Text Mining in the Medical Domain: Automated extraction of PICO elements for systematic reviews

Soufyan Belkaid
MA Text Mining
Vrije Universiteit
Amsterdam, Netherlands
soufyan@isda.xyz
2525048

Thomas Maaiveld
MSc Artificial Intelligence
Vrije Universiteit
Amsterdam, Netherlands
t.m.maaiveld@student.vu.nl
2528586

Ersin Topuz
MSc Artificial Intelligence
Vrije Universiteit
Amsterdam, Netherlands
e.topuz@student.vu.nl
2530852

Sanne Hamersma
MSc Text Mining
Vrije Universiteit
Amsterdam, Netherlands
s.p.hamersma@student.vu.nl
2580904

Aju Shrestha
MA Text Mining
Vrije Universiteit
Amsterdam, Netherlands
a5.shrestha@student.vu.nl

***Abstract*—This paper presents an approach to automating the process of annotating medical text by Machine Learning methods. Two different models are applied to extract PIO elements from abstracts of medical publications. To examine the differences between a sequence-based model and an instance-based model, a Conditional Random Field model and a Random Forest model are trained on an annotated corpus of medical journal abstracts and their performance is evaluated and compared. We discuss the influence of choices of features and parameters, the effect of certain model design choices on the learning infrastructure and present an analysis of the results and frequent classification errors returned by each model. The results are connected to a broader discussion on the task of NLP processing of medical data and text annotation, including potential directions for future research.**

***Index Terms*—machine learning, text mining, natural language processing, evidence-based medicine, systematic reviews**

I. Introduction

Medical professionals make use of evidence based medicine (EBM) to provide health care and conduct research. The core motive in EBM is to make use of all evidence available to provide information for patient care to allow for the provision of health care based on robust, up-to-date information. This evidence mainly consists of corpora of publications describing randomized control trials (RCT) in biomedical journals [1]. RCT are scientific experiments where, in order to limit specific factors of bias, participants are randomly assigned into two or more groups that are subjected to different conditions and subsequently compared. In conducting these, the effectiveness of certain treatment(s) for people with specific clinical conditions is examined [2] [3]. These studies are catalogued into systematic reviews that focus on a precise clinical question in composing all relevant published evidence into a comprehensive synthesis.

An important aspect of systematic review are the PICO elements that are used to describe and categorize research carried out in database publications. 'PICO' refers to the Population: subjects of an experiment; Intervention: the treatment(s) applied; Comparison: treatment(s) for the control group(s); and Outcome of a study. The Intervention and Comparison are typically grouped together. The precise clinical question, which is at the center of a systematic review, is derived via compartmentalisation into PICO frames. These criteria shape the foundation for the obtaining and inclusion of published evidence in a review. Systematic reviews additionally contain Study type: whether the article describes a RCT, a clinical trial and so forth, and Context: the environment where the Intervention is introduced and interacts [4], while the PICO elements cover the core research.

This research focuses on the the task of extracting sequences from peer-reviewed medical journal text that denote the Population, Intervention and Outcome (PIO) of that study. Given a sequence of sentences, we aim to construct a model that can label each token in each sequence with a P, I and O label, modeling the task as a multilabel classification problem. Since meta-reviews usually only describe RCTs, a corpus consisting of RCTs was used and the classification task was limited to the remaining three labels. The scope was limited to labeling abstracts of publications with two different models, in light of limits on time, computational resources and availability of annotated data. To provide a research contribution to automated medical text annotation, we conduct a comparison of results between two modeling approaches. Additionally, we examine the effect of different feature selection procedures for this task. While a full-fledged tool that directly addresses the specified use case proved infeasible, we present the results of our analysis in this report, as well as providing the code base

on GitHub[1].

The paper gives a description of the background and related work in the domain of medical text mining and sequence extraction, as well as the envisioned use case that underlines to the need for the PICO elements in creating systematic reviews. Subsequently, we describe related work in the field and discuss the performance of previous applications to the PICO extraction task. In describing the use case, the functional and non-functional requirements for a domain specific user in this field are outlined. Baseline models from related work are discussed in the context of the chosen data set which will serve as a point of comparison for this implementation. The methodology section describes the applied models and extracted features, while the results and analysis discuss the outcomes of the research.

II. Background

As the immense stream of data in biomedical science continues to grow, with more than 27000 publications of clinical trials in 2012, systematic reviews are essential in providing graspable overviews. They are regarded to be the highest level of resource for evidence in the field of medicine and health care and within it, their range of usage is all encompassing: from practical patient care to government level policy-making [5]. Systematic reviews are invaluable to healthcare, yet their creation in terms of time and finance are costly. The manual production of a systematic review is a highly labour intensive and tremendously time-consuming process. A 2017 study estimates the construction of a review to require 67 weeks, 11256 hours, from registration to publication [6]. Moreover, this labour is performed by highly trained workers - often medical doctors - and results in this process to be tied to a large financial weight. In addition to that, due to the growing data influx, reviews rapidly go out of date. The current EBM methods, as a result, fail to keep up with the massive and consistently expanding evidence base.

