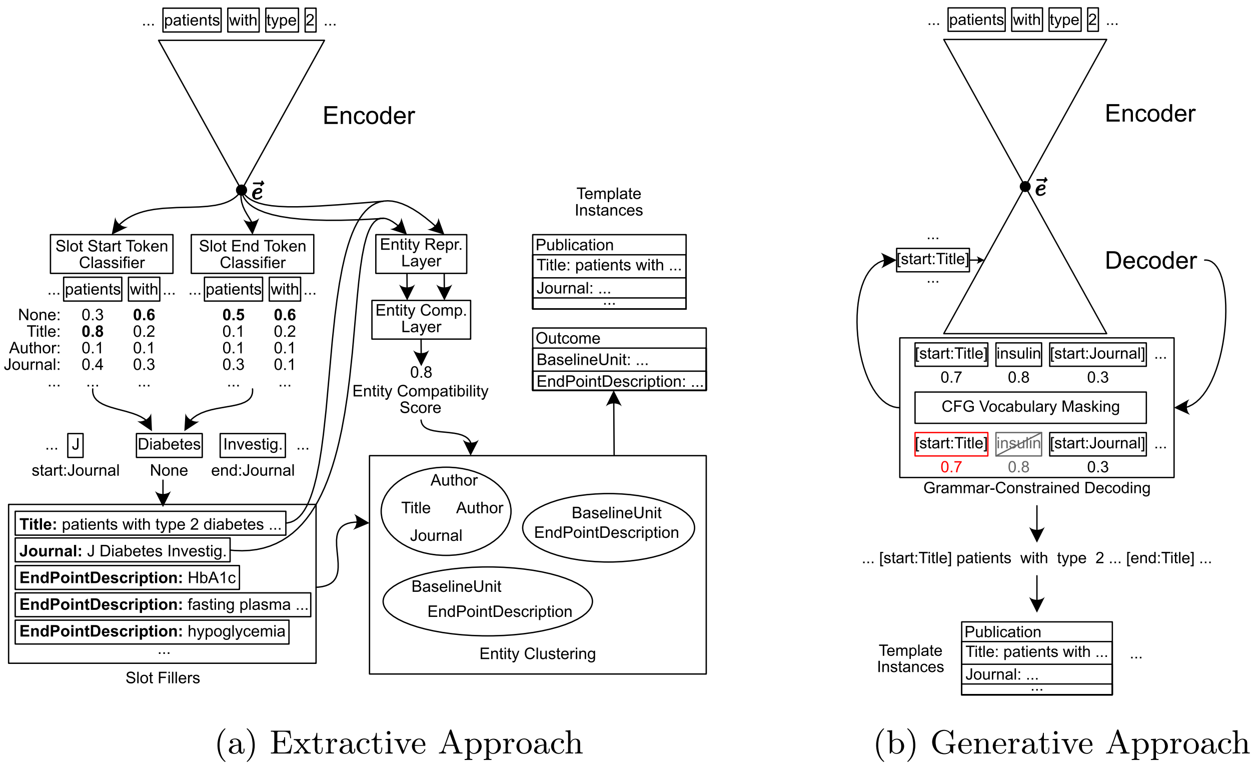
The used data model captures the design and key results of an RCT by way of 10 different templates consisting of a total of 85 different slots that capture various aspects of the PICO elements in a structured way. These templates are based on the C-TrO Ontology that has been designed to support use cases related to the aggregation of evidence from multiple clinical trials [15]. The mean number of slot fillers per template is shown in Table 1. A template t_i is defined by a type $i \in \mathcal{L}$ and a set of slots $\mathcal{S}_i = \bigcup_j s_{ij}$, where s_{ij} denotes slot j of template t_i, $\bigcup_j$ this way denotes the set union over all slots j and $\mathcal{L}$ denotes the set of all template types. A template is instantiated by assigning slot-fillers to its slots, where a slot-filler can be either a text span from the input document or a template instance, depending on the slot. Figure 2 visualizes the used data model. In the following subsections, we describe the extractive and the generative approach in more detail.

Extractive approach

Our extractive approach is based on the Intra-Template Compatibility (ITC) approach [14], which adopts a two-step architecture: In a first step, all textual slot-fillers are extracted from the input document, followed by a second step, which assigns the extracted slot-fillers to template instances. The extraction of slot-fillers and their clustering and assignment are described in the "Extraction of textual slot-fillers" and "Assignment of textual slot-fillers to template instances" Sections, respectively.

Encoding of the input document

The ITC approach uses BERT (Bidirectional Encoder Representations from Transformers) [53] to compute a contextualized representation of each token w_i of the input document $d = (w_1, \ldots, w_n)$. As the length of RCT abstracts typically exceeds the maximum number of tokens of most BERT implementations, the authors of ITC split the document into consecutive chunks and process each chunk separately. However, this approach treats each chunk as an isolated unit and hence the model is not able to learn token representations which incorporate the context of the full input document. Therefore, we adopt the Longformer [16] approach as well as the Flan-T5 model [17] to learn full-document contextualized representations $\mathbf{h}_i \in \mathbb{R}^d$ (with $d = 768$ for both T5 and


... patients with type 2 ...
Encoder
$\vec{e}$
Slot Start Token Classifier
Slot End Token Classifier
Entity Repr. Layer
Entity Comp. Layer
Template Instances
... patients with ...
None: 0.3 0.6
Title: 0.8 0.2
Author: 0.1 0.1
Journal: 0.4 0.3
0.5 0.6
0.1 0.2
0.1 0.1
0.3 0.1
0.8
Entity Compatibility Score
Publication
Title: patients with ...
Journal: ...
Outcome
BaselineUnit: ...
EndPointDescription: ...
... J Diabetes Investig. ...
start:Journal None end:Journal
Title: patients with type 2 diabetes ...
Journal: J Diabetes Investig.
EndPointDescription: HbA1c
EndPointDescription: fasting plasma ...
EndPointDescription: hypoglycemia
Slot Fillers
Author
Title Author
Journal
BaselineUnit EndPointDescription
BaselineUnit EndPointDescription
Entity Clustering
Decoder
[start:Title]
[start:Title] insulin [start:Journal] ...
0.7 0.8 0.3
CFG Vocabulary Masking
[start:Title] insulin [start:Journal] ...
0.7 0.8 0.3
Grammar-Constrained Decoding
... [start:Title] patients with type 2 ... [end:Title] ...
Template Instances
Publication
Title: patients with ...
Journal: ...

(a) Extractive Approach (b) Generative Approach

Fig. 1 Illustration of both described approaches starting with the tokenized input and ending with the generated template instances

Table 1 Mean and standard deviation of the number of slot fillers per template in the used dataset, separated by type of disease. Numbers rounded to two decimal places

Template	Type 2 diabetes	Glaucoma
Arm	7.01 (±2.79)	4.8 (±2.07)
ClinicalTrial	14.63 (±3.07)	15.1 (±3.12)
DiffBetweenGroups	3.61 (±0.81)	3.32 (±0.72)
Endpoint	1.68 (±0.85)	1.81 (±0.95)
EvidenceQuality	1.00 (±0.00)	4.00 (±0.00)
Intervention	1.91 (±0.79)	2.24 (±0.74)
Medication	1.98 (±1.11)	2.13 (±1.17)
Outcome	2.53 (±1.14)	3.45 (±1.62)
Population	3.13 (±1.86)	2.32 (±1.09)
Publication	12.45 (±3.44)	10.14 (±3.79)

Longformer models) for each token w_i of the input document, where d is the output dimension of the encoder of the respective model.

Extraction of textual slot-fillers

The ITC approach extracts slot-fillers from the input document by predicting start and end tokens of slot-fillers, followed by a step which joins the predicted start and end tokens. This is realized by training two linear layers which take the contextualized representation $\mathbf{h}_i$ of the tokens w_i as input and predicts whether or not this is a slot-filler start or end token, respectively:

$$\mathbf{p}_{s,i} = \operatorname{softmax}(\mathbf{W}_s\mathbf{h}_i + \mathbf{b}_s) \qquad \mathbf{W}_s \in \mathbb{R}^{|\mathcal{S}|\times d}, \quad \mathbf{b}_s \in \mathbb{R}^{|\mathcal{S}|} \tag{1}$$

$$\mathbf{p}_{e,i} = \operatorname{softmax}(\mathbf{W}_e\mathbf{h}_i + \mathbf{b}_e) \qquad \mathbf{W}_e \in \mathbb{R}^{|\mathcal{S}|\times d}, \quad \mathbf{b}_e \in \mathbb{R}^{|\mathcal{S}|} \tag{2}$$

where $\mathcal{S} = \bigcup_i \mathcal{S}_i \cup \{\mathbb{O}\}$ is the set of all slots including the special no-slot label $\mathbb{O}$ which indicates that a token is not classified as a start/end token of a slot-filler. The vectors $\mathbf{p}_{s,i}, \mathbf{p}_{e,i}$ denote the predicted probability distribution over the slots that a token w_i is the start/end of the respective slots. The final prediction is determined by the arg max operation.

The predicted start/end tokens are joined sentence-wise by minimizing the distance between start and end tokens in terms of tokens in between. More precisely, for a given sentence, we first collect all predicted start and end tokens. For each predicted start token w_s, at position i we seek an end token w_e at position $j \geq i$ with matching label and minimal distance to w_s and assign it to w_s as its end token. Finally, we discard predicted start/end tokens which have no matching end/start token. This slightly differs from the IOB format [58], as only start and end token of a sequence are tagged and all tokens in between

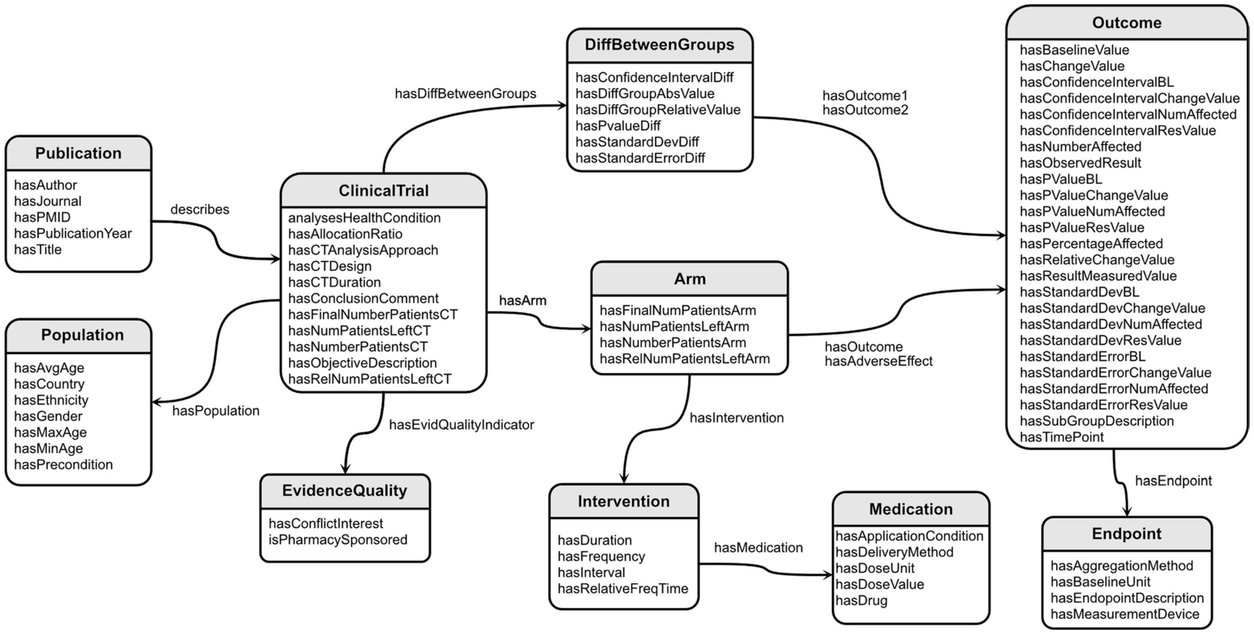


Fig. 2 Schema of the PICO data model used in the experiments

are classified just like tokens which are not part of any sequence. A comparison of both tagging schemes can be found in Table 2.

For each extracted slot-filler i with start/end tokens w_s resp. w_e with corresponding token representations $\mathbf{h}_s$, resp. $\mathbf{h}_e$, ITC computes a representation $\mathbf{e}_i$ by summing the representations of the start and end tokens followed by a dense layer with ReLU [59] activation function:

$$\mathbf{e}_i = \mathrm{relu}(\mathbf{W}_r(\mathbf{h}_s + \mathbf{h}_e) + \mathbf{b}_r) \qquad \mathbf{W}_r \in \mathbb{R}^{d\times d}, \quad \mathbf{b}_r \in \mathbb{R}^d \quad (3)$$

The learned representations $\mathbf{e}_i$ of the extracted slot-fillers (SFs) are then used as input to subsequent modules. In the remainder of this paper, we denote the set of all extracted slot fillers as $\mathcal{E}$, where each slot filler in $\mathcal{E}$ is represented by its vector representation computed by Eq. (3).

Assignment of textual slot-fillers to template instances

Typically, for some slot types like the textual slot fillers of the `Outcome` template, there are several slot fillers of the same type extracted from an original document. Therefore, we need a way to group these slot fillers such that actual template instances, e.g. multiple `Outcome` instances, can be created from these slot fillers. Deciding which slot fillers belong together is however not a trivial task.

