JULIE Lab & Med Uni Graz @ TREC 2019 Precision Medicine Track

Erik Faessler
Jena University Language &
Information Engineering (JULIE) Lab
Friedrich-Schiller-Universität Jena
Jena, Germany
erik.faessler@uni-jena.de

Michel Oleynik
Institute for Medical Informatics,
Statistics and Documentation,
Medical University of Graz
Graz, Austria
michel.oleynik@stud.medunigraz.at

Udo Hahn
Jena University Language &
Information Engineering (JULIE) Lab
Friedrich-Schiller-Universität Jena
Jena, Germany
udo.hahn@uni-jena.de

ABSTRACT

The 2019 Precision Medicine Track at TREC (TREC-PM) aimed at identifying relevant documents from two collections, namely PubMed (biomedical abstracts) and ClinicalTrials.gov (clinical trials), given 40 precision medicine topics representing (virtual) patients. The organizers also proposed a new subtask on treatment retrieval from PubMed. We describe our contributions based on five runs for each task, including two runs for the treatment subtask using a naïve strategy. Our approach builds upon carefully designed weighted queries based on our experience from last year's participation and explores the usefulness of Learning to Rank (LETOR), trained on either the previous official gold standards or an internal reference standard for the topics chosen for the 2019 challenge. Our best results culminated in infNDCG = 0.5783, P@10 = 0.6525, and R-Prec = 0.3572 for the biomedical abstracts task and infNDCG = 0.6451, P@10 = 0.5474, and R-Prec = 0.4814 for the clinical trials task, obtained with a baseline retrieval strategy. LETOR worsened our results, especially when using the internal reference standard.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; *Content analysis and feature selection*; Retrieval effectiveness; Specialized information retrieval; • **Applied computing** → Health informatics.

KEYWORDS

information retrieval, precision medicine, search engine evaluation

1 INTRODUCTION

Driven by the decreasing costs of whole genome sequencing, the field of *precision medicine* has gained traction as a way to deliver optimal treatments for patients with specific biomarkers [2, 3, 5]. In this scenario, health professionals have to deal with an increasingly large amount of information available in scientific studies and clinical trials. In order to gain deeper insights into this poorly structured process, since 2017 the National Institute of Standards and Technology (NIST) has organized the TREC Precision Medicine (TREC-PM) challenge. TREC-PM aims at retrieving relevant documents from two collections, namely biomedical abstracts (BA) from PubMed and clinical trials (CT) from ClinicalTrials.org, given topics representing virtual patients (as an example, see Figure 1).

```
<topic number="1">
  <disease>melanoma</disease>
  <gene>BRAF (E586K)</gene>
  <demographic>64-year-old female</demographic>
</topic>
```

Figure 1: An example of a TREC-PM topic.

In 2019, TREC-PM for the first time did not only have topics exclusively about cancer, but additionally included ten topics on other health conditions such as "aortic aneurysm", "long QT syndrome", and "malignant hyperthermia". Furthermore, the organizers of TREC-PM also proposed a subtask on treatment ranking for BA with the goal of maximizing recall of possible treatments. Finally, a newer snapshot of PubMed and ClinicalTrials.org was provided, incorporating the previously distinct collections from the American Association for Cancer Research (AACR) and the American Society of Clinical Oncology (ASCO).

In this paper, we describe our participation at the TREC-PM 2019 challenge (team labeled "`julie-mug`"). In Section 2, we detail the strategies underlying our experimental framework [7, 9] (Section 2.1) in order to obtain baseline results (Section 2.2) and then introduce the construction of an internal reference standard (Section 2.3) that allowed us to experiment with LETOR directly on the 2019 TREC-PM topics (Section 2.4). We also describe our approach to treatment ranking (Section 2.5) and indicate improvements for clinical trials (Section 2.6). We finally present the results of our approach in Section 3 and discuss their limitations in Section 4. Section 5 summarizes our findings to foster future research.

2 METHODS

2.1 Experimental Framework

We built upon our previous work using the Free and Open-Source Software (FOSS) Java framework based on query templates and query decorators, described in detail in López-García et al. [7], Oleynik et al. [9]. We further expanded this framework to allow the incorporation of manually-defined terminologies to better handle TREC-PM-specific query expansions. This allowed us to accommodate to topics not related to cancer (which should not boost keywords such as "cancer" like we do for cancer-related topics), specifically map solid tumors, and define additional mappings not found in terminologies (such as "colon" ↔ "colorectal"). Our source code is publicly available at https://github.com/JULIELab/trec-pm.