In this light, the (semi-)automation of the production of systematic reviews is an important innovation within this field to dramatically increase efficiency. This has formed into its own sub-field where methods in Natural Language Processing, Machine Learning and Text Mining are used to extract specific data from the unstructured data stream of natural language in biomedical articles [7]. The automation of systematic reviews can be divided into (1) the aggregation of a corpus of relevant articles (which can again be divided into sub-tasks); (2) extracting specific data from these texts; and (3) data synthesis review is contained.

A. Related Work

Studies on extracting PICO elements so far have yet to produce a fully automated system and have mainly been focused on semi-automation involving a human reviewer. Previous studies have applied rule-based methods [8], machine learning models or a combination of both [9].

[1]github.com/tmaaiveld/tmd_pico

Among the challenges specific to this task are the scarcity in training data, resulting in the models either being trained on a relatively small data set of manually annotated texts, or the applications of algorithmically retrieved labels - based on existing structured data - on a large dataset known as “distant-supervision” (DS). As creating human-annotated data is an expensive and lengthy process, DS offers an option worthy of exploring. As this method aggregates a large corpus, increasing the training data, it presents the benefit of bigger data sets commonly resulting in improved model performance [10]. However, a disadvantage is that this approach allows for noise in the data, thus resulting in higher error rates than models using strictly human annotations for training [11].

While the usage of abstracts as training data allows for a larger corpus as it is quicker to annotate these given their concise form and size, full text articles provide a more detailed description of the desired PIO elements. Despite the benefits of using full text documents, abstracts have been proven to be useful for the extraction of clinically salient data and in supporting information retrieval in biomedical literature [9] [12] [13]. In the task of extracting data, models tend to over-retrieve, as a higher recall is favoured, as it is preferable that an outcome ensures studies are not excluded in the resulting systematic review [7].

There are currently very few ready-to-use applications available for the purpose of data extractions for EBM practitioners. Among them is RobotReviewer, of which a demo is readily available online to upload PDF formats of RCT reports. It retrieves the sentences and phrases that may contain the PICO elements. It is trained on 12,808 PDFs of full-text articles with semi-automated labeling using distant-supervision. The structure makes use of a soft-margin SVM model with two tasks: (1) it labels the articles with a risk of bias assessment (low, high, unclear); and (2) it simultaneously extracts the sentences relevant [11].

For an elaborate overview on data extraction for systematic review, see this relatively recent compilation [14].

B. Use Case & Application

Systematic reviews are vital for professionals in the very broad and essential fields of healthcare and medicine. There is a pressing need for automation of these reviews in order to realise cost-effectiveness and efficiently in their production. An additional need is for acceleration and automation of the process of updating reviews, to ensure they remain up-to-date with the consistent flood of data production in bio-medical research.

The ultimate instrument that we envisioned would be a ready-to-use database where a search query can be inserted. It is a given that at least one of the PIO elements is named in the query (e.g. ‘What is the effect of administering incretin mimetics (GLP-1 agonists) to a person diagnosed with Diabetes?’). The results of this query are the all relevant publications and their PIO elements. The database is constantly expanded as new biomedical publications are automatically uploaded to the database. As we are limited in time and resources for the

realisation of the full application, we can provide a research contribution towards the goal of developing such a system by testing and comparing various machine learning approaches on this task.

III. Data set and Baselines

This section outlines the data set used to train the models for the PIO elements prediction task. We also briefly explain the baseline that was introduced in [15].

A. Data Set

EBM-NLP, the medical abstract text corpus used in this research, contains 4,993 annotated abstracts of medical articles and is described in [15]. The annotations were generated by crowd-sourcing the task of labeling to multiple annotators with different domain expertise and aggregating the results. First, the abstracts were marked with the spans that may contain a PICO element, after which a finer grained distinction is made by also assigning a label of the gradient of the category. The data set was subsequently labeled by domain experts, in this case people with medical training, which is considered the gold data. The data is saved as separate documents in the source data, identified by their PubMed ID. The tokens and labels are provided in pairs using a tokenization scheme outlined in [15]. This pre-tokenized format of the data set required a specialized approach to indexing it correctly. First, sentence breaks in the data were marked by detecting periods, and the data was indexed on a document, sentence and token level. This ensured that the required data structures to compute the features given in the Methods section were available. The labels were separated and treated as binary labels for the purposes of classification (one-against-all). In the interest of focusing on the task of extracting PIO label text sequences, only the starting spans are used as data in this research.