The assignment of extracted SFs to template instances is therefore done in ITC by a clustering approach per template based on a pairwise similarity or compatibility function $q : \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, 1]$. q scores the similarity between two SFs in the sense that they belong to the same template instance, where $g(\mathbf{e}_i, \mathbf{e}_j) = 1$ indicates maximum similarity such that $\mathbf{e}_i$ and $\mathbf{e}_j$ should be assigned to the same template instance. Note that $\mathbf{e}_i$ and $\mathbf{e}_j$ are entity representations calculated based on the contextualized embeddings generated by the used models. Thus, we can use results from the established field of (density-based) clustering to figure out the SF grouping. The similarity function q is implemented in a slightly more complex way compared to the original paper, using two linear layers with a ReLU activation function in between and followed by a sigmoid activation function:

Table 2 Comparison of used tagging schema with the IOB format, where O represents tokens outside of a sequence and I-Frequency represents tokens which are part of a slot filler sequence of type frequency. In contrast, None represents tokens which are neither start nor end token of a slot filler, start:Frequency marks the start and end:Frequency the end of a frequency slot filler sequence

Tokens	NPH	insulin	once	or	twice	daily	in
IOB	O	O	I-Frequency	I-Frequency	I-Frequency	I-Frequency	O
Used	None	None	start:Frequency	None	None	end:Frequency	None

$$q'(\mathbf{e}_i, \mathbf{e}_j) = \mathrm{relu}(\mathbf{W}_h(\mathbf{e}_i + \mathbf{e}_j) + \mathbf{b}_h) \qquad \mathbf{W}_h \in \mathbb{R}^{d \times d}, \qquad \mathbf{b}_h \in \mathbb{R}^d \tag{4}$$

$$q(\mathbf{e}_i, \mathbf{e}_j) = \sigma(\mathbf{w}_s^T(q'(\mathbf{e}_i, \mathbf{e}_j)) + \mathbf{b}_s) \qquad \mathbf{w}_s \in \mathbb{R}^d, \qquad \mathbf{b}_s \in \mathbb{R} \tag{5}$$

Note that due to the symmetry of $+$, also q is a symmetric function, i.e. $q(\mathbf{e}_i, \mathbf{e}_j) = q(\mathbf{e}_j, \mathbf{e}_i)$ for all pairs of $\mathbf{e}_i, \mathbf{e}_j$. Then the mean pairwise similarity between SFs of a cluster $C_i \subseteq \mathcal{E}$ is given by

$$g(C_i) = \frac{1}{|C_i \times C_i|} \sum_{(\mathbf{e}_i, \mathbf{e}_j) \in C_i \times C_i} q(\mathbf{e}_i, \mathbf{e}_j) \tag{6}$$

The score of a clustering $\mathbb{C}_i = \{C_1, \ldots, C_{m_i}\}$ of SFs $\mathcal{E}_i \subseteq \mathcal{E}$ for template t_i is the mean score of its cluster scores:

$$h(\mathbb{C}_i) = \frac{1}{|\mathbb{C}_i|} \sum_{C_k \in \mathbb{C}_i} g(C_k) \tag{7}$$

The ITC approach seeks a clustering $\mathbb{C}_i^*$ of m_i clusters which maximizes the score given by Eq. (7):

$$\mathbb{C}_i^*(m_i) = \arg \max_{\mathbb{C}_i \in \mathcal{U}_{i,m_i}} h(\mathbb{C}_i) \tag{8}$$

where $\mathcal{U}_{i,m_i}$ denotes the set of all clusterings of the set $\mathcal{E}_i$ with m_i clusters. Note that the optimization objective defined by Eq. (8) is parameterized by the number of clusters m_i. In order to alleviate the assumption that the number of instances of templates needs to be known a priori, we propose a clustering step to induce the number of template instances per template type using Hierarchical Agglomerative Clustering (HAC) with a threshold based on the average of values computed for the training data, namely:

- the average similarity values of pairs belonging to the same template instance
- the average similarity values of pairs belonging to different instances

After the clustering $\mathbb{C}_i^*(m_i)$ has been estimated, the template instances t_{ij} are derived from those clusters $C_j^* \in \mathbb{C}_i^*(m_i)$. The slot to which a SF $\mathbf{e}_k \in C_j^*$ is assigned is given by the label assigned by the SF extraction module by Eqs. (1) and (2). In summary, the assignment of SFs to template instances is done as follows:

1. For each template t_i, the set $\mathcal{E}_i \subseteq \mathcal{E}$ of SFs which can be assigned to instances of template type t_i is estimated.
2. Equation (8) or Agglomerative Hierarchical Clustering is used to find some clustering of the SFs in $\mathcal{E}_i$.
3. The template instances are derived from the clusters in the clustering.

As an example, we consider the following four extracted slot fillers:

1. PercentageAffected: 16
2. PercentageAffected: 8
3. TimePoint: week 24
4. TimePoint: week 12

Additionally, we assume our trained similarity function gives us the similarities presented in Table 3.

Given these similarities and a clustering threshold of, e.g., 0.5, this results in two clusters which can be then directly used to create the corresponding `Outcome` template instances. These two clusters are:

1. PercentageAffected: 16 and TimePoint: week 24
2. PercentageAffected: 8 and TimePoint: week 12

The clustering thus provides a robust and flexible way to both determine the number of template instances to generate as well as the groups of slot fillers those instances comprise.

Generative approach

In this section we propose a simple generative approach for extracting template instances from unstructured text based on the Transformer [60] encoder-decoder model. As encoder-decoder models require the output to be a linear token sequence, the set of TIs needs to be converted into a sequence of tokens. In Section "Linearization of sets of template instances", we present a simple recursive method for linearizing sets of TIs along a context free grammar (CFG) for describing the linearized structures. In Section "Decoding" we adopt the presented CFG for generating valid token sequences representing sets of TIs.

Table 3 Example similarities/compatibilities between four slot fillers, slot types in first row have been omitted

	16	**8**	**week 24**	**week 12**
PercentageAffected: 16	-	0.1	0.7	0.4
PercentageAffected: 8	0.1	-	0.3	0.8
TimePoint: week 24	0.7	0.3	-	0.2
TimePoint: week 12	0.4	0.8	0.2	-

Transformer-based encoder-decoder models

Transformer-based [60] encoder-decoder models are seq2seq models which haven been used on a variety of natural language processing tasks like machine translation [61] and text summarization [62]. The encoder part of the Transformer learns a contextualized representation of the input tokens $w_1, \ldots, w_n$ via multi-headed self-attention [60], converting the input sequence into a sequence of vectors $\mathbf{h}_1, \ldots, \mathbf{h}_n \in \mathbb{R}^d$, where d is the dimension of the Transformer model. Then the decoder part takes the vector sequence from the encoder as input and produces an output vector sequence $\mathbf{d} = (\mathbf{d}_1, \ldots, \mathbf{d}_n \in \mathbb{R}^d)$ via multi-headed cross-attention. The computational complexity of self-attention grows quadratically with the number of tokens. Beltagy et al. [16] proposed the Longformer encoder-decoder, which combines local and global multi-headed self-attention in the encoder, reducing computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

The output vector sequence $\mathbf{d}$ is used to compute a probability distribution over the vocabulary of the underlying model via the following equation:

$$p(y_i|x, y_1, \ldots, y_{t-1}) = \text{softmax}(\mathbf{v}_i^T \mathbf{d}_{t-1} + b_i) \tag{9}$$

where $\mathbf{v}_i \in \mathbb{R}^d$ is the embedding of token y_i, b_i is a bias for token y_i, $\mathbf{d}_{t-1}$ is the output vector of the decoder at position $(t-1)$ and d is the model dimension. The probability of token y_t at position t is conditioned on the input token sequence x and the past decoded tokens $y_1, \ldots, y_{t-1}$. This dependence is encoded through the vector $\mathbf{d}_{t-1}$ via multi-headed self- and cross-attention.

Token prediction in the decoder is done by maximum a posteriori probability (MAP) inference. Hence the predicted token at position i is given by the token with maximal posterior probability:

$$y_t = \arg\max_i p(y_i|x, y_1, \ldots, y_{t-1}) \tag{10}$$

The generative model is trained via teacher forcing by minimizing the cross entropy loss between the predicted token distribution described by Eq. (9) and the ground truth label.

Linearization of sets of template instances

As encoder-decoder models expect the output space to be token sequences, we present a simple recursive linearization procedure of template instances (TIs). First, note that TIs are described by the content of their slots (i.e., their slot-fillers), and that slot-fillers can be either text spans from the input document or other TIs. Hence the recursion base is given by the linearization of textual slot-fillers. Let $f = w_{k_1}, \ldots, w_{k_m}$ be a token sequence which represents a textual slot-filler f for a slot of name `SLOT`. Then the linearization of this slot-filler is the token sequence itself enclosed by the special tokens `[start:SLOT]` and `[end:SLOT]`, i.e. `[start:SLOT]` $\odot w_{k_1} \odot \ldots \odot w_{k_m} \odot$ `[end:SLOT]`, where $\odot$ denotes the concatenation of tokens. If the slot-filler is a TI, then it is recursively linearized and the resulting token sequence is enclosed by the special tokens `[start:SLOT]` and `[end:SLOT]`. The linearization of TIs is described below.

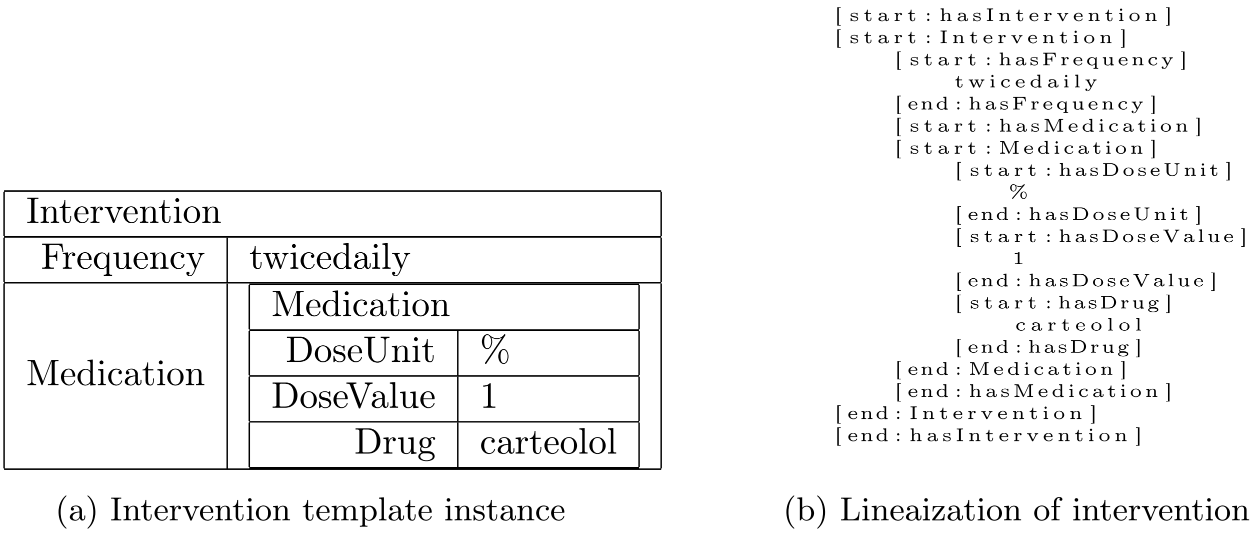
In general, more than one slot-filler can be assigned to a slot of a TI. Therefore, we denote the complete content of a slot as a set $\mathcal{F}$ of slot-fillers. As sets, in contrast to sequences, are unordered constructs by definition, the linearization of sets of slot-fillers is inherently ambiguous. To get an unambiguous order, we introduce a slot ordering operator ω which converts sets of slot-fillers into sequences of slot-fillers according to predefined criteria (e.g. position within input document in case of textual slot-fillers). Then sets $\mathcal{F}$ of slot-fillers are linearized as follows: First, we sort the elements of $\mathcal{F}$ according to the sorting operator ω and obtain a sequence F of slot-fillers. Then we linearize each slot-filler in F as described above and concatenate the resulting token sequences, respecting the ordering of slot-fillers in F.

Next, we describe the linearization of TIs. As TIs are represented by the content of their slots, the linearization of a TI has to include the linearization of its slots. However, a template does not impose any ordering of its slots, and hence the linearization order of the slots of a TI is undefined. Therefore, we introduce another ordering operator Ω which orders the slots of a template. Then the linearization of a TI is the concatenation of the linearizations of its slots according to the ordering of its slots given by the ordering operator Ω.

Any set of TIs induces a graph with TIs as nodes and links between TIs as edges. Recall that there is a link from TI t_{ij} to TI t_{kl} iff t_{kl} is a slot-filler of t_{ij}. In order to guarantee that the linearization algorithm described above is well defined, we require the induced graph to be 1) acyclic and 2) connected. The first requirement ensures that the linearization algorithm terminates, while the second ensures the absence of isolated TI, which can not be linearized.

However, choosing ω and Ω is only necessary for training but not for inference purposes, as the decoding allows to fill template slots in any order. Therefore, we choose arbitrary but fixed ω and Ω for the experiments described in the "Experimental results" Section.

A full example for a whole linearized publication template instance can be found in Listing 2 in Appendix 6. A shorter example for an intervention template instance with both textual and template slot fillers can be found in Fig. 3.



