PICO Element Detection in Medical Text via Long Short-Term Memory Neural Networks

Di Jin
MIT
jindi15@mit.edu

Peter Szolovits
MIT
psz@mit.edu

Abstract

Successful evidence-based medicine (EBM) applications rely on answering clinical questions by analyzing large medical literature databases. In order to formulate a well-defined, focused clinical question, a framework called PICO is widely used, which identifies the sentences in a given medical text that belong to the four components: Participants/Problem (P), Intervention (I), Comparison (C) and Outcome (O). In this work, we present a Long Short-Term Memory (LSTM) neural network based model to automatically detect PICO elements. By jointly classifying subsequent sentences in the given text, we achieve state-of-the-art results on PICO element classification compared to several strong baseline models. We also make our curated data public as a benchmarking dataset so that the community can benefit from it.

1 Introduction

The paradigm of evidence-based medicine (EBM) involves the incorporation of current best evidence, such as the reports of randomized controlled trials (RCTs), into decision making for patient care (Sackett, 1997). Such evidence, integrated with the physician's own expertise and patient-specific factors, can lead to better patient outcomes and higher quality health care (Sackett et al., 1996). In practice, successful EBM applications rely on answering clinical questions via analysis of large medical literature databases such as PubMed. And most often, a PICO framework is used to formulate a well-defined, focused clinical question, which decomposes the question into four parts: Participants/Problem (P), Intervention (I), Comparison (C) and Outcome (O) (Richardson et al., 1995).

Typically the analyses that underlie EBM begin by selecting a set of potentially relevant papers, which are then further refined by human judgment to form the evidence base on which the answer to a specific question depends. To facilitate this selection process, it would be advantageous that all papers (or at least their abstracts) can be organized according to the PICO foci. Unfortunately, a significant portion of the medical literature contains either unstructured or sub-optimally structured abstracts, without specifically identified PICO elements. Therefore, we would like to introduce a method to automate the identification of PICO elements in medical abstracts in order to make possible the automated selection of possibly relevant articles for a proposed study.

In this paper, we present a system based on artificial neural networks (ANN) to tackle the issue of extracting PICO elements in medical abstracts as a classification task at the sentence level. Our key contributions are as follows:

1. Previous methods for PICO elements extraction focused on shallow models such as Naive Bayes (NB), Support Vector Machines (SVM) and Conditional Random Fields (CRF), which are limited in modeling capacity. To significantly boost the performance, we propose a Long Short-Term Memory (LSTM) based ANN model to solve this task.

2. Most previous systems detected the PICO elements one by one; thus several classifiers needed to be built and trained separately, which is sub-optimal in efficiency. That approach also cannot take advantage of shared structure among the individual classifiers. In

Proceedings of the BioNLP 2018 workshop, pages 67–75
Melbourne, Australia, July 19, 2018.

this work we extract PICO components simultaneously from any given medical abstract.

3. In all previous works, the only dataset used for training and test and made public is from (Kim et al., 2011). However, this dataset contains only 1000 abstracts, which is not enough for a ANN based deep learning model to obtain good generalization results. Therefore, we curate a dataset comprising of over tens of thousands of abstracts and make it public as a benchmark dataset so that everyone else can use it.

4. Instead of normally treating PICO detection as a single sentence classification problem, we view it as a sequential sentence classification task, where the sequence of sentences in an abstract is jointly predicted. In this way, the information from the context sentences can be used to help predict the current sentence, which does improve the classification accuracy considerably. Leveraging this strategy, we obtain state-of-the-art PICO elements extraction accuracy, significantly outperforming all previous methods.

2 Related Work

In many previous user studies, the generalized use of the PICO framework or similar schema by clinicians has been validated for its performance improvement on searching literature for clinical questions (Schardt et al., 2007; Boudin et al., 2010c; Znaidi et al., 2015). This has greatly fueled academic interest in the development of systems for automatic PICO element detection. Over the last decade, the research progress for this task can be summarized according to three aspects: models for classification, dataset generation, and task formulation.