B. Evaluation

493 abstracts (circa 10%) were withheld from the data and used as a test set to produce the final prediction scores for the models presented. Because of constraints on time and computation resources, a full k-fold cross-validation and parameter tuning procedure could not be implemented. However, by choosing models that calculate explicit importances for each feature they process, insights may be gained into what information is informative in the task of medical text classification. Evaluations were conducting using the standard measurements of precision, recall and f1-score, in order to allow for comparison with published work on the subject. In the medical domain, high recall is considered a higher priority in most prediction tasks, as most problems feature imbalanced class importances, and failing to classify certain events can be more consequential than succeeding in identifying their absence. In the PIO extraction task, a similar focus is adopted in order to address the use case outlined in the previous section.

	Participant	Interventions	Outcome
precision	0.55	0.65	0.83
recall	0.51	0.21	0.17
f1-score	0.53	0.32	0.29

TABLE I: CRF Baseline system results [15]

C. Baseline

Two baseline systems were introduced by the authors. A linear conditional random field (CRF) and a Long Short-Term Memory (LSTM) neural tagging model. For the CRF model the following features were used: current, previous and next word, POS-tag generated by using the Standford CoreNLP[2] tagger and separate character information of the word.

As for the LSTM model, the distributed vector representation of the token was used and the separate character information of the word. As this approach differed significantly from the tasks applied in this research, only the CRF model was included in this overview. Also, the baseline CRF was used for the sub-task token based labeling which is in agreement with our interpretation of the task. The result for the baseline model will be offset against our CRF model in the methods section.

IV. Methods

Exploratory Analysis The project was started with a literature review, where the specifics of the medical text domain, systematic review and text mining in this domain were considered. The biomedical publications often had a unique set of words and an academic writing style. A research of previous work in this domain was carried out focusing on machine learning approaches of subtasks. The feasibility of implementations and challenges in the task became clear. Furthermore, to achieve insight in the need for automatizing of systematic reviews a meeting with Cochrane took place. Cochrane is an international organization carrying out systematic reviews and organizing them in a database. This meeting helped to form the use case.

In order to label and extract sequences in the data, the features the model is trained on should be able to represent the context the word appears in in the text. Some models may naturally process contextual information by batching or sequencing the data, usually imposing conditions on the length or dimensionality of the input as a consequence. In order to examine the differences between a sequence-based model and an instance-based model, two models were trained to predict PIO labels, a Conditional Random Field (CRF) model, and a Random Forest (RF) model.

A. Conditional Random Field Classifier

In [16], CRF is recommended as a sequence labeling model because of its undirected graphical based nature. It assumes features are dependent and considers future observations while learning by implementing dependencies between the predictions. This means that CRF takes context words into account

[2]https://stanfordnlp.github.io/CoreNLP/

in comparison with a classifier which assigns one label to a sample and disregards the neighbouring samples. To achieve this, the context of a token needs to be represented by the features. Since CRF is a graphical-based model, it can handle and responds well to categorical variables such as tokens or label tags. Because of these properties, the CRF algorithm seems highly applicable in this sequence labelling task of PIO elements. Unlike LSTM or LSTM-CRF, it is not limited to a fixed input length and can handle the variable-length sentences of the corpus. Moreover, categorical features with a large amount of levels are much more sparsely represented than in a one-hot coded format. Sentences were indexed and predictions generated; the resulting predictions were mapped to each token. A separate model was trained for each variable. An additional special feature of the CRF model is its ability to link its prediction to the prediction of its neighbours. Probabilistic mappings between the tokens should cause tokens in the same sequence to be more strongly related, as they are represented in the text.

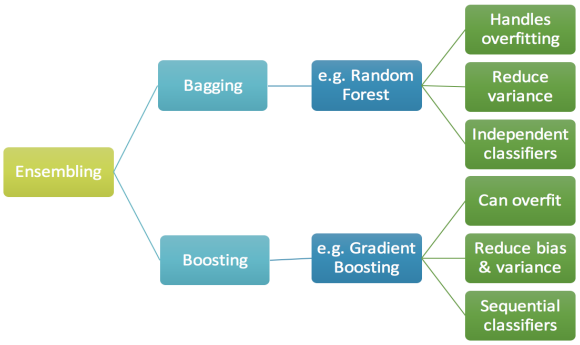

Fig. 1: Two most used ensembling methods: Bagging and Boosting

B. Random Forest Classifier

Random Forest classifiers use a technique called Bagging (Boostrap Aggregation) to aggregate many weak learning algorithms into one strong predictive model. By training a large number of shallow decision tree classifiers on the data and averaging these predictions, a class probability and label are generated for each sample. This type of ensembling method is useful for classification tasks because it explicitly gives the importance of features. The model only allows for categorical input if it is one-hot coded. Therefore, inclusion of categorical features with a large number of levels results in a sparse feature space with a high dimensionality, which adversely affects model performance. Using the importances the model assigns to each feature, feature selection can be performed to reduce the feature space, and dimensionality reduction techniques can assist in making the features more informative for the model. A shortcoming of tree ensemble methods is that they cannot easily learn from sparse categorical variables. Fully one-hot-coding every categorical feature would produce a feature space of around 1.250.000 tokens and around 600,000 features, which would be impossible to learn from. Thus, dropping such features and reducing the dimensionality of the data proved an important part of developing this model. Additionally, unlike CRF, tree models have no natural way of representing indexed sequences. Each token is labelled individually, uninfluenced by its neighbours' labelings. This necessitated the addition of tokens to represent information of the neighbours and relations of a token.