Intervention		
Frequency	twicedaily	
Medication	Medication	
	DoseUnit	%
	DoseValue	1
	Drug	carteolol

(a) Intervention template instance

```
[start:hasIntervention]
[start:Intervention]
     [start:hasFrequency]
          twicedaily
     [end:hasFrequency]
     [start:hasMedication]
     [start:Medication]
          [start:hasDoseUnit]
               %
          [end:hasDoseUnit]
          [start:hasDoseValue]
               1
          [end:hasDoseValue]
          [start:hasDrug]
               carteolol
          [end:hasDrug]
     [end:Medication]
     [end:hasMedication]
[end:Intervention]
[end:hasIntervention]
```

(b) Lineaization of intervention

Fig. 3 Illustration of linearization of an intervention template instance

A context-free grammar for describing linearization of sets of template instances

In the following, we describe the linearization of sets of TIs (described in Section "Linearization of sets of template instances") by a context-free grammar (CFG) which is used in the decoding process ("Decoding" Section) to constrain the generation of tokens. A CFG is defined by a 4-tuple $\mathcal{G} = (N, T, R, S)$, where N is a set of non-terminal symbols, T is a set of terminal symbols, R is a set of production rules and $S \in T$ is the start symbol of the grammar. The set of terminal symbols is defined by the vocabulary of the underlying encoder-decoder model together with some special tokens for defining the production rules R. The recursion base of the linearizations of sets of TIs is given by the linerization of textual slots which we describe by the following equation:

$$\texttt{TEMPLATE} \;:=\; \texttt{[start:SLOT]}\ \texttt{TEXT}\ \texttt{[end:SLOT]}\ \texttt{TEMPLATE} \tag{11}$$

where TEMPLATE and SLOT are placeholders for names of template and slots, respectively, TEXT is a placeholder for any token sequence from the input document and [start:SLOT], [end:SLOT] are special tokens enclosing the textual slot-filler. Eq. (11) schematically defines production rules for textual slot-fillers, and TEMPLATE is the non-terminal symbol which is used to identify the respective production rules. Note that the non-terminal symbol TEMPLATE on the right-hand side of Eq. (11) allows recursion and hence the application of more the one production rule associated with the non-terminal symbol TEMPLATE. The recursion base of production rules is given by

$$\texttt{TEMPLATE} \;:=\; \texttt{[end:TEMPLATE]} \tag{12}$$

where TEMPLATE is again a placeholder for the template name and [end:TEMPLATE] is a special token indicating the end of the linearization of the template TEMPLATE. Production rules for TIs are described by

$$\texttt{TEMPLATE\_HEAD} \;:=\; \texttt{[start:TEMPLATE]}\ \texttt{TEMPLATE} \tag{13}$$

Analogously to the production rules defined by Eq. (11) for textual slot-fillers, the production rules for slots containing TIs as slot-fillers is defined by

$$\texttt{TEMPLATE} \;:=\; \texttt{[start:SLOT]}\ \texttt{TEMPLATE\_HEAD}\ \texttt{[end:SLOT]}\ \texttt{TEMPLATE} \tag{14}$$

where TEMPLATE_HEAD is a placeholder for any non-terminal symbol whose associated production rules are derived from Eq. (13). Listing 1 shows the production rules for the data model used in our experiments.

Decoding

In Section "A context-free grammar for describing linearization of sets of template instances", we presented a CFG which describes valid token sequences representing a set of TIs. In this section, we describe a simple method to constrain token prediction such that only such token sequences are generated which are valid according the CFG. For example, consider a slot Drug which can have textual slot-fillers for describing drug names for a medication. After the special token `[start:Drug]` has been predicted, we know that the set of next possible tokens would consist of all tokens from the input document plus the special token `[end:Drug]`. This information is encoded by the CFG, and the decoding method described in this section uses this information to constrain token prediction.

In this paper, we slightly generalize the constrained decoding approach of Lu et al. [20] to arbitrary right-linear CFGs by applying a strategy similar to recursive descent parsing.

Beginning with a start symbol, in our case `PUBLICATION_HEAD`, the set of possible next tokens is calculated in each decoding step. This set is then used to generate a mask for the model vocabulary to discard all tokens which would not comply with the production rules of the CFG. From the remaining tokens, we select the token with the maximum value in a greedy fashion. The implementation of a beam search to optimize the decoding output even more remains for future work.

To keep track of the decisions and possible next tokens, a stack data structure is used to guide the decoding. Whenever a start token of a slot like `[start:NumberAffected]` is chosen as the decoded token, this decision is saved by adding this to the decoding stack. This is then used to constrain the tokens in the next step to be only those which can follow a `[start:NumberAffected]` token. Similarly, when an end token like `[end:NumberAffected]` is chosen, the top stack element is removed from the stack.

This way, the decoding is guided to comply with the requirements imposed by the CFG and this way ensuring the output can then be parsed into actual TIs.

Experimental results

In this section, we discuss the setting of our experiments as well as the results of those experiments.

Experimental setting

In our experiments, we use the same dataset as Witte and Cimiano [14] for type 2 diabetes and glaucoma. The dataset comprises a total of 211 documents for two diseases: type 2 diabetes (104) and glaucoma (107). The 104 type 2 diabetes documents are split up into a training, validation and test sets of size 68, 16 and 20, respectively. Analogously, the 107 glaucoma documents are split up into a training, validation and test sets of size 69, 17 and 21, respectively. We use the same fixed train-validation-test split and run separate experiments for those two diseases. Both the extractive and the generative approach were then evaluated using multiple base models, namely `allenai/longformer-base-4096` [16] for the extractive approach and `allenai/led-base-16384` [16] as well as `google/flan-t5-base` [17] for both approaches. As the extractive approach requires just an encoder whereas the generative approach needs a decoder due to its seq2seq nature, we compare two encoder-decoder models from which only the encoder is used in the extractive approach. Additionally, we also evaluate an encoder-only model for the extractive approach to ensure the partial usage of the encoder-decoder models does not harm the performance.

For these models and diseases, we then run hyperparameter optimizations using Optuna [63] with 30 trials each and measuring performance using validation F_1 scores. In each trial, an initial learning rate (between $1e^{-3}$ and $1e^{-5}$, using logarithmic domain) and a λ for the lambda learning rate scheduler (between 0.9 and 1.0, using logarithmic domain, learning rate calculated with $lr(epoch) = \lambda^{epoch}$) are sampled from Optuna. The used batch size is 1 and the number of epochs is 50 in all experiments. Each experiment is then run on a single NVIDIA A40 GPU. The best hyperparameters for each disease-approach-model-combination are then used to train 10 additional models. Unless stated differently, mean and standard deviation in tables refer to the different results of these 10 training runs. The means and standard deviations of the test F_1 scores of these 10 trained models are listed in Table 4 for each combination.

Slot-filler extraction results

In all categories, the extractive approach paired with the `flan-t5-base` model performs best. In summary, for glaucoma, the extractive approach performs best with model `flan-t5-base` and a mean test F_1 score of 0.636 (±0.006 standard deviation across the 10 training runs with the best found hyperparameters of the category). This way, it outperforms the other tested models of the extractive approach as well as all models of the generative approach by 0.02 or more. For type 2 diabetes, the extractive approach performs best as well with model `flan-t5-base` and a mean F_1 score of 0.547 (±0.006 standard deviation). This indicates that the extractive approach is superior to the generative approach, although the lead is much smaller for type 2 diabetes than for glaucoma.

Table 5 shows the mean F_1 scores per template on the type 2 diabetes and glaucoma test set. The table shows the values of the best models of each category (w.r.t. validation F_1 score), i.e. the `flan-t5-base` models in all four cases. The mean F_1 values are calculated for each of the 10 models trained using the best hyperparameters of their respective category. The values in the

Table 4 Mean and standard deviation σ of test F_1 scores across 10 models trained using best-performing (F_1 on validation dataset) configuration found in 30 trials of hyperparameter optimization. Numbers rounded to three decimal places, best configuration of each disease marked bold

Type 2 diabetes		**Glaucoma**	
Model	**Mean** $F_1 (\pm\sigma)$	**Model**	**Mean** $F_1 (\pm\sigma)$
Extractive			
flan-t5-base	**0.547 (± 0.006)**	**flan-t5-base**	**0.636 (± 0.006)**
led-base-16384	0.525 (± 0.009)	led-base-16384	0.572 (± 0.010)
longformer-base-4096	0.540 (± 0.008)	longformer-base-4096	0.613 (± 0.007)
Generative			
flan-t5-base	0.539 (± 0.029)	flan-t5-base	0.584 (± 0.025)
led-base-16384	0.400 (± 0.079)	led-base-16384	0.353 (± 0.106)

Table 5 Mean slot F_1 values per template. Each cell shows mean and standard deviation of 10 training runs with the best found hyperparameters for best (w.r.t. validation F_1 score) configurations of each category. Numbers rounded to two decimal places, best values marked bold

	Type 2 diabetes $F_1 (\pm\sigma)$		**Glaucoma** $F_1 (\pm\sigma)$	
Template name	**Generative**	**Extractive**	**Generative**	**Extractive**
Arm	0.7 (± 0.21)	**0.87 (± 0.02)**	0.34 (± 0.06)	**0.36 (± 0.04)**
ClinicalTrial	0.62 (± 0.02)	**0.82 (± 0.02)**	0.63 (± 0.03)	**0.78 (± 0.02)**
DiffBetweenGroups	0.41 (± 0.06)	**0.45 (± 0.03)**	0.28 (± 0.08)	**0.37 (± 0.04)**
Endpoint	0.39 (± 0.03)	**0.43 (± 0.01)**	0.33 (± 0.04)	**0.42 (± 0.09)**
Intervention	0.61 (± 0.06)	**0.62 (± 0.02)**	0.26 (± 0.02)	**0.42 (± 0.12)**
Medication	**0.48 (± 0.02)**	0.34 (± 0.02)	**0.62 (± 0.08)**	0.53 (± 0.02)
Outcome	**0.2 (± 0.03)**	0.11 (± 0.01)	0.35 (± 0.04)	**0.38 (± 0.01)**
Population	0.22 (± 0.03)	**0.52 (± 0.07)**	**0.56 (± 0.04)**	0.52 (± 0.03)
Publication	0.95 (± 0.03)	**0.96 (± 0.01)**	0.86 (± 0.02)	**0.9 (± 0.02)**

table correspond to the mean and standard deviation of those mean F_1 scores per template. The generative approach performs better than the extractive one on the Medication templates (0.48 vs. 0.34 and 0.62 vs. 0.53 F_1 score for type 2 diabetes and glaucoma, respectively). On the Population and Outcome template, the results are mixed with one approach performing better for one disease dataset but not for the other. On all six remaining templates, the extractive approach performs better, although with different margins.

Mean F_1 scores per slot are shown in Table 7 in the Appendix 2, again with mean and standard deviation (of the mean F_1 scores) calculated for the 10 models trained using the best hyperparameters of their respective category. The F_1 scores of the different slots range from over 0.9, e.g. PMID or PublicationYear, to below 0.1, e.g. FinalNumPatientsArm or ObservedResult. There are also some noticeable differences between the diseases, with Journal achieving scores of 0.96 and 0.92 for type 2 diabetes in contrast to 0.67 and 0.74 for glaucoma. There are also slots where one approach performs better than the other across both datasets, e.g. DoseUnit (0.77/0.8 generative vs. 0.24/0.6 extractive) and NumberPatientsCT (0.65/0.65 generative vs. 0.93/0.86 extractive).

Joint training on both datasets

Additionally to the main experiment described above, we ran another small experiment, training the best-performing generative and extractive model (flan-t5-base in both cases) with the best-performing respective parameters in 10 trials on the union of the type 2 diabetes and glaucoma training, validation and test datasets, respectively. The resulting models are then again evaluated on the separated datasets for comparability reasons. The resulting mean F_1 scores ($\pm\sigma$) for the generative approach are 0.556 (± 0.026) for type 2 diabetes and 0.626 (± 0.015) for glaucoma. For the extractive approach, the mean F_1 scores ($\pm\sigma$) are 0.560 (± 0.007) for type 2 diabetes and 0.644 (± 0.008) for glaucoma. Therefore, the performance increases for both datasets and both approaches compared to the original results trained on the separated datasets. Moreover, the generative approach achieves comparable performance to the scores of the extractive approach trained on the separated datasets. At the same time, the extractive approach gets even better when also trained on both datasets at the same time.

Considering the relatively small datasets, this might indicate that performance for both diseases benefits from similar data in the other dataset, respectively. Therefore, we are optimistic that the training of a single general model (in contrast to specialized models for each disease as described in the main experiment) is possible with comparable or even better performance on diseases the model has been trained on (i.e., in-distribution data) and acceptable performance on different but similar diseases (i.e., out-of-distribution data). However, another dataset would be necessary to test this hypothesis such that this remains to be investigated in future work.

Inferred template cardinality results

In this section, we evaluate the ability of our models to infer the correct number of instances for each template type. For this, we compare the number of inferred templates to the number of instances in the gold standard by computing the mean abolsute deviation. Table 6 shows the mean absolute deviation between the ground truth and predicted template cardinality of the best extractive and generative model on the type 2 diabetes and glaucoma test sets. The mean absolute deviation values are calculated separately for each of the 10 models trained using the best hyperparameters of their respective category. The values in the table are then mean and standard deviation of those mean absolute deviations across the respective 10 trained models. Additionally, in Appendix 5, the corresponding mean ground truth (GT) and predicted template cardinalities are listed in order to allow a judgement whether or not a certain deviation is high. Note that the templates `Publication`, `ClinicalTrial` and `Population` are not mentioned in these tables as their cardinality is always one.

On the type 2 diabetes dataset, the extractive approach yields better results than the generative approach in terms of template cardinality prediction for the `DiffBetweenGroups`, `Endpoint` and `Medication` templates, whereas the generative approach yields better results for the `Arm`, `Intervention` and `Outcome` templates. On the glaucoma dataset, the generative approach performs better than the extractive one in terms of cardinality inference on all templates except `DiffBetweenGroups` (0.39 vs. 0.17) and `Endpoint` (2.91 vs. 0.35).