Many well-known machine learning techniques have been proposed to build stronger models for this task, including Nave Bayes (NB) (Huang et al., 2013; Boudin et al., 2010a; Demner-Fushman and Lin, 2007), Random Forest (RF) (Boudin et al., 2010a), Support Vector Machine (SVM) (Boudin et al., 2010a; Hansen et al., 2008), Conditional Random Field (CRF) (Kim et al., 2011; Chung, 2009; Chung and Coiera, 2007) and Multi-Layer Perceptron (MLP) (Boudin et al., 2010a; Huang et al., 2011). Also Boudin et al. in (Boudin et al., 2010b) proposed a location-based weighting strategy as an extension to the language modeling approach inspired by the special distribution pattern of PICO elements in medical abstracts. All these models heavily rely on careful selections of hand-engineered features including lexical features such as bag of words (BOW), stemmed words and cue-words/verbs, and semantic features such as synonyms and hypernyms provided by some ontologies (*e.g.,* WordNet). As an important complement to this task, most recent work from Dernoncourt et al. (Dernoncourt et al., 2016) proposed the model based on currently emerging deep ANN architectures such as LSTM for further performance boosting, as well as to remove the need for hand-crafted features. However, this work has not targeted to address the issue of PICO element detection.

To generate the datasets for both training and test, earlier works mainly relied on manual annotation, which resulted in small corpora on the order of hundreds of abstracts (Demner-Fushman and Lin, 2007; Dawes et al., 2007; Chung, 2009; Kim et al., 2011). Afterwards, later works made use of the structural information embedded in some abstracts for which the authors have clearly stated distinctive sentence headings (Boudin et al., 2010a; Huang et al., 2011, 2013). Specifically, some abstracts contain explicit headings such as “PATIENTS”, “SAMPLE” or “OUTCOMES”, which can be used to locate sentences corresponding to PICO elements. In this way, tens of thousands of abstracts that contain PICO elements from PubMed can be automatically compiled as a well-annotated dataset, which can increase the size of dataset by two orders of magnitude.

In terms of task formulation, most previous works focused on categorizing one PICO class at a time using an individual classifier (Boudin et al., 2010a; Huang et al., 2013). Therefore, in order to detect all four PICO components, one would need to build and train four individual models, which is inefficient. Furthermore, it is hard to disambiguate the classification label conflicts between different model predictions on the same sentence. These limitations were resolved by working directly on the labels of interest for EBM, allowing multi-label classification instead of binary and allowing sentences that are unrelated to labels of interest to be labeled as an ”Other” category (Kim et al., 2011; Demner-Fushman and Lin, 2007). This is a

more realistic setting and ought to provide better insight into the performance we should expect for this kind of task.

3 The Proposed Model

First we introduce our notation. We denote scalars in italic lowercase (e.g., k), vectors in bold lowercase (e.g., $\mathbf{s}$) and matrices in italic uppercase (e.g., W). Colon notations $x_{i:j}$ and $\mathbf{s}_{i:j}$ are used to denote the sequence of scalars $(x_i, x_{i+1}, ..., x_j)$ and vectors $(\mathbf{s}_i, \mathbf{s}_{i+1}, ..., \mathbf{s}_j)$.

Our model is composed of three components: the token embedding layer, the sentence-level label inference layer, and the label sequence optimization layer (Figure 1). In the following sections they will be discussed in detail.

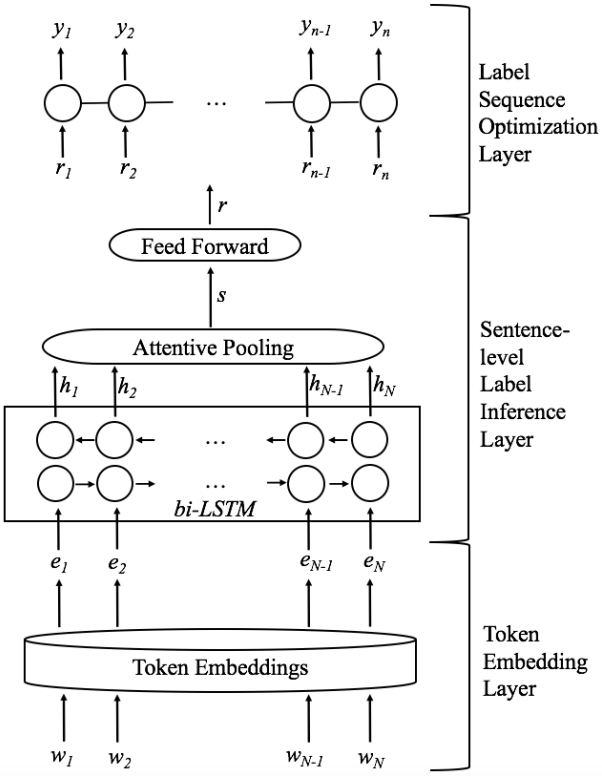

Figure 1: Model architecture. **w**: original token; **e**: token embedding; **h**: bi-LSTM hidden state; **s**: sentence representation vector; **r**: sentence label probability vector; y: predicted sentence label. Replacing bi-LSTM with convolutional neural network (CNN) did not improve the results: we therefore used bi-LSTM.