C. Features

A broad range of features were extracted from the data to map the positional, syntactic and semantic information contained in the corpus to a feature space interpret-able by both models. To present a clear overview, the features are presented in categories in the following subsections.

1) Semantic Embeddings:

Using a pre-trained FastText[3] model trained by the authors of [17], embeddings of 200 dimensions were generated for the data set. The BioWordVec model was trained on a corpus of 1.7 million medical text documents and is highly suited to mapping domain-specific embeddings as features. The downside of this approach is the increase in the dimensionality of the feature set, and the introduction of a set of numerical variables. The features proved informative in early testing on small datasets and was included in the final results.

Additionally, to examine whether polarity or subjectivity bias in medical text could be indicative of a PIO label. Sentiment annotations were applied to the sentences in the data, producing two numerical features representing the polarity and subjectivity of the sentence the word is contained in.

2) Word Sentence Information:

The part-of-speech tag, dependency relation, and lemma were parsed to CoNNL format to provide the model with syntactical information related to other words in its sentence. The CoNNL format provides base word-level information, but also maps the relations in the corpus to a row format. This method of decoding parse trees is useful for inputting tree-based relations to an instance-based model. The procedure for parsing the relations is described in Besides parsing the dependency tree data and tags, several additional tags were formulated: whether the token is an integer or decimal number, whether it is in a set of stop words, or whether it is the first word of the sentence. The TF-IDF of each token in the document was also calculated and added as a feature for each instance to represent the word's frequency. Adding the index information of the tokens proved of little value, as these unnormalized linear features constitute poor predictors in this task.

3) Lag Columns & Dependency Parsing:

To alleviate the difficulty tree ensemble methods have with modeling adjacency relationships in the data, a selection of features was shifted (lagged) on document level and padded to produce lag columns representing the value of a certain feature for an adjacent word. Examples include the part-of-speech tag or lemma of the token. This would increase the

[3]https://fasttext.cc/

feature space, but adding lag columns for binary features did not severely impact model performance or training time. A window size of 5 was applied for the lag columns: each token was given information on the preceding and following tokens, as well as the ones preceding and following those. The TF-IDF was also shifted to create a distribution of five TF-IDF values at each token's position.

The CoNNL data format includes a head relationship column, which points to the word in the sentence that is its parent in the syntactic parse tree. This feature was used to add information from each token's parent to its own features. These features included the lemma and POS tag for the CRF, the part-of-speech tag and dependency relation of the parent, and the distance to the parent as a numerical feature.

The features used for the model were the subject of broad experimentation over the course of the research project. Semantic and syntactic relations proved challenging to extract, as abstracts contain dense descriptions and formal language. The goal of this approach was to be able to model the necessary information for prediction on several linguistic levels. Since the applied Random Forest model explicitly calculates the importance of each feature during training, retrieving these values after initial training runs provided valuable insights into which features to explore further and which proved uninformative.

D. Feature selection & reduction

For the Random Forest classifier, categorical columns containing more than 100 levels were removed from the data, as applying a one-hot encoding to these columns resulted in too many variables. First, an attempt was made to scrub all but the highest-ranking tokens in a TF-IDF table, leaving some tokens to be placed in one-hot coded columns, but this proved counterproductive as such large feature spaces slowed down model training and negated any benefit they may offer. Low-dimensionality categorical features such as part-of-speech tags and dependency relation identifiers were included in both models. While lag columns were produced withinnly binary and part-of-speech tag columns were given a lag for the tree boosting model.

After extracting the PubMed embeddings from the data, Principal Component Analysis (PCA) was applied to reduce the dimensionality and remove unnecessary information the models are unable to retrieve. Principal components were fit to the generated embeddings, and subsequently selected by relevance to reduce the number of features. PCA is a validated technique for data dimensionality reduction [18]. Figure 2 shows the explained variance of each feature, ranked in descending order. By arbitrarily designating a cutoff point, a desired amount of features can be dropped by mapping the features to a smaller number of principal components in the feature space. $p < 0.9$ was used as a cutoff, which resulted in a dimensionality reduction of 62.

Since the applied Random Forest model explicitly calculates the importance of each feature during training, retrieving these values after initial training runs provided valuable insights into which features to explore further.