Discussion

The overall slot-filler extraction results of both models in terms of micro F_1 measure indicate that the extractive approach is slightly superior to the generative approach, although the margin is especially small for the type 2 diabetes dataset (cf. Table 4). Moreover, the mean F_1 scores per template (Table 5) suggest that the extractive approach performs better than the generative one on most templates on both datasets.

However, the full picture is a little more complex and both approaches have areas in which they perform better or worse than the other one and vice versa, and that for a variety of reasons.

First, it is noticeable that the F_1 scores for glaucoma are, on average, higher than those for type 2 diabetes. Nevertheless, the difference between the results for both datasets is not the same for both approaches, although the trend is the same. For the generative approach, the

Table 6 Mean absolute deviation between ground truth and predicted template cardinality. Each cell shows mean and standard deviation of 10 training runs with the best found hyperparameters for best (w.r.t. validation F_1 score) configurations of each category. Numbers rounded to two decimal places, best values marked bold

	Type 2 diabetes		**Glaucoma**	
Template name	**Generative**	**Extractive**	**Generative**	**Extractive**
Arm	**0.01 (± 0.02)**	1.09 (± 0.03)	**0.02 (± 0.04)**	1.33 (± 0.05)
DiffBetweenGroups	1.01 (± 0.96)	**0.69 (± 0.08)**	0.39 (± 0.36)	**0.17 (± 0.13)**
Endpoint	5.32 (± 1.11)	**3.83 (± 0.03)**	2.91 (± 1.09)	**0.35 (± 0.22)**
Intervention	**0.1 (± 0.04)**	1.22 (± 0.03)	**0.19 (± 0.05)**	0.8 (± 0.1)
Medication	0.21 (± 0.07)	**0.18 (± 0.1)**	**0.11 (± 0.08)**	0.47 (± 0.2)
Outcome	**1.08 (± 0.82)**	8.36 (± 0.07)	**0.94 (± 0.57)**	2.98 (± 0.04)

performance of the best-performing flan-t5-base model decreases by just 0.045 (around 7.7% relatively) and the led-base-16384 version even increases its mean performance.

In contrast, the best-performing extractive version, again flan-t5-base, loses 0.089 (around 14% relatively) in terms of F_1 performance - relatively almost twice as much as the generative approach. This may indicate that the extractive approach is better able to exploit certain characteristics which are specific to the glaucoma dataset and which are not present in the type 2 diabetes dataset, whereas the generative approach is more robust against those differences - both in a positive and in a negative way - and that way maybe generalizing a little more due to the more complex nature of the seq2seq task. However, it is not clear which properties of the data cause this deviation.

Considering robustness and the different complexity of the tasks of the extractive and generative task, this is to some degree also mirrored by the standard deviations of the two approaches. While the standard deviation for the extractive approach is not greater than 0.01, the standard deviation of the generative models is not smaller than 0.025 and gets up to 0.106 for led-base-16384. Therefore, it is more than doubled at least compared to the extractive approach.

Moreover, the standard deviation appears to be correlated to the chosen model, with flan-t5-base giving the lowest deviation, followed by (for the extractive part) longformer-base-4096 and finally led-base-16384 consistently across both datasets.

The different strengths and weaknesses of both approaches become even more apparent examining the different performances separated by templates (Table 5) and, ultimately, single slots (Table 7 in the Appendix 2).

For whole templates, Table 5 shows an in parts mixed picture of which approach performs best. In many cases in which the extractive approach performs best, both approaches perform similarly well (e.g., Publication). However, there are also different cases like Clinical Trial where the margin is larger, but also Medication where the generative approach outperforms the extractive approach by around 0.1 although the standard deviation is also quite high for the generative glaucoma case. In other cases there are large differences between the two datasets, which is also true for the evaluation per slot.

As an example for unexpected single slot differences, consider the Journal slot. One would expect the recognition of the Journal slot to be a comparably simple task across both datasets. However, the performance greatly differs between the datasets, although both approaches achieve good scores on this slot. For the type 2 diabetes dataset, the performance is nearly perfect with scores above 0.9. In contrast, the scores for the glaucoma dataset are still good but much worse with scores around 0.7. The different possible slot fillers are shown in Table 9 in the Appendix 4. Looking at the different slot fillers, it is not immediately clear why the diabetes case is so much easier for both approaches than the glaucoma case. Both tables have approximately the same number of different entries and in both cases the journal names are in many cases trivial to recognize (containing either Diabetes or Ophthalmol).

However, the distribution of occurrences might partially explain the performance differences here. Although both datasets have similar number of Journal slot fillers with up to three occurrences, only the type 2 diabetes dataset has (even multiple) Journal slot fillers with a high number of occurrences (more than ≈ 8, e.g.). Therefore, the reason why the Journal slot appears to be so much easier to recognize in the type 2 diabetes dataset might not be due to the textual form of the slot fillers but instead because fewer slot fillers account for a larger majority of the general slot occurrences compared to the glaucoma dataset. The absolute numbers and differences are still quite small, however, but this might allow to get much better scores just by recognizing two or three Journal slot fillers. There may be many more examples which are not discussed here.

All in all it is not clear in all cases what properties of the data cause those partial differences in performance. However, it underlines on the one hand how much data variance can influence information extraction approaches like the two presented ones. On the other hand, this also emphasizes how both approaches can have different strengths and weaknesses and a flat evaluation only considering the final single performance score does not do justice to the complex nature of the task.

Case study

Similarly to the work by Witte and Cimiano [14], we conduct a case study on a single RCT abstract in which we compare the predicted and ground truth results for one exemplary document out of the type 2 diabetes test dataset. For this case study, we use the same publication as considered by Witte and Cimiano [14] which is the one by Shankar et al. [64]. The results of this case study can be found in Table 8 in the Appendix 3.

Both the extractive and the generative approach succeed in extracting the basic characteristics of the trial which are part of the Publication template, e.g. authors, title and publication year. This is consistent with the results of Table 5, which indicate that Publication is an especially easy template to extract. Similarly, the ClinicalTrial and Medication instances are, except some small errors, extracted almost perfectly. The template instance for the used Intervention is also extracted without errors by both approaches, which is a little more surprising taking into account the slightly lower score of around 0.6. Moreover, both approaches correctly predict that there are no textual slot fillers of the Arm template in the text.

For the Population template instance, we first encounter moderate differences to the gold standard. Although both approaches manage to extract USA as slot fillers for the Country slot, both fail to extract the second slot filler Australia as well as Ethnicity. The latter is at least in line with the fact that the first gold standard precondition - mentioning the ethnicity of the patients - is not recognized by both approaches. For the second Precondition slot filler, both approaches get a part of it but not the full slot filler, with the generative approach recognizing a slightly larger part of the actual slot filler. This is to some degree unexpected, as the mean performance of the extractive approach on the Population templates of the type 2 diabetes dataset is more than twice as high as the score of the generative approach.

For the DiffBetweenGroups template, the extractive approach returns a perfect result in this case, whereas the generative approach misses the $P < 0.001$ slot filler but delivers a duplicate of the $P = 0.013$ slot filler. The mean results of Table 5 suggest similar performance, which is not the case here.

For the Endpoint template instances, both approaches manage to extract most slot fillers at least partially but show issues grouping them together correctly. The extractive approach puts all of the extracted slot fillers in just two instances, missing most instances of the gold standard. For the generative approach, however, it is the other way around and too many instances (containing some duplicates) are generated. Nevertheless, some of the generated instances are correct and in some cases there is just a part missing. Generally, the performance is rather unsatisfying here but is consistent with the comparably poor mean performance of around 0.4 on the Endpoint template, indicating this is an especially hard template to extract.

However, the situation is even worse for the Outcome template instances, which was to be expected considering the mean performance on the type 2 diabetes dataset of just 0.2 and 0.11 for the generative and extractive approach, respectively. Again, both approaches at least partially recognize most slot fillers, but fail to group them together correctly. Similarly to the Endpoint template instances, the extractive approach generates too few instances whereas the generative approach generates more instances. Nevertheless, those instances are not entirely correct in most cases. This suggests future work has to improve this grouping beyond simple similarity calculations or fully relying on the language model and constrained decoding.

Taken together, the current results, while promising, are not accurate enough to support the full automatic creation of a systematic review as proposed by Sanchez-Graillet et al. [10]. However, the proposed approach could considerably reduce the workload for teams to extract key information from a set of publications in the sense proposed by Thomas et al. [65]. The results, however, would need to be manually controlled. While the approach is not yet suited to support the full creation of a systematic review at high-quality, it could be used to summarize the existing literature in a cost-effective fashion to allow researchers to get a first overview of existing clinical evidence or as a basis to form hypothesis to be validated further on.

Conclusion

We have presented an extended extractive and a generative approach for extracting structured information from Randomized Controlled Trial abstracts, which can both support clinicians in finding best therapies on the basis of clinical evidence and in creating systematic reviews of the whole body of available clinical evidence. The extractive approach is realized by a two-step architecture which first extracts slot-fillers from the input document, followed by a clustering step which assigns the extracted slot-fillers to template instances. The best models of this approach yield an average F_1 score of 0.547 on type 2 diabetes and 0.636 on glaucoma test sets, respectively. In the generative approach, the structured information given by the template instances is encoded as a linear token sequence which is decoded at inference time by utilizing a context-free grammar for guidance. The best models of the generative approach yield an average F_1 score of 0.539 on type 2 diabetes and 0.584 on glaucoma test sets, respectively.

Future work should investigate whether the lead of the extractive approach persists when the base models of both approaches are scaled up, e.g. by using flan-t5-large, flan-t5-xl or even flan-t5-xxl or other large language models. The benefits of the extractive and generative approach could also be combined by adding a pointer network to the generative model. We will also investigate whether integrating a pointer network into the generative model can improve results. It would be also interesting to test the results in an actual evidence generation and comparison case study to assess whether the approach can indeed support the process of summarizing results from the clinical literature for a particular research question.