3.1 Token Embedding Layer

This layer takes as input a given sentence **w** comprising N words $\mathbf{w} = [w_1, w_2, ...w_N]$ and outputs its corresponding vector representation. Token representations are encoded by the column vector in the embedding matrix $W^{word} \in \mathbb{R}^{d^w \times |V|}$, where d^w is the dimension of the word vector and V is the vocabulary of the dataset. Each column $W_i^{word} \in \mathbb{R}^{d^w}$ is the word embedding vector for the i^{th} word in the vocabulary. To transform a certain word w into its corresponding embedding vector e^w, we use the following equation:

$$\mathbf{e}^w = W^{word}\mathbf{v}^w, \tag{1}$$

where $\mathbf{v}^w$ is the one hot vector of word w with dimension of $|V|$ that has 1 at the corresponding index and zero in all other positions. The word embeddings W^{word} can be pre-trained on large unlabeled datasets using unsupervised algorithms such as word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and fasttext (Bojanowski et al., 2016).

3.2 Sentence-level Label Inference Layer

This layer takes as input the embedding vector **e** of each token in a sentence from the token embedding layer and produces a vector $\mathbf{r} \in \mathbb{R}^l$ to represent the probability that this sentence belongs to each label, where l is the number of labels. To this aim, the sequence of embedding vectors **e** is first input into a bi-directional LSTM (bi-LSTM), which outputs a sequence of hidden states $\mathbf{h}_{1:N}$ ($\mathbf{h} \in \mathbb{R}^{d^h}$) for a sentence of N words with each hidden state corresponding to a token. To form the final representation vector **s** of this sentence, attentive pooling is used, which can be described using the following equations (Yang et al., 2016):

$$\mathbf{u}_i = \tanh(W_s\mathbf{h}_i + \mathbf{b}_s), \tag{2}$$

$$\alpha_i = \frac{\exp(\mathbf{u}_i^\top \mathbf{u}_s)}{\sum_j \exp(\mathbf{u}_j^\top \mathbf{u}_s)}, \tag{3}$$

$$\mathbf{s} = \sum_i \alpha_i \mathbf{h}_i, \tag{4}$$

where $\mathbf{u}_s \in \mathbb{R}^{d^s}$ is the token level context vector used to measure the relevance or importance of each token with respect to the whole sentence, and $W_s \in \mathbb{R}^{d^s \times d^h}$ is the transformation matrix for soft alignment.

The obtained vector **s** is subsequently input to a feed-forward neural network with only one hidden layer, which outputs the corresponding probability vector **r**.

3.3 Label Sequence Optimization Layer

Each medical abstract consists of several sentences with the sentence category following some patterns, such as that the category "Results" is always followed by "Conclusion". Such patterns can yield better classification performance via the conditional random field (CRF) algorithm. Given the sequence of probability vectors $\mathbf{r}_{1:n}$ from the last label inference layer for an abstract of n sentences, this layer outputs a sequence of labels $y_{1:n}$, where y_i represents the predicted label assigned to the i^{th} sentence.

In order to model dependencies between subsequent labels, we incorporate a matrix T that contains the transition probabilities between two subsequent labels; we define $T[i, j]$ as the probability that a token with label i is followed by a token with the label j. The score of a label sequence $y_{1:n}$ is defined as the sum of the probabilities of individual labels and the transition probabilities:

$$s(y_{1:n}) = \sum_{i=1}^{n} \mathbf{r}_i(y_i) + \sum_{i=2}^{n} T[y_{i-1}, y_i]. \quad (5)$$

The score in the above equation can be transformed into the probability of a certain label sequence by taking a softmax operation over all possible label sequences:

$$p(y_{1:n}) = \frac{e^{s(y_{1:n})}}{\sum_{\hat{y}_{1:n} \in Y} e^{s(\hat{y}_{1:n})}}, \quad (6)$$

where Y denotes the set of all possible label sequences. During the training phase, the objective is to maximize the probability of the gold label sequence. While in the testing phase, given an input sequence, the corresponding sequence of predicted labels is chosen as the one that maximizes the score using the Viterbi algorithm (Forney, 1973).