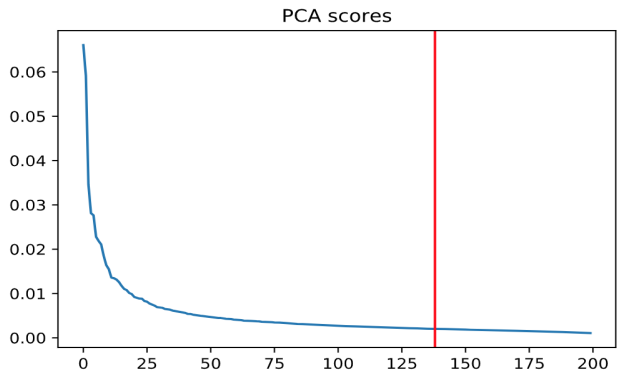

Fig. 2: Principal components ranked by their explained variance ratio. The red line indicates the cutoff point for feature relevance ($p < 0.9$)

1) Parameter Tuning:

Given the timespan and scope of the project, extensive parameter tuning, optimalization and cross-validation to achieve the best possible classification score proved impossible. Nonetheless, mapping the evaluation scores within the test set for a (similar) gradient boost tree tested during an earlier phase of the project provided insight into the operation of the tree depth parameter in ensemble tree methods. Early tests with various tree ensemble methods on the dataset showed that the model loss decreases faster per generation as the max tree depth is set higher. This is consistent with the expectation that deeper decision trees in ensemble methods can model more complex dependencies in the data, but also encourage overfitting. Furthermore, the amount of decision trees to be trained and the amount to be trained per iteration could be configured. However, performing a structured search over these parameters proved infeasible; thus, parameters for the Random Forest model were adjusted by trial-and-error and basic comparison of model scores, while the tree depth was configured by setting a parameter γ, a threshold for the information criterion used to select features to split on. Thus, if no leaf nodes qualify, the tree has reached its max depth. The chosen information criterion for features selection was the Gini impurity.

The parameter for For the CRF model, recommended parameter settings $c_1 = 0.1, c_2 = 0.1$ and $c_1 = 1.0, c_2 = 0.001$ were tested on 3000 documents. $c_1 = 0.1, c_2 = 0.1$ achieved a better score and were used as model parameters, and all possible transitions were considered. These parameters were used to tune the final model.

V. Results

The three metrics that were considered most important to the PIO extraction task were the precision, recall and F1-score of each model. For each metric, the average score of the Prediction, Intervention and Outcome label is calculated

for the given experiment. The results are presented in Tables II and III.

The classification results indicate that the CRF model and Random Forest models both had difficulty extracting certain labels from the data. The recall scores are low, although the CRF classifier appears to have outperformed the Random Forest classifier on all labels in this criterion. Both models score best on Participants, which is consistent with the baseline presented in [15]. The Random Forest model achieved an F1-score of 0.51, while CRF achieves a score of 0.65 on this label. While the CRF implementation described in this paper also fails to achieve a good recall and F1-score on the Outcome label, the Random Forest model appears to score significantly better in this regard. Medical outcomes may be heavily represented in the FastText embeddings, meaning the CRF model is not easily able to decode these semantic meanings and recognize a label. Although it can handle numerical input, the embeddings were not included in the most predictive features (given for the CRF model in the appendix).

CRF	Participants	Intervention	Outcome
precision	0.7422	0.5864	0.6637
recall	0.5797	0.4258	0.4720
f1-score	0.6509	0.4934	0.5517
support	17676	12992	17380

TABLE II: CRF

Random Forest	Participants	Intervention	Outcome
precision	0.7377	0.6483	0.6404
recall	0.3904	0.2741	0.3847
f1-score	0.5106	0.3853	0.4806
support	17809	13387	16178

TABLE III: Random Forest

VI. Analysis & Discussion

A. Model performance

Model performance was strongly affected by the selection of model parameters and selected data instances. Training on smaller subsets often misrepresented the effect of certain parameter choices, which made testing more difficult. As both models learn from semantic and syntactic features in the text, they are strongly dependent on the embedding information they are provided. While further tuning of weights and parameters could improve model performance and a cross-validated setup could produce scores that reflect a different comparison, a broad-level comparison indicates that the CRF model's advantage in decoding sentence-based information in the data gives it a significant advantage over the tree model design on the PIO extraction task. A key aspect of training the Random Forest classifier was the adjustment of the class weight. The standard rule of thumb for configuring a class weight for class imbalance is to calculate the ratio between the negative and the positive instances and use this as a weight. The same approach was used for the models applied in this research, which encouraged models to label with a lower margin for the positive class. This had a positive effect on the Random Forest model output, while lowering the precision as a trade-off. Nonetheless, tuning this parameter proved a difficult task, and the CRF model does not require class or input weights to generate unbiased predictions. The most important contrast between the two models used is the approach towards representing the data. The handling of the categorical or numerical features in the data determines how well the model is able to learn from this data, and this is reflected in the results in V. While the scores and predictions are not a complete mismatch, the CRF model does not appear to be able to formulate a good prediction from the categorical data it marks as important. The Random Forest model can extract semantic information from the numerical embeddings and principal components, but fails to consider more obvious semantic closeness in the tokens it does not classify. After a set number of learning iterations, the Random Forest model can no longer improve on its classification result. Naturally, we may not have explored the full range of possibilities of representing the presented features, nor did we conduct an exhaustive overview of the feature extraction and processing methods available in NLP sequence extraction. However, these results indicate some respective strengths and weaknesses that each model presents for this task, and these reflect on the quality of the data presented to the model and how applicable they are to its prediction method.