Appendix 1 Grammar definition

```
PUBLICATION_HEAD := [start:Publication] PUBLICATION
PUBLICATION := [start:describes] CLINICAL_TRIAL_HEAD [end:describes] PUBLICATION
PUBLICATION := [start:Title] TEXT [end:Title] PUBLICATION
PUBLICATION := [start:Author] TEXT [end:Author] PUBLICATION
PUBLICATION := [start:PublicationYear] TEXT [end:PublicationYear] PUBLICATION
PUBLICATION := [start:Journal] TEXT [end:Journal] PUBLICATION
PUBLICATION := [start:PMID] TEXT [end:PMID] PUBLICATION
PUBLICATION := [end:Publication]

CLINICAL_TRIAL_HEAD := [start:ClinicalTrial] CLINICAL_TRIAL
CLINICAL_TRIAL := [start:DiffBetweenGroups] DIFF_BETWEEN_GROUPS_HEAD [end:
↪ DiffBetweenGroups] CLINICAL_TRIAL
CLINICAL_TRIAL := [start:Arm] ARM_HEAD [end:Arm] CLINICAL_TRIAL
CLINICAL_TRIAL := [start:EvidQualityIndicator] EVIDENCE_QUALITY_HEAD [end:
↪ EvidQualityIndicator] CLINICAL_TRIAL
CLINICAL_TRIAL := [start:Population] POPULATION_HEAD [end:Population] CLINICAL_TRIAL
CLINICAL_TRIAL := [start:ObjectiveDescription] TEXT [end:ObjectiveDescription]
↪ CLINICAL_TRIAL
CLINICAL_TRIAL := [start:NumberPatientsCT] TEXT [end:NumberPatientsCT] CLINICAL_TRIAL
CLINICAL_TRIAL := [start:CTDuration] TEXT [end:CTDuration] CLINICAL_TRIAL
CLINICAL_TRIAL := [start:AllocationRatio] TEXT [end:AllocationRatio] CLINICAL_TRIAL
CLINICAL_TRIAL := [start:CTAnalysisApproach] TEXT [end:CTAnalysisApproach] CLINICAL_
↪ TRIAL
CLINICAL_TRIAL := [start:CTDesign] TEXT [end:CTDesign] CLINICAL_TRIAL
CLINICAL_TRIAL := [start:ConclusionComment] TEXT [end:ConclusionComment] CLINICAL_
↪ TRIAL
CLINICAL_TRIAL := [start:analysesHealthCondition] TEXT [end:analysesHealthCondition]
↪ CLINICAL_TRIAL
CLINICAL_TRIAL := [start:FinalNumPatientsCT] TEXT [end:FinalNumPatientsCT] CLINICAL_
↪ TRIAL
CLINICAL_TRIAL := [start:NumPatientsLeftCT] TEXT [end:NumPatientsLeftCT] CLINICAL_
↪ TRIAL
CLINICAL_TRIAL := [start:RelNumPatientsLeftCT] TEXT [end:RelNumPatientsLeftCT]
↪ CLINICAL_TRIAL
CLINICAL_TRIAL := [end:ClinicalTrial]

POPULATION_HEAD := [start:Population] POPULATION
POPULATION := [start:MinAge] TEXT [end:MinAge] POPULATION
POPULATION := [start:MaxAge] TEXT [end:MaxAge] POPULATION
POPULATION := [start:AvgAge] TEXT [end:AvgAge] POPULATION
POPULATION := [start:Precondition] TEXT [end:Precondition] POPULATION
POPULATION := [start:Gender] TEXT [end:Gender] POPULATION
POPULATION := [start:Ethnicity] TEXT [end:Ethnicity] POPULATION
POPULATION := [start:Country] TEXT [end:Country] POPULATION
POPULATION := [end:Population]

DIFF_BETWEEN_GROUPS_HEAD := [start:DiffBetweenGroups] DIFF_BETWEEN_GROUPS
DIFF_BETWEEN_GROUPS := [start:Outcome] OUTCOME_HEAD [end:Outcome] DIFF_BETWEEN_GROUPS
DIFF_BETWEEN_GROUPS := [start:DiffGroupRelativeValue] TEXT [end:DiffGroupRelativeValue
↪ ] DIFF_BETWEEN_GROUPS
DIFF_BETWEEN_GROUPS := [start:DiffGroupAbsValue] TEXT [end:DiffGroupAbsValue] DIFF_
↪ BETWEEN_GROUPS
DIFF_BETWEEN_GROUPS := [start:PvalueDiff] TEXT [end:PvalueDiff] DIFF_BETWEEN_GROUPS
DIFF_BETWEEN_GROUPS := [start:ConfidenceIntervalDiff] TEXT [end:ConfidenceIntervalDiff
↪ ] DIFF_BETWEEN_GROUPS
DIFF_BETWEEN_GROUPS := [start:StandardDevDiff] TEXT [end:StandardDevDiff] DIFF_BETWEEN
↪ _GROUPS
DIFF_BETWEEN_GROUPS := [start:StandardErrorDiff] TEXT [end:StandardErrorDiff] DIFF_
↪ BETWEEN_GROUPS
DIFF_BETWEEN_GROUPS := [end:DiffBetweenGroups]

ARM_HEAD := [start:Arm] ARM
ARM := [start:Outcome] OUTCOME_HEAD [end:Outcome] ARM
ARM := [start:AdverseEffect] OUTCOME_HEAD [end:AdverseEffect] ARM
ARM := [start:Intervention] INTERVENTION_HEAD [end:Intervention] ARM
ARM := [start:NumberPatientsArm] TEXT [end:NumberPatientsArm] ARM
ARM := [start:FinalNumPatientsArm] TEXT [end:FinalNumPatientsArm] ARM
ARM := [start:NumPatientsLeftArm] TEXT [end:NumPatientsLeftArm] ARM
ARM := [start:RelNumPatientsLeftArm] TEXT [end:RelNumPatientsLeftArm] ARM
ARM := [end:Arm]

INTERVENTION_HEAD := [start:Intervention] INTERVENTION
INTERVENTION := [start:Medication] MEDICATION_HEAD [end:Medication] INTERVENTION
INTERVENTION := [start:Frequency] TEXT [end:Frequency] INTERVENTION
INTERVENTION := [start:Interval] TEXT [end:Interval] INTERVENTION
INTERVENTION := [start:Duration] TEXT [end:Duration] INTERVENTION
INTERVENTION := [start:RelativeFreqTime] TEXT [end:RelativeFreqTime] INTERVENTION
INTERVENTION := [end:Intervention]

OUTCOME_HEAD := [start:Outcome] OUTCOME
OUTCOME := [start:EndTEXT] ENDTEXT_HEAD [end:EndTEXT] OUTCOME
OUTCOME := [start:BaseLineValue] TEXT [end:BaseLineValue] OUTCOME
OUTCOME := [start:PValueBL] TEXT [end:PValueBL] OUTCOME
OUTCOME := [start:StandardDevBL] TEXT [end:StandardDevBL] OUTCOME
OUTCOME := [start:StandardErrorBL] TEXT [end:StandardErrorBL] OUTCOME
OUTCOME := [start:ConfidenceIntervalBL] TEXT [end:ConfidenceIntervalBL] OUTCOME
OUTCOME := [start:ResultMeasuredValue] TEXT [end:ResultMeasuredValue] OUTCOME
OUTCOME := [start:PValueResValue] TEXT [end:PValueResValue] OUTCOME
OUTCOME := [start:StandardDevResValue] TEXT [end:StandardDevResValue] OUTCOME
OUTCOME := [start:StandardErrorResValue] TEXT [end:StandardErrorResValue] OUTCOME
OUTCOME := [start:ConfidenceIntervalResValue] TEXT [end:ConfidenceIntervalResValue]
↪ OUTCOME
OUTCOME := [start:ChangeValue] TEXT [end:ChangeValue] OUTCOME
OUTCOME := [start:PValueChangeVal] TEXT [end:PValueChangeVal] OUTCOME
OUTCOME := [start:StandardDevChangeVal] TEXT [end:StandardDevChangeVal] OUTCOME
OUTCOME := [start:StandardErrorChangeVal] TEXT [end:StandardErrorChangeVal] OUTCOME
OUTCOME := [start:ConfidenceIntervalChangeVal] TEXT [end:ConfidenceIntervalChangeVal]
↪ OUTCOME
OUTCOME := [start:RelativeChangeValue] TEXT [end:RelativeChangeValue] OUTCOME
OUTCOME := [start:NumberAffected] TEXT [end:NumberAffected] OUTCOME
OUTCOME := [start:PercentageAffected] TEXT [end:PercentageAffected] OUTCOME
OUTCOME := [start:PValueNumAffected] TEXT [end:PValueNumAffected] OUTCOME
OUTCOME := [start:StandardDevNumAffected] TEXT [end:StandardDevNumAffected] OUTCOME
OUTCOME := [start:StandardErrorNumAffected] TEXT [end:StandardErrorNumAffected]
↪ OUTCOME
OUTCOME := [start:ConfidenceIntervalNumAffected] TEXT [end:
↪ ConfidenceIntervalNumAffected] OUTCOME
OUTCOME := [start:ObservedResult] TEXT [end:ObservedResult] OUTCOME
OUTCOME := [start:TimeTEXT] TEXT [end:TimeTEXT] OUTCOME
OUTCOME := [start:SubGroupDescription] TEXT [end:SubGroupDescription] OUTCOME
OUTCOME := [end:Outcome]

ENDTEXT_HEAD := [start:EndTEXT] ENDTEXT
ENDTEXT := [start:EndTEXTDescription] TEXT [end:EndTEXTDescription] ENDTEXT
ENDTEXT := [start:AggregationMethod] TEXT [end:AggregationMethod] ENDTEXT
ENDTEXT := [start:BaselineUnit] TEXT [end:BaselineUnit] ENDTEXT
ENDTEXT := [start:MeasurementDevice] TEXT [end:MeasurementDevice] ENDTEXT
ENDTEXT := [end:EndTEXT]

MEDICATION_HEAD := [start:Medication] MEDICATION
MEDICATION := [start:Drug] TEXT [end:Drug] MEDICATION
MEDICATION := [start:DoseValue] TEXT [end:DoseValue] MEDICATION
MEDICATION := [start:DoseUnit] TEXT [end:DoseUnit] MEDICATION
MEDICATION := [start:DeliveryMethod] TEXT [end:DeliveryMethod] MEDICATION
MEDICATION := [start:ApplicationCondition] TEXT [end:ApplicationCondition] MEDICATION
MEDICATION := [end:Medication]
```

Listing 1 Grammar of the data model used for decoding in our experiments

Appendix 2 Slot evaluation

Table 7 Mean test F_1 scores of the best models of each category per slot (mean and standard deviation of 10 training runs)

	Type 2 diabetes F_1		Glaucoma F_1	
Slot name	**Generative**	**Extractive**	**Generative**	**Extractive**
AggregationMethod	0.41 (± 0.07)	0.54 (± 0.03)	0.52 (± 0.08)	0.67 (± 0.13)
AllocationRatio	0.0 (± 0.0)	0.92 (± 0.05)	-	-
analysesHealthCondition	0.84 (± 0.05)	0.73 (± 0.06)	0.87 (± 0.02)	0.86 (± 0.03)
Author	0.97 (± 0.03)	0.92 (± 0.01)	0.8 (± 0.02)	0.94 (± 0.04)
AvgAge	0.0 (± 0.0)	0.37 (± 0.06)	-	-
BaselineUnit	0.42 (± 0.03)	0.44 (± 0.05)	0.54 (± 0.06)	0.56 (± 0.07)
BaselineValue	0.49 (± 0.06)	0.3 (± 0.03)	0.67 (± 0.08)	0.59 (± 0.03)
CTDesign	0.82 (± 0.02)	0.9 (± 0.01)	0.8 (± 0.04)	0.82 (± 0.03)
CTduration	0.89 (± 0.03)	0.89 (± 0.04)	0.78 (± 0.06)	0.87 (± 0.04)
ChangeValue	0.41 (± 0.06)	0.19 (± 0.05)	0.59 (± 0.07)	0.52 (± 0.05)
ConclusionComment	0.73 (± 0.04)	0.88 (± 0.04)	0.84 (± 0.02)	0.91 (± 0.02)
ConfIntervalChangeValue	0.0 (± 0.0)	0.0 (± 0.0)	-	-
ConfIntervalDiff	0.46 (± 0.12)	0.43 (± 0.05)	0.29 (± 0.11)	0.28 (± 0.11)
Country	0.68 (± 0.07)	0.64 (± 0.08)	0.86 (± 0.05)	0.86 (± 0.03)
DeliveryMethod	0.0 (± 0.0)	0.0 (± 0.0)	0.34 (± 0.2)	0.42 (± 0.06)
DiffGroupAbsValue	0.45 (± 0.09)	0.43 (± 0.06)	0.31 (± 0.16)	0.43 (± 0.11)
DoseDescription	0.0 (± 0.0)	0.0 (± 0.0)	-	-
DoseUnit	0.77 (± 0.04)	0.24 (± 0.04)	0.8 (± 0.08)	0.6 (± 0.06)
DoseValue	0.79 (± 0.04)	0.77 (± 0.07)	0.75 (± 0.07)	0.65 (± 0.06)
Drug	0.82 (± 0.05)	0.7 (± 0.02)	0.58 (± 0.04)	0.45 (± 0.06)
Duration	-	-	0.0 (± 0.0)	0.2 (± 0.32)
EndoPointDescription	0.34 (± 0.02)	0.3 (± 0.02)	0.26 (± 0.05)	0.25 (± 0.03)
FinalNumPatientsArm	0.0 (± 0.0)	-	0.0 (± 0.0)	0.02 (± 0.06)
FinalNumberPatientsCT	-	-	0.0 (± 0.0)	0.64 (± 0.13)
Frequency	0.61 (± 0.06)	0.62 (± 0.02)	0.77 (± 0.06)	0.71 (± 0.04)
Journal	0.96 (± 0.05)	0.92 (± 0.05)	0.67 (± 0.08)	0.74 (± 0.07)
MeasurementDevice	-	-	0.0 (± 0.0)	0.2 (± 0.32)
MinAge	0.0 (± 0.0)	0.67 (± 0.17)	-	-
NumberAffected	0.16 (± 0.19)	0.08 (± 0.13)	0.4 (± 0.22)	0.0 (± 0.0)
NumberPatientsArm	0.83 (± 0.09)	0.87 (± 0.02)	0.68 (± 0.12)	0.7 (± 0.06)
NumberPatientsCT	0.65 (± 0.08)	0.93 (± 0.04)	0.65 (± 0.09)	0.86 (± 0.02)
ObjectiveDescription	0.43 (± 0.06)	0.49 (± 0.05)	0.49 (± 0.07)	0.51 (± 0.09)
ObservedResult	0.03 (± 0.03)	0.01 (± 0.01)	0.01 (± 0.03)	0.0 (± 0.0)
PMID	0.97 (± 0.03)	1.0 (± 0.0)	0.98 (± 0.02)	0.99 (± 0.01)
PValueChangeValue	0.1 (± 0.11)	0.0 (± 0.0)	0.0 (± 0.0)	0.33 (± 0.05)

	Type 2 diabetes F_1		Glaucoma F_1	
Slot name	Generative	Extractive	Generative	Extractive
PercentageAffected	0.59 (± 0.08)	0.21 (± 0.03)	0.31 (± 0.18)	0.15 (± 0.03)
Precondition	0.22 (± 0.08)	0.4 (± 0.07)	0.27 (± 0.05)	0.18 (± 0.04)
PublicationYear	0.97 (± 0.03)	1.0 (± 0.0)	0.98 (± 0.02)	1.0 (± 0.0)
PvalueDiff	0.31 (± 0.05)	0.48 (± 0.02)	0.24 (± 0.06)	0.39 (± 0.06)
RelativeChangeValue	0.04 (± 0.09)	0.0 (± 0.0)	0.13 (± 0.2)	0.57 (± 0.12)
RelativeFreqTime	-	-	0.0 (± 0.0)	0.36 (± 0.1)
ResultMeasuredValue	0.27 (± 0.07)	0.21 (± 0.04)	0.57 (± 0.07)	0.35 (± 0.02)
SdDevBL	0.18 (± 0.18)	0.14 (± 0.19)	0.53 (± 0.09)	0.62 (± 0.05)
SdDevChangeValue	0.02 (± 0.06)	0.0 (± 0.0)	0.38 (± 0.14)	0.45 (± 0.07)
SdDevResValue	0.2 (± 0.13)	0.19 (± 0.02)	0.62 (± 0.13)	0.34 (± 0.02)
SdErrorChangeValue	-	-	0.0 (± 0.0)	0.57 (± 0.0)
SubGroupDescription	0.0 (± 0.0)	0.0 (± 0.0)	-	-
TimePoint	0.35 (± 0.11)	0.22 (± 0.03)	0.39 (± 0.07)	0.41 (± 0.03)
Title	0.86 (± 0.05)	0.93 (± 0.02)	0.88 (± 0.06)	0.85 (± 0.03)
Total Micro F_1 Score	0.54 (± 0.03)	0.55 (± 0.01)	0.58 (± 0.02)	0.64 (± 0.01)