4 Experiments

4.1 Dataset Preparation

The dataset used in this study[1] is curated from MEDLINE, which is a free access database on medical articles. Specifically, we extracted 489,026 abstracts from PubMed by stating the following search limits: 1. Text Availability: Abstract; 2. Languages: English; 3. Publication Types: Randomized Controlled Trial (Search conducted on 2017/08/28). Among them, abstracts with structured section headings were selected for automatic annotation of sentence category. Although P, I and O headings were our detection targets, we also annotated the other types of sentences into one of the AIM (A), METHOD (M), RESULTS (R) and CONCLUSION (C) labels to facilitate the use of our CRF label sequence optimization method. Note that, although we have 7 labels in total, we only care about the detection accuracy of the P, I and O labels and thus mainly discuss their performance in the following sections.

In this study, the C component was incorporated into the I category since the "COMPARISON" section also refers to a kind of intervention in an RCT. And in fact, there are very few abstracts with comparison labels found in PubMed.

We annotated a certain section heading into one of the 7 labels based on whether it contains the key words that belong to the assigned label as shown in Table 1 (section headings are only used to generate gold labels and not used for model training and inference). In very rare cases, the section heading of a certain sentence may contain the key words of more than one category, in which case that sentence will be assigned into multi-labels according to Table 1. Table 2 presents a typical abstract example with section headings annotated into the 7 labels. A total of 24,668 abstracts contain at least one of the P/I/O labels. There are 21,198 abstracts with P-labels, 13,712 with I-labels and 20,473 with O-labels (Table 3). Note that, the abstracts in PubMed follow a diversity of rhetorical structure and only a small fraction of them contain PICO elements based on their section headings.

4.2 Training Settings

Ten-fold cross-validation was employed to assess the results statistically, where abstracts were randomly split into 10 equal partitions. Nine of them were used for training and the remaining one for testing. This step repeats for ten rounds. For each round of training, 10% of the training set was randomly extracted as the development set for early stopping, that is, the test set was evaluated at the highest development set performance, which is measured by the average F1 score of all three P/I/O labels.

The token embeddings were pre-trained on a large corpus combining Wikipedia, PubMed and

[1] https://github.com/jind11/PubMed-PICO-Detection

Category	Heading Name Key Words
Aim (A)	Objective, Background, Purpose, Importance, Introduction, Aim, Rationale, Goal, Context, Hypothesis
Participants (P)	Population, Participant, Sample, Subject, Patient
Intervention (I)	Intervention
Outcome (O)	Outcome, Measure, Variable, Assessment
Method (M)	Method, Setting, Design, Material, Procedure, Process, Methodology
Results (R)	Result, Finding
Conclusion (C)	Conclusion, Implication, Discussion, Interpretation

Table 1: Key words of section headings in structured abstracts for automatic annotation.

Heading Name	Cate.	Sentences
AIMS	A	[...] The aims of the trial were to test for differences between standard 1-and 0.5-mg doses (both twice daily during 8weeks) in (1) abstinence, (2) adherence and (3) side effects.
DESIGN	M	Open-label randomized parallel-group controlled trial with 1-year follow-up. [...]
SETTING	M	Stop-Smoking Clinic of the Virgen Macarena University Hospital in Seville, Spain.
PARTICIPANTS	P	The study comprised smokers (n=484), 59.5% of whom were men with a mean age of 50.67years and a smoking history of 37.5 pack-years.
INTERVENTION	I	Participants were randomized to 1mg (n=245) versus 0.5mg (n=239) and received behavioural support, which consisted of a baseline visit and six follow-ups during 1year.
MEASUREMENTS	O	The primary outcome was continuous self-reported abstinence during 1year, with biochemical verification. [...] Also measured were baseline demographics, medical history and smoking characteristics.
FINDINGS	R	Abstinence rates at 1year were 46.5% with 1mg versus 46.4% with 0.5mg [odds ratio (OR)=0.997; 95% confidence interval (CI) = 0.7-1.43; P=1.0]; [...]
CONCLUSIONS	C	There appears to be no difference in smoking cessation effectiveness between 1mg and 0.5mg varenicline, [...].

Table 2: A typical abstract example with section headings and their corresponding annotated labels. The PMID of this abstract is 28449281.