The handling of the multi-label classification problem as a set of binary classification problems is an obvious choice, although there are other approaches to be explored. A total of 3% of the tokens in the dataset have overlapping labels [15], which means that generating new labels for overlapping classes would be infeasible as they would have too little representation.

B. Error analysis

Apart from analyzing errors and improving the models over the course of the project, we provide an analysis of frequently encountered errors in this section. A frequently observed error when examining predictions of the model is related to punctuation. Examples of this type of error are provided in IV. A punctuation token on its own cannot function as information on the PIO elements of a publication. However, the model labels this positively. This misclassification can be attributed to the rarity of features for those tokens: the model only obtains information from the semantic features to classify. Punctuation often occurs in the positively annotated tokens and since the part-of-speechtag and embedding of this token is equivalent to those cases, it is being classified false positive. Parsing the dependency relations and including time-shifted information helped alleviate this problem over the course of the development of the models, but the issue was not resolved completely. A post-processing procedure could be formulated to treat such tokens and provide a more generous recall of labels. Adding dependency relations as a feature for the second model could solve this type of error. Removing punctuation would also be possible, although one would have to decide how to handle the time-series alignment of the data.

57	volunteers	0	0
58	.	0	1
59	Alfentanil	0	0

TABLE IV: First example of a misclassification of punctuation by the first model (prediction shown on the right).

36	.	0	0
37	Accordingly	0	0
38	,	0	0
39	the	0	0
40	authors	0	0
41	user	0	0
42	mood	0	1
43	inventories	0	1
44	and	0	1
45	psychomotor	0	1
46	tests	0	1
47	to	0	0

TABLE V: Example of a misclassification of a quote used in an abstract (prediction shown on the right).

Table **??** shows the misclassification of a citation as an intervention. The positively labelled sequence is indeed an intervention, but not of this biomedical publication. The abstract mentions another publication where this intervention is being proposed, causing a sequence to be misclassified. The error can also be explained by the scarceness of features. Adding dependency relations and coreference resolution to the model could solve this type of error because the tokens 'the authors' would not be linked to the authors of this medical publication. This approach was not within the scope of the project, but would be a valuable direction for future research.

Overall, the most frequent misclassification error is that only part of the tokens in a sequence are assigned, with gaps in between. This adjacency is poorly mapped by the Random Forest model, and the advantages of the CRF model in observing the assigned labels of its neighbours does not seem to solve this error. Further approaches to post-processing the results would need to be explored to make these modeling techniques effective for the PIO extraction task.

token	not_participants	participants	pred	true
in	0.94	0.06	not_part.	not_part.
patients	0.025	0.975	part.	part.
with	0.0175	0.9825	part.	part.
left	0.3	0.7	part.	part.
ventricular	0.5775	0.4225	not_part.	part.
dysfunction	0.135	0.865	part.	part.
.	0.43	0.57	part.	part.

TABLE VI: Misclassification of the token 'ventricular' in a sequence

VII. Conclusion & Further Work

The goal of the project, to explore the possibility of implementing a system for automatic annotation of medical text abstracts to be used by researchers, was partially fulfilled. The models were able to extract and learn syntactic and semantic relationships in the data, and learned to assign a label based on this. A simplified task, such as simply predicting the presence of some PIO label, would have been far easier to achieve but far less appropriate for the presented use case. However, the required precision, recall and F-score were not achieved to develop a fully-fledged application that researchers can access. Further research into modeling medical text is required to extract relationships accurately enough to be able to automate large parts of the meta-review process.

Further research into the topic of annotating medical text sequences should focus on exploring the different levels of information available in a time-series-aligned corpus. While bag-of-word features and embeddings may offer a generalized picture of the meaning of words, decoding local and semantic relationships continues to prove a difficult challenge in text sequence annotation. Not all models handle class imbalance elegantly, and the varying input dimensions and sequences that occur in a sentence-based annotated corpus can make it difficult to find the appropriate model for the task. Performing parameter tuning and cross-validation on models not able to extract enough information will yield little improvement in model performance. Furthermore, it is important that data be made available for researchers to access to further progress in meta-review, as there are few corpora with the required level of annotation and size to support this learning task.

A. Acknowledgements

The authors would like to thank Roser Morante for her guidance, assistance and input over the course of the project. She also set up the meeting with René Spijkers from the Cochrane Organisation, whom we would also like to thank for presenting to us.