Appendix 3 Case study

Table 8 Case study for disease Type 2 Diabetes. Multiple entries for same slot in same template instance separated by |

Slot name	Gold standard	Extractive prediction	Generative prediction
Template Publication			
describes			
Author	Shankar RR | Bao Y | Han P | Hu J | Ma J | Peng Y | Wu F | Xu L | Engel SS | Jia W	Shankar RR | Bao Y | Han P | Hu J | Ma J | Peng Y | Wu F | Xu L | Engel SS | Jia W	Shankar RR | Bao Y | Han P | Hu J | Ma J | Peng Y | Wu F | Xu L | Engel SS | Jia W
Journal	J Diabetes Investig.	J Diabetes Investig.	J Diabetes Investig.
PMID	27740719	27740719	27740719
PublicationYear	2017	2017	2017
Title	Sitagliptin added to stable insulin therapy with or without metformin in Chinese patients with type 2 diabetes.	Sitagliptin added to stable insulin therapy with or without metformin in Chinese patients with type 2 diabetes.	Sitagliptin added to stable insulin therapy with or without metformin in Chinese patients with type 2 diabetes.
Template Intervention			
Frequency	once daily	once daily	once daily
Medication			
Template Population			
Country	USA | Australia	USA	USA
Ethnicity	Chinese		
Precondition	Chinese patients with type 2 diabetes mellitus receiving stable insulin therapy alone or in combination with metformin | patients with inadequate glycemic control on insulin (glycated hemoglobin [HbA1c] < 7. 5 % and < 11 %)	patients with inadequate glycemic control on insulin	patients with inadequate glycemic control on insulin (glycated hemoglobin [HbA1c] <
Template Arm			
AdverseEffect			
Intervention			
Outcome			
Template Endpoint			
BaselineUnit	%	%	
EndoPointDescription	HbA1c	HbA1c of <7. 0 %	adverse events
BaselineUnit		mg / dL | mg / dL	
EndoPointDescription	HbA1c of < 7. 0 %	fasting plasma glucose | 2 - h post - meal glucose | adverse events of hypoglycemia	hypoglycemia
BaselineUnit	mg / dL		%
EndoPointDescription	2 - h post - meal glucose		HbA1c
BaselineUnit	mg / dL		
EndoPointDescription	fasting plasma glucose		HbA1c of <
BaselineUnit			mg / dL
EndoPointDescription	hypoglycemia (symptomatic or asymptomatic)		2 - h post - meal glucose
BaselineUnit			
EndoPointDescription	bodyweight		fasting plasma glucose
BaselineUnit			
EndoPointDescription			adverse events
BaselineUnit			
EndoPointDescription			hypoglycemia
BaselineUnit			%
EndoPointDescription			HbA1c

Slot name	Gold standard	Extractive prediction	Generative prediction
BaselineUnit			
EndoPointDescription			HbA1c of <
BaselineUnit			mg / dL
EndoPointDescription			2 - h post - meal glucose
BaselineUnit			
EndoPointDescription			fasting plasma glucose
Template ClinicalTrial			
analysesHealthCondition	type 2 diabetes mellitus	type 2 diabetes	type 2 diabetes mellitus
AllocationRatio	1 : 1	1 : 1	
Arm			
CTDesign	randomized	randomized	randomized
CTduration	24 weeks	24 weeks	24 weeks
ConclusionComment	After 24 weeks, sitagliptin added to stable insulin therapy (< metformin) was generally well tolerated and improved glycemic control in Chinese patients with type 2 diabetes mellitus.	After 24 weeks, sitagliptin added to stable insulin therapy (<metformin) was generally well tolerated and improved glycemic control in Chinese patients with type 2 diabetes mellitus.	After 24 weeks, sitagliptin added to stable insulin therapy (< |) was generally well tolerated and improved glycemic control in Chinese patients with type 2 diabetes mellitus.
DiffBetweenGroups			
NumberPatientsCT	467	467	467
ObjectiveDescription	We evaluated the tolerability and efficacy of the addition of sitagliptin in Chinese patients with type 2 diabetes mellitus receiving stable insulin therapy alone or in combination with metformin.	We evaluated the tolerability and efficacy of the addition of sitagliptin in Chinese patients with type 2 diabetes mellitus receiving stable insulin therapy alone or in combination with metformin.	We evaluated the tolerability and efficacy of the addition of sitagliptin in Chinese patients with type 2 diabetes mellitus receiving stable insulin therapy alone or in combination with metformin.
Population			
Template Medication			
DoseUnit	mg		mg
DoseValue	100	100	100
Drug	sitagliptin	sitagliptin	sitagliptin
DoseUnit		mg	
DoseValue			
Drug	placebo	placebo	placebo
Template Outcome			
ChangeValue	0. 7	0. 7	
Endpoint			
NumberAffected			64
ObservedResult			
PercentageAffected			27. 4
TimePoint		week 24	
ChangeValue	0. 3	26. 5 | 0. 3 | 14. 4	
Endpoint			
NumberAffected			51
ObservedResult			
PercentageAffected		16 | 21. 9 | 64 | 8 | 27. 4 | 51	21. 9
TimePoint		week 24	
ChangeValue			0. 7
Endpoint			
NumberAffected			
ObservedResult			
PercentageAffected	16		
TimePoint	week 24		week 24
ChangeValue			
Endpoint			
NumberAffected			
ObservedResult			
PercentageAffected	8		16
TimePoint	week 24		week 24
ChangeValue	26. 5		
Endpoint			
NumberAffected			
ObservedResult			
PercentageAffected			
TimePoint			
ChangeValue	14. 4		
Endpoint			
NumberAffected			51
ObservedResult			
PercentageAffected			21. 9
TimePoint			
ChangeValue	10. 7		
Endpoint			
NumberAffected			51
ObservedResult			

Slot name	Gold standard	Extractive prediction	Generative prediction
PercentageAffected			21. 9
TimePoint			
ChangeValue			0. 3
Endpoint			
NumberAffected	64		
ObservedResult			
PercentageAffected	27. 4		
TimePoint			week 24
ChangeValue			
Endpoint			
NumberAffected	51		
ObservedResult			
PercentageAffected	21. 9		8
TimePoint			week 24
ChangeValue			
Endpoint			
NumberAffected			
ObservedResult	Neither group had a significant change from baseline in bodyweight.		
PercentageAffected			
TimePoint			
Template DiffBetweenGroups			
Outcome1			
Outcome2			
PvalueDiff	P < 0. 001	P <0. 001	P = 0. 013
Outcome1			
Outcome2			
PvalueDiff	P = 0. 013	P = 0. 013	P = 0. 013
Outcome1			
Outcome2			
PvalueDiff	P < 0. 001	P <0. 001	

Appendix 4 Journal slot fillers

Table 9 Slot fillers of slot `Journal` with number of occurrences in the type 2 diabetes dataset and glaucoma training and test datasets

Type 2 diabetes	#	Glaucoma	#
‘Ann Intern Med .’	2	‘Acta Ophthalmol .’	1
‘Arch Intern Med .’	1	‘Acta Ophthalmol Scand .’	3
‘Arch Med Res .’	1	‘Acta Ophthalmol Scand’	2
‘BMC Endocr Disord .’	1	‘Acta Ophthalmol’	1
‘Cardiovasc Diabetol .’	1	‘Adv Ther .’	2
‘Clin Drug Investig .’	1	‘Am J Ophthalmol .’	4
‘Clin Ther .’	4	‘Am J Ophthalmol’	2
‘Curr Med Res Opin .’	6	‘Arch Ophthalmol .’	3
‘Diabet Med .’	6	‘BMC Ophthalmol .’	2
‘Diabetes Care .’	2	‘Br J Ophthalmol .’	5
‘Diabetes Care’	12	‘Br J Ophthalmol’	1
‘Diabetes Metab Res Rev .’	2	‘Clin Ophthalmol’	1
‘Diabetes Obes Metab .’	19	‘Clin Ther .’	4
‘Diabetes Res Clin Pract .’	1	‘Clin Ther’	1
‘Diabetes Technol Ther .’	2	‘Coll Antropol’	1
‘Diabetes’	1	‘Curr Med Res Opin .’	5
‘Diabetologia’	3	‘Curr Med Res Opin’	1
‘Endocr J .’	1	‘Curr Ther Res Clin Exp .’	1
‘Exp Clin Endocrinol Diabetes’	1	‘Eur J Ophthalmol .’	2
‘Horm Metab Res .’	3	‘Eur J Ophthalmol’	1
‘Horm Res .’	1	‘Eye (Lond) .’	6
‘J Diabetes Complications’	2	‘Eye (Lond)’	1
‘J Diabetes Investig .’	1	‘Graefes Arch Clin Exp Ophthalmol .’	4
‘J Diabetes’	3	‘Int Ophthalmol .’	1
‘Lancet’	2	‘Invest Ophthalmol Vis Sci .’	1
‘Med J Malaysia’	1	‘J Fr Ophtalmol .’	1
‘Obesity (Silver Spring)’	1	‘J Glaucoma’	9
‘Open Access Maced J Med Sci .’	1	‘J Ocul Pharmacol .’	1
‘PLoS One’	1	‘J Ocul Pharmacol Ther .’	5
‘Pediatr Diabetes’	1	‘J Ocul Pharmacol Ther’	2
‘Pharmacotherapy’	1	‘Jpn J Ophthalmol’	2
‘Srp Arh Celok Lek .’	1	‘Lancet’	1
		‘Nippon Ganka Gakkai Zasshi .’	1
		‘Ophthalmology’	8
		‘Saudi Med J’	1
		‘Surv Ophthalmol .’	1

Appendix 5 Template cardinalities

Table 10 Cardinality Evaluation Type 2 Diabetes Generative

Template name	Mean GT count	Mean predicted count	Abs diff
Arm	2.00	1.99 (± 0.02)	0.01 (± 0.02)
DiffBetweenGroups	2.75	3.78 (± 0.95)	1.01 (± 0.96)
Endpoint	5.85	11.16 (± 1.11)	5.32 (± 1.11)
EvidenceQuality	1.00	0.0 (± 0.0)	1.0 (± 0.0)
Intervention	2.10	2.0 (± 0.04)	0.1 (± 0.04)
Medication	2.20	1.99 (± 0.07)	0.21 (± 0.07)
Outcome	10.35	11.16 (± 1.11)	1.08 (± 0.82)

Table 11 Cardinality Evaluation Type 2 Diabetes Extractive

Template name	Mean GT count	Mean predicted count	Abs diff
Arm	2.00	0.91 (± 0.03)	1.09 (± 0.03)
DiffBetweenGroups	2.75	2.06 (± 0.08)	0.69 (± 0.08)
Endpoint	5.85	2.02 (± 0.03)	3.83 (± 0.03)
EvidenceQuality	1.00	0.0 (± 0.0)	1.0 (± 0.0)
Intervention	2.10	0.88 (± 0.03)	1.22 (± 0.03)
Medication	2.20	2.02 (± 0.1)	0.18 (± 0.1)
Outcome	10.35	1.99 (± 0.07)	8.36 (± 0.07)

Table 12 Cardinality Evaluation Glaucoma Generative

Template name	Mean GT count	Mean predicted count	Abs diff
Arm	2.00	1.98 (± 0.04)	0.02 (± 0.04)
DiffBetweenGroups	1.62	1.99 (± 0.38)	0.39 (± 0.36)
Endpoint	2.48	5.39 (± 1.09)	2.91 (± 1.09)
EvidenceQuality	1.00	0.0 (± 0.0)	1.0 (± 0.0)
Intervention	2.19	2.0 (± 0.05)	0.19 (± 0.05)
Medication	2.33	2.31 (± 0.14)	0.11 (± 0.08)
Outcome	5.05	5.39 (± 1.09)	0.94 (± 0.57)

Table 13 Cardinality Evaluation Glaucoma Extractive

Template name	Mean GT count	Mean predicted count	Abs diff
Arm	2.00	0.67 (± 0.05)	1.33 (± 0.05)
DiffBetweenGroups	1.70	1.85 (± 0.16)	0.17 (± 0.13)
Endpoint	2.48	2.13 (± 0.22)	0.35 (± 0.22)
EvidenceQuality	1.00	0.0 (± 0.0)	1.0 (± 0.0)
Intervention	2.19	1.39 (± 0.1)	0.8 (± 0.1)
Medication	2.33	1.86 (± 0.2)	0.47 (± 0.2)
Outcome	5.05	2.07 (± 0.04)	2.98 (± 0.04)