Category	Abstracts	Sentences
P	21,198	27,695
I	13,712	24,602
O	20,473	32,525

Table 3: Number of times each of the categories P, I and O appear in abstracts and in sentences in the data.

PMC texts (Moen and Ananiadou, 2013) using the word2vec tool[2]. They are fixed during the training phase to avoid over-fitting[3].

The model is trained using the Adam optimization method (Kingma and Ba, 2014). For regularization, dropout is applied to each layer and l_2 regularization is also used. Hyperparameters were optimized via grid search and the best configuration is shown in Table 4. Code for this work is available online[4].

[2]https://code.google.com/archive/p/word2vec/

[3]http://bio.nlplab.org/

[4]https://github.com/jind11/LSTM-PICO-Detection

Para.	Para. Name	Value
d^w	Token Embed. Size	200
d^h	LSTM Hidden Size	150
d^s	Attention Vector Size	300
bz	Batch Size	40
lr	Learning Rate	0.001
β	l_2 Regularization Ratio	0.0001

Table 4: Hyperparameters. Batch size refers to the number of abstracts in one batch.

5 Results and Discussion

Table 5 and 6 detail the results of classification for each label in terms of performance scores (precision, recall and F1) and confusion matrix, respectively (for one fold). It can be seen that the classifier is very good at predicting the labels of AIM, RESULTS and CONCLUSION but has difficulty in distinguishing among the labels of PARTICIPANTS, INTERVENTION, OUTCOME and METHOD. Indeed, the PARTICIPANTS, INTERVENTION and OUTCOME sections can be deemed as more specific aspects of the METHOD descriptions, therefore, it is naturally more difficult to tell the P/I/O elements apart from the METHOD section. Since our main goal is to accurately extract the P/I/O components from a given abstract, we will only discuss their performance in the following.

Cate.	p (%)	r (%)	F1 (%)	Support
A	97.7	98.0	97.8	3811
P	88.5	82.8	85.6	2722
I	74.9	81.5	78.1	2331
O	84.5	83.2	83.8	3219
M	87.0	84.2	85.6	5623
R	93.3	96.4	94.8	9236
C	93.8	91.1	92.5	4312
Total	90.1	90.0	90.0	31254

Table 5: Results in terms of precision (p), recall (r) and F-measure (F1) on the test set for each class obtained by our model for one of the ten folds.

Table 7 compares our model against several previously widely-used baseline models. Since there is no benchmarking dataset, we cannot compare with published best models (this is one of the reasons why we want to publish this dataset).

The first baseline is the logistic regression (LR) model that uses the n-gram features extracted from the current sentence for classification. In this scenario, each sentence is predicted individually without context information from the surrounding sentences considered. Likewise, the second baseline MLP first computes the vector representation for each sentence by taking the max pooling operation of the embeddings of all tokens in the sentence, then classifies the current sentence via a neural network with three hidden layers (hidden layer dimensions are 400, 400 and 200, respectively). On the other hand, the third baseline is a CRF model that also uses n-grams as features (only the first 100 tokens were used for each sentence since most sentences are shorter than 100 tokens) and outputs the most probable label sequence for the whole abstract. Therefore, the CRF baseline takes into account both preceding and succeeding sentences when classifying the current sentence.

	P	M	C	A	R	O	I
P	2213	197	5	29	84	49	145
M	181	4804	9	40	30	242	317
C	0	6	3904	8	393	1	0
A	4	43	3	3743	6	11	1
R	9	21	175	0	8952	65	14
O	15	277	11	20	136	2688	72
I	40	278	0	0	28	142	1843

Table 6: Confusion matrix obtained by our model for one of the ten folds. Rows correspond to predicted labels, and columns correspond to true labels.

As presented by Table 7, the LR baseline performs worst, which is quite reasonable considering that it is still a very shallow model and only uses the local sentence information. As a comparison, the MLP model also only considers the features from the current sentence but performs better than LR because its modeling capacity is much larger. By incorporating the surrounding sentences, the CRF baseline performs even better than MLP system, which verifies that context information is quite useful in sequential classification problems.