Word count: 5260

References

[1] B. C. Wallace, J. Kuiper, and A. Sharma, "Extracting PICO Sentences from Clinical Trial Reports using Supervised Distant Supervision," p. 26, 2016.

[2] J. M. Kendall, "Designing a research project: randomised controlled trials and their principles," *Emergency Medicine Journal*, vol. 20, no. 2, pp. 164–168, Mar. 2003. [Online]. Available: http://emj.bmj.com/cgi/doi/10.1136/emj.20.2.164

[3] E. Hariton and J. J. Locascio, "Randomised controlled trials - the gold standard for effectiveness research: Study design: randomised controlled trials," *BJOG: An International Journal of Obstetrics & Gynaecology*, vol. 125, no. 13, pp. 1716–1716, Dec. 2018. [Online]. Available: http://doi.wiley.com/10.1111/1471-0528.15199

[4] A. Booth, G. Moore, K. Flemming, R. Garside, N. Rollins, Tunçalp, and J. Noyes, "Taking account of context in systematic reviews and guidelines considering a complexity perspective," *BMJ Global Health*, vol. 4, no. Suppl 1, p. e000840, Jan. 2019. [Online]. Available: http://gh.bmj.com/lookup/doi/10.1136/bmjgh-2018-000840

[5] "Cochrane Handbook for Systematic Reviews of Interventions," library Catalog: training.cochrane.org. [Online]. Available: /handbook/current

[6] R. Borah, A. W. Brown, P. L. Capers, and K. A. Kaiser, "Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the PROSPERO registry," *BMJ Open*, vol. 7, no. 2, Feb. 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5337708/

[7] I. J. Marshall and B. C. Wallace, "Toward systematic review automation: a practical guide to using machine learning tools in research synthesis," *Systematic Reviews*, vol. 8, no. 1, pp. 163, s13 643–019–1074–9, Dec. 2019. [Online]. Available: https://systematicreviewsjournal.biomedcentral.com/articles/10.1186/s13643-019-1074-9

[8] G. Karystianis, K. Thayer, M. Wolfe, and G. Tsafnat, "Evaluation of a rule-based method for epidemiological document classification towards the automation of systematic reviews," *Journal of Biomedical Informatics*, vol. 70, pp. 27–34, Jun. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1532046417300734

[9] D. Demner-Fushman and J. Lin, "Answering Clinical Questions with Knowledge-Based and Statistical Techniques," *Computational Linguistics*, vol. 33, no. 1, pp. 63–103, Mar. 2007. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/coli.2007.33.1.63

[10] M. Banko and E. Brill, "Scaling to very very large corpora for natural language disambiguation," in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*. Toulouse, France: Association for Computational Linguistics, 2001, pp. 26–33. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1073012.1073017

[11] I. Marshall, J. Kuiper, E. Banner, and B. C. Wallace, "Automating Biomedical Evidence Synthesis: RobotReviewer," in *Proceedings of ACL 2017, System Demonstrations*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 7–12. [Online]. Available: http://aclweb.org/anthology/P17-4002

[12] F. Boudin, L. Shi, and J.-Y. Nie, "Improving Medical Information Retrieval with PICO Element Detection," in *Advances in Information Retrieval*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, S. Rüger, and K. van Rijsbergen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 5993, pp. 50–61, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-12275-0_8

[13] A. for Computational Linguistics, Ed., *Proceedings of the 45th annual meeting of the Association for Computational Linguistics: June 23 - 30, 2007, Prague, Czech Republic. Main vol: ...* Stroudsburg, Pa: Association for Computational Linguistics, 2007, meeting Name: Annual meeting of the Association for Computational Linguistics (ACL) OCLC: 836762196.

[14] S. R. Jonnalagadda, P. Goyal, and M. D. Huffman, "Automating data extraction in systematic reviews: a systematic review," *Systematic Reviews*, vol. 4, no. 1, p. 78, Dec. 2015. [Online]. Available: https://systematicreviewsjournal.biomedcentral.com/articles/10.1186/s13643-015-0066-7

[15] B. Nye, J. J. Li, R. Patel, Y. Yang, I. J. Marshall, A. Nenkova, and B. C. Wallace, "A Corpus with Multi-Level Annotations of Patients, Interventions and Outcomes to Support Language Processing for Medical Literature," *arXiv:1806.04185 [cs]*, Jun. 2018, arXiv: 1806.04185. [Online]. Available: http://arxiv.org/abs/1806.04185

[16] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Jun. 2001. [Online]. Available: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1162&context=cis_papers

[17] Y. Zhang, Q. Chen, Z. Yang, H. Lin, and Z. Lu, "BioWordVec, improving biomedical word embeddings with subword information and MeSH," *Scientific Data*, vol. 6, no. 1, p. 52, Dec. 2019. [Online]. Available: http://www.nature.com/articles/s41597-019-0055-0

[18] I. Witten and I. Frank, "Data Mining - Practical Machine Learning Tools and Techniques," *Morgan Kaufmann*, vol. 31, Mar. 2002.