Appendix 6 Linearized publication

```
[start:Publication] [start:describes] [start:ClinicalTrial] [start:analysesHealthCondition]
 ocular hy pert ension [end:analysesHealthCondition] [start:analysesHealthCondition] open
angle gl au com a [end:analysesHealthCondition] [start:hasArm] [start:Arm] [start:
hasAdverseEffect] [start:Outcome] [start:hasEndpoint] [start:Endpoint] [start:
hasEndoPointDescription] vert igo [end:hasEndoPointDescription] [end:Endpoint] [end:
hasEndpoint] [start:hasNumberAffected] one [end:hasNumberAffected] [end:Outcome] [end:
hasAdverseEffect] [start:hasAdverseEffect] [start:Outcome] [start:hasEndpoint] [start:
Endpoint] [start:hasEndoPointDescription] super f icial p unct ate ker at itis [end:
hasEndoPointDescription] [end:Endpoint] [end:hasEndpoint] [start:hasNumberAffected] one [
end:hasNumberAffected] [end:Outcome] [end:hasAdverseEffect] [start:hasFinalNumPatientsArm]
74 [end:hasFinalNumPatientsArm] [start:hasIntervention] [start:Intervention] [start:
hasFrequency] once daily [end:hasFrequency] [start:hasMedication] [start:Medication] [start
:hasDoseUnit] % [end:hasDoseUnit] [start:hasDoseValue] 1 [end:hasDoseValue] [start:hasDrug]
 cart e ol ol al gin ate [end:hasDrug] [end:Medication] [end:hasMedication] [start:
hasRelativeFreqTime] AM [end:hasRelativeFreqTime] [end:Intervention] [end:hasIntervention]
[start:hasIntervention] [start:Intervention] [start:hasMedication] [start:Medication] [
start:hasDrug] place bo [end:hasDrug] [end:Medication] [end:hasMedication] [start:
hasRelativeFreqTime] even ing [end:hasRelativeFreqTime] [end:Intervention]
[end:hasIntervention] [start:hasOutcome] [start:Outcome] [start:hasChangeValue] 5 . 67 [end
:hasChangeValue] [start:hasEndpoint] [start:Endpoint] [start:hasAggregationMethod] mean [
end:hasAggregationMethod] [start:hasBaselineUnit] mm H g [end:hasBaselineUnit] [start:
hasEndoPointDescription] t rough [end:hasEndoPointDescription] [start:
hasEndoPointDescription] peak [end:hasEndoPointDescription] [end:Endpoint] [end:hasEndpoint
] [start:hasPValueChangeValue] p < 0 . 005 [end:hasPValueChangeValue] [start:
hasSdDevChangeValue] 3 . 30 [end:hasSdDevChangeValue] [start:hasTimePoint] day 60 [end:
hasTimePoint] [end:Outcome] [end:hasOutcome] [start:hasOutcome] [start:Outcome] [start:
hasChangeValue] 6 . 70 [end:hasChangeValue] [start:hasEndpoint] [start:Endpoint] [start:
hasAggregationMethod] mean [end:hasAggregationMethod] [start:hasBaselineUnit] mm H g [end:
hasBaselineUnit] [end:Endpoint] [end:hasEndpoint] [start:hasPValueChangeValue] p < 0 . 005
[end:hasPValueChangeValue] [start:hasSdDevChangeValue] 2 . 81 [end:hasSdDevChangeValue] [
start:hasTimePoint] 11 . 00 [end:hasTimePoint] [end:Outcome] [end:hasOutcome] [end:Arm] [
end:hasArm] [start:hasArm] [start:Arm] [start:hasAdverseEffect] [start:Outcome] [start:
hasEndpoint] [start:Endpoint] [start:hasEndoPointDescription] dec re ase in blood pressure
[end:hasEndoPointDescription] [end:Endpoint] [end:hasEndpoint] [start:hasNumberAffected]
one [end:hasNumberAffected] [end:Outcome] [end:hasAdverseEffect] [start:
hasFinalNumPatientsArm] 75 [end:hasFinalNumPatientsArm] [start:hasIntervention] [start:
Intervention] [start:hasFrequency] tw ice daily [end:hasFrequency] [start:hasMedication] [
start:Medication] [start:hasDoseUnit] % [end:hasDoseUnit] [start:hasDoseValue] 1 [end:
hasDoseValue] [start:hasDrug] cart e ol ol [end:hasDrug] [end:Medication] [end:
hasMedication] [end:Intervention] [end:hasIntervention] [start:hasOutcome] [start:Outcome]
[start:hasChangeValue] 6 . 32 [end:hasChangeValue] [start:hasEndpoint] [start:Endpoint] [
start:hasAggregationMethod] mean [end:hasAggregationMethod] [start:hasBaselineUnit] mm H g
[end:hasBaselineUnit] [start:hasEndoPointDescription] t rough [end:hasEndoPointDescription]
 [start:hasEndoPointDescription] peak [end:hasEndoPointDescription] [end:Endpoint] [end:
hasEndpoint] [start:hasPValueChangeValue] p < 0 . 005 [end:hasPValueChangeValue] [start:
hasSdDevChangeValue] 2 . 87 [end:hasSdDevChangeValue] [start:hasTimePoint] day 60 [end:
hasTimePoint] [end:Outcome] [end:hasOutcome] [start:hasOutcome] [start:Outcome] [start:
hasChangeValue] 6 . 55 [end:hasChangeValue] [start:hasEndpoint] [start:Endpoint] [start:
hasAggregationMethod] mean [end:hasAggregationMethod] [start:hasBaselineUnit] mm H g [end:
hasBaselineUnit] [end:Endpoint] [end:hasEndpoint] [start:hasPValueChangeValue] p < 0 . 005
[end:hasPValueChangeValue] [start:hasSdDevChangeValue] 3 . 35 [end:hasSdDevChangeValue] [
start:hasTimePoint] 11 . 00 [end:hasTimePoint] [end:Outcome] [end:hasOutcome] [end:Arm] [
end:hasArm] [start:hasCTDesign] double − mask ed [end:hasCTDesign] [start:hasCTDesign]
random ized [end:hasCTDesign] [start:hasCTDesign] par allel group [end:hasCTDesign] [start:
hasCTDesign] mask ing [end:hasCTDesign] [start:hasCTduration] 2 months [end:hasCTduration]
[start:hasConclusionComment] The new al gin ate form ulation of long − acting cart e ol ol
1 % given once daily is as effective as standard 1 % cart e ol ol given tw ice daily , with
 no meaning ful diff erences reg arding safety . [end:hasConclusionComment] [start:
hasConclusionComment] This effic acy w asy was verified at 9 AM ( 24 hours after the last
drop of long − acting cart e ol ol or 12 hours after that of standard cart e ol ol ) and at
 11 AM ( 2 hours after the morning drop ) . [end:hasConclusionComment] [start:
hasConclusionComment] The new al gin ate form ulation of long − acting cart e ol ol 1 %
given once a day is effective and well t oler ated by gl au com a pat ients who require ch
ronic treatment . [end:hasConclusionComment] [start:hasFinalNumberPatientsCT] 149 [end:
hasFinalNumberPatientsCT] [start:hasNumberPatientsCT] 151 [end:hasNumberPatientsCT] [start:
hasNumberPatientsCT] 151 [end:hasNumberPatientsCT] [start:hasObjectiveDescription] The
object ive of this study was to evaluate the effic acy and safety of long − acting 1 % cart
 e ol ol al gin ate s olution comp ared to standard 1 % cart e ol ol s olution . [end:
hasObjectiveDescription] [start:hasPopulation] [start:Population] [start:hasCountry] France
 [end:hasCountry] [start:hasPrecondition] Pat ients with ocular hy pert ension or open
angle gl au com a [end:hasPrecondition] [start:hasPrecondition] Pat ients using ocular hyp
ot ensive med ication were required to under go a wash out . [end:hasPrecondition] [end:
Population] [end:hasPopulation] [end:ClinicalTrial] [end:describes] [start:hasAuthor] Tr
inqu and C [end:hasAuthor] [start:hasAuthor] Roman et JP [end:hasAuthor] [start:hasAuthor]
N ord mann JP [end:hasAuthor] [start:hasAuthor] All aire C [end:hasAuthor] [start:
hasJournal] J Fr O ph t alm ol . [end:hasJournal] [start:hasPMID] 126 60 585 [end:hasPMID]
[start:hasPublicationYear] 2003 [end:hasPublicationYear] [start:hasTitle] E ffic acy and
safety of long − acting cart e ol ol 1 % once daily . [end:hasTitle] [end:Publication]
```

Listing 2 Linearization of Glaucoma Publication Template Instance

Abbreviations

BERT	Bidirectional encoder representations from transformers
C-TrO	Clinical trial ontology; CFG: context-free grammar
CNN	Convolutional neural network
CRF	Conditional random field
EBM	Evidence-based medicine
Flan	Finetuning language models
GenIE	Generative information extraction
GT	Ground truth
HAC	Hierarchical agglomerative clustering
IE	Information extraction
iff	if and only if
ITC	Intra-template compatibility
LED	Longformer-encoder-decoder
LSTM	Long short-term memory network
MAD	Mean absolute deviation
MAP	Maximum a posteriori probability
PICO	Patient, intervention, comparison, outcomes
RCT	Randomized controlled trial
REBEL	Relation extraction by end-to-end language generation
ReLU	Rectified linear unit
SF	Slot-filler
T5	Text-to-text transfer transformer
TI	Template instance

Acknowledgements
Not applicable.

Authors' contributions
CW contributed to the introduction and model section as well as worked on both approaches and their implementation. DS finalized many parts of the implementation of both the extractive and the generative approach and ran the experiments as well as adapted the original draft to the current state of the code. Moreover, DS was the main author of the experimental results, discussion and related work section and decoding subsection. CW and DS equally contributed to this paper. PC supervised all of the above steps. All authors read and approved the final manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. The research of David M. Schmidt is funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia under the grant no NW21-059A (SAIL). Christian Witte has been funded by a grant from the Federal Ministry of Health (BMG) as part of the KINBIOTICS project. Philipp Cimiano acknowledges funding from the Transregio 318 "Constructing Explainability" (Projects B1 and C5). We acknowledge the financial support of the German Research Foundation (DFG) and the Open Access Publication Fund of Bielefeld University for the article processing charge.

Availability of data and materials
The code and datasets generated and/or analysed during the current study are available in the Zenodo repository, https://doi.org/10.5281/zenodo.10419786 [66].

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Author details
[1]Semantic Computing Group, Center for Cognitive Interaction Technology, Bielefeld University, Inspiration 1, Bielefeld 33619, NRW, Germany.

Received: 5 January 2024 Accepted: 5 April 2024
Published online: 23 April 2024

References

1. Bastian H, Glasziou P, Chalmers I. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? PLoS Med. 2010;7(9): e1000326. https://doi.org/10.1371/journal.pmed.1000326.
2. Sackett DL, Rosenberg WM, Gray JM, Haynes RB, Richardson WS. Evidence based medicine. BMJ. 1996;313(7050):170.
3. Shekelle PG, Ortiz E, Rhodes S, Morton SC, Eccles MP, Grimshaw JM, et al. Validity of the Agency for Healthcare Research and Quality clinical practice guidelines: how quickly do guidelines become outdated? JAMA. 2001;286(12):1461–7.
4. Shojania KG, Sampson M, Ansari MT, Ji J, Doucette S, Moher D. How quickly do systematic reviews go out of date? A survival analysis Ann Intern Med. 2007;147(4):224–33.
5. Beller EM, Chen JKH, Wang ULH, Glasziou PP. Are systematic reviews up-to-date at the time of publication? Syst Rev. 2013;2:36. https://doi.org/10.1186/2046-4053-2-36.
6. Koch G. No improvement–still less than half of the Cochrane reviews are up to date. In: XIV Cochrane Colloquium. Dublin; 2006.
7. Tsafnat G, Glasziou P, Choong MK, et al. Systematic review automation technologies Syst Rev. 2014;3:74. https://doi.org/10.1186/2046-4053-3-74.
8. Beller E, Clark J, Tsafnat G, et al. Making progress with the automation of systematic reviews: principles of the International Collaboration for the Automation of Systematic Reviews (ICASR). Syst Rev. 2018;7:77. https://doi.org/10.1186/s13643-018-0740-7.
9. O'Connor AM, Tsafnat G, Gilbert SB, Thayer KA, Shemilt I, Thomas J, et al. Still moving toward automation of the systematic review process: a summary of discussions at the third meeting of the International Collaboration for Automation of Systematic Reviews (ICASR). Syst Rev. 2019;8:57. https://doi.org/10.1186/s13643-019-0975-y.
10. Sanchez-Graillet O, Witte C, Grimm F, Grautoff S, Ell B, Cimiano P. Synthesizing evidence from clinical trials with dynamic interactive argument trees. J Biomed Semant. 2022;13(1):16. https://doi.org/10.1186/s13326-022-00270-8.
11. Boudin F, Nie JY, Bartlett JC, Grad R, Pluye P, Dawes M. Combining classifiers for robust PICO element detection. BMC Med Inform Decis Mak. 2010;10(1):1–6.
12. Jin D, Szolovits P. Pico. Element detection in medical text via long short-term memory neural networks. In: Proceedings of the BioNLP 2018 workshop. Melbourne: Association for Computational Linguistics; 2018. p. 67–75. https://aclanthology.org/W18-2308. https://doi.org/10.18653/v1/W18-2308.
13. Trenta A, Hunter A, Riedel S. Extraction of evidence tables from abstracts of randomized clinical trials using a maximum entropy classifier and global constraints. 2015. arXiv preprint arXiv:1509.05209.
14. Witte C, Cimiano P. Intra-Template Entity Compatibility based Slot-Filling for Clinical Trial Information Extraction. In: Proceedings of the 21st Workshop on Biomedical Language Processing. Dublin: Association for Computational Linguistics; 2022. p. 178–192. https://aclanthology.org/2022.bionlp-1.18. https://doi.org/10.18653/v1/2022.bionlp-1.18.
15. Sanchez-Graillet O, Cimiano P, Witte C, Ell B. C-TrO: An Ontology for Summarization and Aggregation of the Level of Evidence in Clinical Trials. In: Proc. of the 5th Joint Ontology Workshops (JOWO): Ontologies and Data in the Life Sciences. 2019. https://ceur-ws.org/Vol-2518/paper-ODLS7.pdf.
16. Beltagy I, Peters ME, Cohan A. Longformer: The long-document transformer. 2020. arXiv preprint arXiv:2004.05150.
17. Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, et al. Scaling Instruction-Finetuned Language Models. CoRR. 2022. https://doi.org/10.48550/ARXIV.2210.11416. arXiv:2210.11416
18. Cabot PLH, Navigli R. REBEL: Relation extraction by end-to-end language generation. In: Findings of the Association for Computational Linguistics: EMNLP 2021. Punta Cana: Association for Computational Linguistics; 2021. p. 2370–2381. https://aclanthology.org/2021.findings-emnlp.204.https://doi.org/10.18653/v1/2021.findings-emnlp.204.
19. Josifoski M, De Cao N, Peyrard M, West R. GenIE: generative information extraction. 2021. arXiv preprint arXiv:2112.08340.