Lastly but most importantly, our proposed model performs much better than all the baselines for all three P/I/O labels. The advantages of our model and the reasons for its improved performance are summarized below:

No human-engineered features Our model does not rely on any hand-engineered features that require much domain experience and are quite dif-

Models	P-element (%)			I-element (%)			O-element (%)		
	p	r	F1	p	r	F1	p	r	F1
LR	66.9	68.5	67.7	55.6	55.0	55.3	65.4	67.0	66.2
MLP	77.8	74.1	75.8	64.3	65.9	64.9	73.8	77.9	75.8
CRF	82.2	77.5	79.8	67.8	70.3	69.0	76.0	76.3	76.2
Our Model	**87.8**	**83.4**	**85.5**	**72.7**	**81.3**	**76.7**	**81.1**	**85.3**	**83.1**

Table 7: Performance in terms of precision (p), recall (r) and F-measure (F1) on the test set with several baselines and our proposed model (average value based on 10 fold cross validation). Since the dataset used here was introduced in this work, there is no previously published method for reference.

ficult to craft.

No n-gram features Unlike many other systems that rely heavily on n-grams, our model simply uses the token embedding vector to represent each token and feeds it into the recurrent neural network (RNN) model for inference. In this way, the pre-trained embeddings on large corpora can encode the syntactic and semantic information of words for better language understanding. This can also help combat word scarcity problem. For example, the alternatively spelled tokens "tendonitis" and "tendinitis" are two different unigrams, however, their semantic meanings are the same, and this similarity can be revealed by their corresponding closely parallel embedding vectors.

Joint prediction Instead of predicting each sentence one by one, our model classifies all sentences in one abstract jointly, which improves the overall classification performance by implying the constraints of coherency between subsequent predicted labels. This improvement is clearly evidenced by Table 8.

Sequence modeling An RNN model is good at modeling sequences such as sentences by considering the dependency between tokens, which cannot be accounted for by context-free models such as those using bag of words features. And the long-term memory characteristic of LSTM model further grants the RNN model the ability to cope with long sentences.

Figure 2 presents an example of the transition matrix after the model has been trained, which encodes the transition probability between two subsequent labels. It effectively reflects what label is the most likely one that should follow the current one. For example, a sentence pertaining to the RESULTS is typically followed by a sentence pertaining to the CONCLUSION (1.16), which makes sense. From this transition matrix, we can figure out the most probable label sequence: $A \rightarrow M \rightarrow P \rightarrow I \rightarrow O \rightarrow R \rightarrow C$, which is also consistent with our observations.

Model	F1 (%)		
	P	I	O
Full Model	**85.5**	**76.7**	**83.1**
-sequence optimization	78.2	68.2	78.3

Table 8: Ablation analysis. 10 fold cross validation F1-scores are reported. "-sequence optimization" is our model without the label sequence optimization layer.

Table 9 presents a few examples of prediction errors that are related to P/I/O labels. This error analysis suggests that part of the model error comes from the ambiguity between some label pairs, such as O and M, O and R, and I and M. For example, the sentence "Plasma volume and total body haemoglobin were determined at rest." can be deemed as a METHOD description in a general sense, however, it can also be further specified as an OUTCOME. On the other hand, a fair number of sentence labels are indeed debatable. For instance, the sentence "Iron supplementation was given to one group as a substitution remedy, another group was given iron and folic acid and the third group was without supplementation during the collection period." belongs to the PARTICIPANT label according to the gold standard, but it makes more sense that it should be classified as an INTERVENTION.

6 Conclusion

In this work we have presented an LSTM based ANN architecture to detect the PICO elements in medical RCT abstracts. We demonstrated that the use of a more advanced LSTM model and jointly predicting the classes of all sentences in a given text can improve the overall classification perfor-

Sentence	Predicted	Gold
The study included 16 patients who were randomized into one of three 6-month treatment protocols.	P	M
Referral service doing n-of-1 trials at the requests of community and academic physicians.	I	M
Iron supplementation was given to one group as a substitution remedy, another group was given iron and folic acid and the third group was without supplementation during the collection period.	I	P
Plasma urea and creatinine concentrations and angiotensin converting enzyme activity were measured at the start of the study and the end of each treatment period.	O	R
Heart rate was recorded continuously throughout the maneuvre, while blood was sampled for catecholamine determinations prior to the start of straining and again approximately 10 s following the end of straining.	O	I
Plasma volume and total body haemoglobin were determined at rest.	O	M

Table 9: Examples of prediction errors of our model that are related to P/I/O labels. The "Predicted" column indicates the label predicted by our model for a given sentence. The "Gold" column indicates the gold label of the sentence.