APPENDIX

Ranking score	Participant	Word
2.728720	0	par_token:variable
2.610971	0	TOKEN_LAG2:therefore
2.535022	0	token:correlated
2.451404	0	par_token:Investigators
2.419309	0	TOKEN_LAG1:i.e.
2.419309	0	LEMMA_LAG1:i.e.
2.398092	0	TOKEN_LAG1:process
2.365207	0	TOKEN_LAG-2:led
2.339631	0	token:STUDY
2.307379	0	par_token:composite
2.292098	0	LEMMA_LAG2:hz
2.242918	0	TOKEN_LAG2:tablet
2.205788	0	TOKEN_LAG-2:trend
2.184494	0	token:AND
2.183203	0	TOKEN_LAG2:Following
2.106117	0	stem:even
2.103827	0	token:SETTING
2.095540	0	par_token:supported
2.064383	1	par_lemma:TMJs
2.064383	1	par_token:TMJs
2.054122	0	token:reducing
2.019017	0	token:OBJECTIVE
2.014109	0	token:PURPOSE
1.989532	1	par_token:Excluded
1.985852	0	TOKEN_LAG1:Effect
1.975079	0	par_token:be
1.967942	0	token:BACKGROUND
1.933787	0	TOKEN_LAG1:Using
1.918957	0	LEMMA_LAG-2:formulation
1.917897	0	LEMMA_LAG-2:introduce

TABLE VII: Appendix A. Ranking score for participants for the CRF model

Ranking score	Intervention	Word
4.277560	0	token:INTRODUCTION
4.263292	1	token:PLAY
4.129590	1	token:TEN
3.665921	1	TOKEN_LAG-2:Adenosine
3.563603	1	TOKEN_LAG-1:Adenosine
3.410214	1	LEMMA_LAG-1:myoblast
3.323709	0	par_lemma:lot
3.316017	0	TOKEN_LAG-2:tumors
3.174562	0	LEMMA_LAG1:arbutamine
3.128649	1	TOKEN_LAG-2:Theories
3.125427	1	stem:pimecrolimus
3.104968	0	token:IE
3.086877	1	TOKEN_LAG2:B??
3.086877	1	LEMMA_LAG2:B??
3.005230	1	lemma:donate
2.982193	1	token:TAU
2.973336	1	TOKEN_LAG1:Adenosine
2.938343	0	TOKEN_LAG1:BID
2.929193	1	LEMMA_LAG-1:C/A
2.929193	1	TOKEN_LAG-1:C/A
2.921754	0	token:OBJECTIVE
2.858854	1	stem:bevacizumab
2.847914	0	par_token:SETTING
2.847159	1	stem:marimastat
2.826308	1	TOKEN_LAG2:neutral/menthol
2.826308	1	LEMMA_LAG2:neutral/menthol
2.824631	1	TOKEN_LAG2:PFS15
2.824631	1	LEMMA_LAG2:pfs15
2.818876	1	token:Restoring
2.811642	1	token:COMET

TABLE VIII: Appendix B. Ranking score for intervention for the CRF model

Ranking Score	Outcome	Word
4.277560	0	token:INTRODUCTION
4.263292	1	token:PLAY
4.129590	1	token:TEN
3.665921	1	TOKEN_LAG-2:Adenosine
3.563603	1	TOKEN_LAG-1:Adenosine
3.410214	1	LEMMA_LAG-1:myoblast
3.323709	0	par_lemma:lot
3.316017	0	TOKEN_LAG-2:tumors
3.174562	0	LEMMA_LAG1:arbutamine
3.128649	1	TOKEN_LAG-2:Theories
3.125427	1	stem:pimecrolimus
3.104968	0	token:IE
3.086877	1	TOKEN_LAG2:B??
3.086877	1	LEMMA_LAG2:B??
3.005230	1	lemma:donate
2.982193	1	token:TAU
2.973336	1	TOKEN_LAG1:Adenosine
2.938343	0	TOKEN_LAG1:BID
2.929193	1	LEMMA_LAG-1:C/A
2.929193	1	TOKEN_LAG-1:C/A
2.921754	0	token:OBJECTIVE
2.858854	1	stem:bevacizumab
2.847914	0	par_token:SETTING
2.847159	1	stem:marimastat
2.826308	1	LEMMA_LAG2:neutral/menthol
2.826308	1	LEMMA_LAG2:neutral/menthol
2.824631	1	TOKEN_LAG2:PFS15
2.824631	1	LEMMA_LAG2:pfs15
2.818876	1	token:Restoring
2.811642	1	token:COMET

TABLE IX: Ranking score

TABLE X: Appendix C. Ranking score for Outcome for the CRF model