20. Lu Y, Lin H, Xu J, Han X, Tang J, Li A, et al. Text2Event: Controllable Sequence-to-Structure Generation for End-to-end Event Extraction. CoRR. 2021. arXiv:2106.09232.
21. Sanchez-Graillet O, Witte C, Grimm F, Cimiano P. An annotated corpus of clinical trial publications supporting schema-based relational information extraction. J Biomed Semant. 2021. Under Review.
22. Hsu I, Huang K, Boschee E, Miller S, Natarajan P, Chang K, et al. Event Extraction as Natural Language Generation. CoRR. 2021. arXiv:2108.12724.
23. Yang H, Sui D, Chen Y, Liu K, Zhao J, Wang T. Document-Level Event Extraction via Parallel Prediction Networks. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics; 2021. p. 6298–6308. https://doi.org/10.18653/v1/2021.acl-long.492.
24. Giorgi J, Bader GD, Wang B. A sequence-to-sequence approach for document-level relation extraction. BioNLP 2022@ ACL 2022. Dublin: Association for Computational Linguistics; 2022. p. 10–25. https://aclanthology.org/2022.bionlp-1.2. https://doi.org/10.18653/v1/2022.bionlp-1.2.
25. Du X, Rush A, Cardie C. GRIT: Generative Role-filler Transformers for Document-level Event Entity Extraction. In: Merlo P, Tiedemann J, Tsarfaty R, editors. Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. Online: Association for Computational Linguistics; 2021. p. 634–644. https://doi.org/10.18653/v1/2021.eacl-main.52. https://aclanthology.org/2021.eacl-main.52.
26. Du X, Rush A, Cardie C. Template Filling with Generative Transformers. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Online: Association for Computational Linguistics; 2021. p. 909–914. https://doi.org/10.18653/v1/2021.naacl-main.70.
27. Wang XD, Weber L, Leser U. Biomedical Event Extraction as Multi-turn Question Answering. In: Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis. Online: Association for Computational Linguistics; 2020. p. 88–96. https://doi.org/10.18653/v1/2020.louhi-1.10.
28. Ramponi A, Van Der Goot R, Lombardo R, Plank B. Biomedical Event Extraction as Sequence Labeling. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics; 2020. p. 5357–5367. https://doi.org/10.18653/v1/2020.emnlp-main.431.
29. Zhu L, Zheng H. Biomedical Event Extraction with a Novel Combination Strategy Based on Hybrid Deep Neural Networks. BMC Bioinformatics. 2020;21(1):47. https://doi.org/10.1186/s12859-020-3376-2.
30. Huang KH, Yang M, Peng N. Biomedical Event Extraction with Hierarchical Knowledge Graphs. In: Cohn T, He Y, Liu Y, editors. Findings of the Association for Computational Linguistics: EMNLP 2020. Online: Association for Computational Linguistics; 2020. p. 1277–1285. https://doi.org/10.18653/v1/2020.findings-emnlp.114. https://aclanthology.org/2020.findings-emnlp.114.
31. Trieu HL, Tran TT, Duong KNA, Nguyen A, Miwa M, Ananiadou S. DeepEventMine: End-to-End Neural Nested Event Extraction from Biomedical Texts. Bioinformatics. 2020;36(19):4910–7. https://doi.org/10.1093/bioinformatics/btaa540.
32. Jiang Y, Kavuluru R. End-to-End *n*-ary Relation Extraction for Combination Drug Therapies. 2023. https://doi.org/10.48550/arXiv.2303.16886. arXiv:2303.16886.
33. Kim Y, Meystre SM. Ensemble Method-Based Extraction of Medication and Related Information from Clinical Texts. J Am Med Inform Assoc. 2020;27(1):31–8. https://doi.org/10.1093/jamia/ocz100.
34. Stylianou N, Kosmoliaptsis P, Vlahavas I. Improved Biomedical Entity Recognition via Longer Context Modeling. In: Maglogiannis I, Macintyre J, Iliadis L, editors. Artificial Intelligence Applications and Innovations. vol. 627. Cham: Springer International Publishing; 2021. p. 45–56. https://doi.org/10.1007/978-3-030-79150-6_4.
35. Farnsworth S, Gurdin G, Vargas J, Mulyar A, Lewinski N, McInnes BT. Extracting Experimental Parameter Entities from Scientific Articles. J Biomed Inform. 2022Feb;126: 103970. https://doi.org/10.1016/j.jbi.2021.103970.
36. Tseo Y, Salkola MI, Mohamed A, Kumar A, Abnousi F. Information Extraction of Clinical Trial Eligibility Criteria. 2020. https://doi.org/10.48550/arXiv.2006.07296. arXiv:2006.07296.
37. Abaho M, Bollegala D, Williamson PR, Dodd S. Assessment of contextualised representations in detecting outcome phrases in clinical trials. CoRR. 2022. https://doi.org/10.48550/ARXIV.2203.03547. arXiv:2203.03547.
38. Abaho M, Bollegala D, Williamson P, Dodd S. Position-based Prompting for Health Outcome Generation. In: Demner-Fushman D, Cohen KB, Ananiadou S, Tsujii J, editors. Proceedings of the 21st Workshop on Biomedical Language Processing. Dublin: Association for Computational Linguistics; 2022. p. 26–36. https://doi.org/10.18653/v1/2022.bionlp-1.3. https://aclanthology.org/2022.bionlp-1.3.
39. Abaho M, Bollegala D, Williamson P, Dodd S. Detect and Classify – Joint Span Detection and Classification for Health Outcomes. In: Moens MF, Huang X, Specia L, Yih SWt, editors. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Online and Punta Cana: Association for Computational Linguistics; 2021. p. 8709–8721. https://doi.org/10.18653/v1/2021.emnlp-main.686. https://aclanthology.org/2021.emnlp-main.686.
40. Ganguly D, Gleize M, Hou Y, Jochim C, Bonin F, Pascale A, et al. Outcome Prediction from Behaviour Change Intervention Evaluations using a Combination of Node and Word Embedding. AMIA Ann Symp Proc. 2021;2021:486–95. Published online 2022 Feb 21.
41. Papanikolaou Y, Staib M, Grace JJ, Bennett F. Slot Filling for Biomedical Information Extraction. In: Demner-Fushman D, Cohen KB, Ananiadou S, Tsujii J, editors. Proceedings of the 21st Workshop on Biomedical Language Processing, BioNLP@ACL 2022, Dublin, Ireland, May 26, 2022. Association for Computational Linguistics; 2022. p. 82–90. https://doi.org/10.18653/v1/2022.bionlp-1.7.
42. Dhrangadhariya A, Müller H. Not so Weak PICO: Leveraging Weak Supervision for Participants, Interventions, and Outcomes Recognition for Systematic Review Automation. JAMIA Open. 2023;6(1):ooac107. https://doi.org/10.1093/jamiaopen/ooac107.
43. Nye BE, DeYoung J, Lehman E, Nenkova A, Marshall IJ, Wallace BC. Understanding Clinical Trial Reports: Extracting Medical Entities and Their Relations. CoRR. 2020. arXiv:2010.03550.
44. Wallace BC, Kuiper J, Sharma A, Zhu MB, Marshall IJ. Extracting PICO Sentences from Clinical Trial Reports using Supervised Distant Supervision. J Mach Learn Res. 2016;17:132:1–25. http://jmlr.org/papers/v17/15-404.html.
45. Liu S, Sun Y, Li B, Wang W, Bourgeois FT, Dunn AG. Sent2Span: Span Detection for PICO Extraction in the Biomedical Text without Span Annotations. In: Moens MF, Huang X, Specia L, Yih SWt, editors. Findings of the Association for Computational Linguistics: EMNLP 2021. Punta Cana: Association for Computational Linguistics; 2021. p. 1705–1715. https://doi.org/10.18653/v1/2021.findings-emnlp.147. https://aclanthology.org/2021.findings-emnlp.147.
46. Jin D, Szolovits P. Advancing PICO element detection in biomedical text via deep neural networks. Bioinform. 2020;36(12):3856–62. https://doi.org/10.1093/bioinformatics/btaa256.
47. Kang T, Zou S, Weng C. Pretraining to Recognize PICO Elements from Randomized Controlled Trial Literature. In: Ohno-Machado L, Séroussi B, editors. MEDINFO 2019: Health and Wellbeing e-Networks for All - Proceedings of the 17th World Congress on Medical and Health Informatics, Lyon, France, 25-30 August 2019. vol. 264 of Studies in Health Technology and Informatics. IOS Press; 2019. p. 188–192. https://doi.org/10.3233/SHTI190209.
48. Chabou S, Iglewski M. Combination of Conditional Random Field with a Rule Based Method in the Extraction of PICO Elements. BMC Med Inform Decis Mak. 2018;18(1):128. https://doi.org/10.1186/s12911-018-0699-2.
49. Yuan X, Xiaoli L, Shilei L, Qinwen S, Ke L. Extracting PICO Elements From RCT Abstracts Using 1-2gram Analysis And Multitask Classification. In: Proceedings of the Third International Conference on Medical and Health Informatics 2019 - ICMHI 2019. Xiamen: ACM Press; 2019. p. 194–199. https://doi.org/10.1145/3340037.3340043.
50. Stylianou N, Razis G, Goulis DG, Vlahavas I. EBM+: Advancing Evidence-Based Medicine via Two Level Automatic Identification of Populations, Interventions, Outcomes in Medical Literature. Artif Intell Med. 2020;108: 101949. https://doi.org/10.1016/j.artmed.2020.101949.
51. Jin D, Szolovits P. PICO Element Detection in Medical Text via Long Short-Term Memory Neural Networks. In: Proceedings of the BioNLP 2018 Workshop. Melbourne: Association for Computational Linguistics; 2018. p. 67–75. https://doi.org/10.18653/v1/W18-2308.
52. Afzal M, Alam F, Malik KM, Malik GM. Clinical Context-Aware Biomedical Text Summarization Using Deep Neural Network: Model Development

and Validation. J Med Internet Res. 2020;22(10): e19810. https://doi.org/10.2196/19810.

53. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT. Minneapolis: Association for Computational Linguistics; 2019. p. 4171–4186. https://aclanthology.org/N19-1423. https://doi.org/10.18653/v1/N19-1423.
54. Schmidt L, Weeds J, Higgins JPT. Data Mining in Clinical Trial Text: Transformers for Classification and Question Answering Tasks. In: Cabitza F, Fred ALN, Gamboa H, editors. Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020) - Volume 5: HEALTHINF, Valletta, Malta, February 24-26, 2020. SCITEPRESS. 2020. p. 83–94. https://doi.org/10.5220/0008945700830094.
55. Zhang T, Yu Y, Mei J, Tang Z, Zhang X, Li S. Unlocking the Power of Deep PICO Extraction: Step-wise Medical NER Identification. CoRR. 2020. arXiv:2005.06601.
56. Whitton J, Hunter A. Automated tabulation of clinical trial results: A joint entity and relation extraction approach with transformer-based language representations. Artif Intell Med. 2023;144:102661. https://doi.org/10.1016/j.artmed.2023.102661.
57. Dhrangadhariya A, Aguilar G, Solorio T, Hilfiker R, Müller H. End-to-End Fine-Grained Neural Entity Recognition of Patients, Interventions, Outcomes. In: Candan KS, Ionescu B, Goeuriot L, Larsen B, Müller H, Joly A, et al., editors. Experimental IR Meets Multilinguality, Multimodality, and Interaction. vol. 12880. Cham: Springer International Publishing; 2021. p. 65–77. https://doi.org/10.1007/978-3-030-85251-1_6.
58. Text chunking using transformation-based learning. In: Natural language processing using very large corpora. Springer. p. 157–176.
59. Agarap AF. Deep learning using rectified linear units (relu). 2018. arXiv preprint arXiv:1803.08375.
60. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017;30:5998-6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.
61. Chen MX, Firat O, Bapna A, Johnson M, Macherey W, Foster G, et al. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne: Association for Computational Linguistics; 2018. p. 76–86. https://aclanthology.org/P18-1008. https://doi.org/10.18653/v1/P18-1008.
62. Shi T, Keneshloo Y, Ramakrishnan N, Reddy CK. Neural abstractive text summarization with sequence-to-sequence models. ACM Trans Data Sci. 2021;2(1):1–37.
63. Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A Next-generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery; 2019. https://doi.org/10.1145/3292500.3330701.
64. Shankar RR, Bao Y, Han P, Hu J, Ma J, Peng Y, et al. Sitagliptin added to stable insulin therapy with or without metformin in Chinese patients with type 2 diabetes. J Diabetes Investig. 2017;8(3):321–9.
65. Thomas J, Noel-Storr A, Marshall I, et al. Living systematic reviews: 2. Combining human and machine effort. J Clin Epidemiol. 2017;91:31–7. https://doi.org/10.1016/j.jclinepi.2017.08.011.
66. Schmidt DM, Witte C, Cimiano P. ag-sc/Clinical-Trial-Information-Extraction: Initial release. Zenodo; 2023. https://doi.org/10.5281/zenodo.10419786.

Publisher's Note