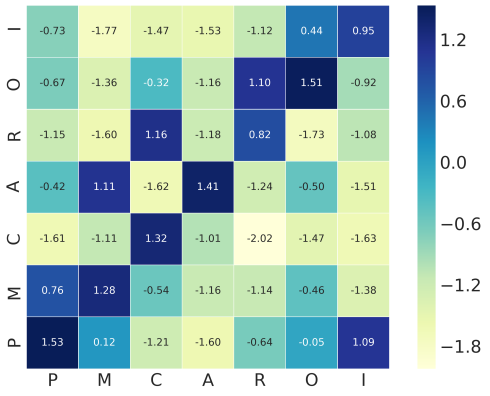

Figure 2: Transition matrix of label sequence. The rows represent the label of the previous sentence, while the columns represent the label of the current sentence.

mance of PICO components. And by publishing our curated dataset for benchmarking, we hope to encourage competition by other approaches than ours and that more effective and efficient methods can be developed in the future.

Acknowledgments

This work was supported by funding grant U54-HG007963 from National Human Genome Research Institute (NHGRI). Thank Matthew McDermott for helping revise the manuscript.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Florian Boudin, Jian-Yun Nie, Joan C Bartlett, Roland Grad, Pierre Pluye, and Martin Dawes. 2010a. Combining classifiers for robust pico element detection. *BMC medical informatics and decision making*, 10(1):29.

Florian Boudin, Jian-Yun Nie, and Martin Dawes. 2010b. Clinical information retrieval using document and pico structure. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 822–830. Association for Computational Linguistics.

Florian Boudin, Lixin Shi, and Jian-Yun Nie. 2010c. Improving medical information retrieval with pico element detection. In *European Conference on Information Retrieval*, pages 50–61. Springer.

Grace Y Chung. 2009. Sentence retrieval for abstracts of randomized controlled trials. *BMC medical informatics and decision making*, 9(1):10.

Grace Y Chung and Enrico Coiera. 2007. A study of structured clinical abstracts and the semantic classification of sentences. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 121–128. Association for Computational Linguistics.

Martin Dawes, Pierre Pluye, Laura Shea, Roland Grad, Arlene Greenberg, and Jian-Yun Nie. 2007. The identification of clinically important elements within medical journal abstracts:

Patient_population_problem, exposure_intervention, comparison, outcome, duration and results (pecodr). *Journal of Innovation in Health Informatics*, 15(1):9–16.

Dina Demner-Fushman and Jimmy Lin. 2007. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1):63–103.

Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2016. Neural networks for joint sentence classification in medical paper abstracts. *arXiv preprint arXiv:1612.05251*.

G David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Marie J Hansen, Nana Ø Rasmussen, and Grace Chung. 2008. A method of extracting the number of trial participants from abstracts describing randomized controlled trials. *Journal of Telemedicine and Telecare*, 14(7):354–358.

Ke-Chun Huang, I-Jen Chiang, Furen Xiao, Chun-Chih Liao, Charles Chih-Ho Liu, and Jau-Min Wong. 2013. Pico element detection in medical text without metadata: Are first sentences enough? *Journal of biomedical informatics*, 46(5):940–946.

Ke-Chun Huang, Charles Chih-Ho Liu, Shung-Shiang Yang, Furen Xiao, Jau-Min Wong, Chun-Chih Liao, and I-Jen Chiang. 2011. Classification of pico elements by text features systematically extracted from pubmed abstracts. In *Granular Computing (GrC), 2011 IEEE International Conference on*, pages 279–283. IEEE.

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. In *BMC bioinformatics*, volume 12, page S5. BioMed Central.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

SPFGH Moen and Tapio Salakoski2 Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan*, pages 39–43.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

W Scott Richardson, Mark C Wilson, Jim Nishikawa, and Robert SA Hayward. 1995. The well-built clinical question: a key to evidence-based decisions. *ACP journal club*, 123(3):A12–A12.

David L Sackett. 1997. *Evidence-based Medicine How to practice and teach EBM*. WB Saunders Company.

David L Sackett, William MC Rosenberg, JA Muir Gray, R Brian Haynes, and W Scott Richardson. 1996. Evidence based medicine: what it is and what it isn't.

Connie Schardt, Martha B Adams, Thomas Owens, Sheri Keitz, and Paul Fontelo. 2007. Utilization of the pico framework to improve searching pubmed for clinical questions. *BMC medical informatics and decision making*, 7(1):16.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Eya Znaidi, Lynda Tamine, and Chiraz Latiri. 2015. Answering pico clinical questions: a semantic graph-based approach. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 232–237. Springer.