Upon manual inspection, we added 13 new terms into our list of domain stop words, a step that caused substantial benefit in experiments with preliminary data. We also streamlined the process for internal gold standard construction (see Section 2.3) by automating the upload of experimental results to an online spreadsheet used for shared annotation efforts and the download of newly generated annotations from the respective sheet in the `.qrels` format.

We leveraged the Unstructured Information Management Architecture (UIMA) to read, process, and index both the biomedical articles and the clinical trial documents following our successful experiments in 2018 (team labeled "`hpi-dhc`") [9]. We enriched documents with gene mention annotations produced by the Banner gene tagger as offered by the `jcore-banner-ae-biomedical-english` component which is part of the JCoRe projects[1] component repository. The employed model was trained on data from the BioCreative II Gene Mention task.[2] We integrated the JeDIS [4] architecture to store the annotated documents in a PostgreSQL database and thus speed up document access for LETOR and creation of different development versions of the the ElasticSearch (ES) indices without the need to run Banner multiple times. We created the ES 5.4 indices with the JCoRe ElasticSearch Consumer.[3] Figure 2 gives an overview of our experimental setup.

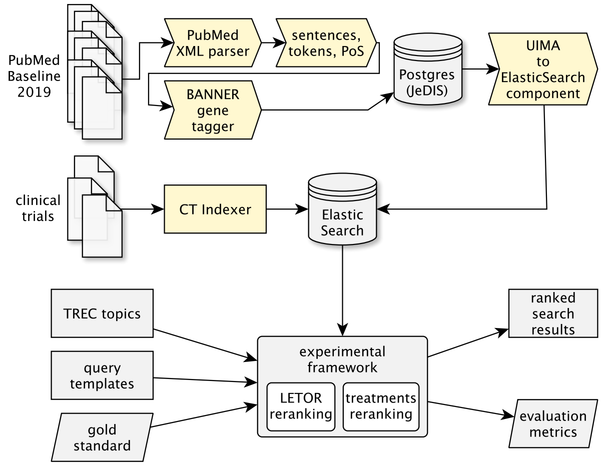

Figure 2: Overview of our experimental setup.

2.2 Baseline Retrieval Strategy

We here describe the baseline retrieval strategy for all of our runs.

Query structure. All queries created from the topics for document retrieval were ultimately formulated as ElasticSearch (ES) JSON queries and shared a common base structure. The main query for each topic consists of a compulsory clause that contains the disease and gene aspects of each topic. An optional clause adds general relevance signals to boost Precision Medicine (PM)-related documents and documents about cancer (for corresponding topics) that are described in [9]. Finally, a prohibitive clause matches documents on the term *non-melanoma* to reduce the number of false positives in our retrieval results.

Query expansion. We expanded the query topic fields *disease* and *gene* to boost the recall of our retrieval runs and feature creation for the LETOR approach. Following Oleynik et al. [9], such aspects were formulated as *dis_max* queries of subquery clauses. The *dis_max* clauses are comprised of the original topic term — the disease or gene name — and one additional clause for each query expansion element. For disease query expansion, we leveraged the Lexigram API.[4] We retrieved disease *preferred names* and *synonyms* and added them to separately weighted search clauses as described next. We also expanded gene symbols with the description and synonyms provided by the NCBI Gene database.[5]

Query boosting. A central element of our current and previous TREC-PM challenge contributions is the query clause weighting schema applied to the ES queries. The weights were chosen manually by experimenting on internal gold standard data and, for newer challenges, the official gold data from previous years. The most important query clauses — the disease and gene *dis_max* query parts — were boosted with a factor of 1.5 to elevate them above the optional relevance signals. The weighting and the specific query type of the nested disease and gene *dis_max* clauses also impact the final results. Table 1 depicts the exact values we used. Details about the query types can be found in the ES documentation.[6] We additionally downgraded documents with empty abstracts,[7] since their value for processing seemed to be be very limited.

Table 1: Weight values and query types for diseases and genes for both tasks.

		Biomedical Abstracts		Clinical Trials	
Expansion Type		Query Type	Weight	Query Type	Weight
Disease	Original	best_fields	1.0	phrase, slop=0	1.0
	Preferred	best_fields	0.1	phrase, slop=0	0.1
	Synonyms	phrase, slop=0	0.1	phrase, slop=0	0.1
Gene	Original	best_fields	1.0	best_fields	1.0
	Description	phrase, slop=10	0.1	phrase, slop=0	0.1
	Synonyms	phrase, slop=0	0.7	phrase, slop=0	0.1

Hand-crafted rules. For the clinical trials task, we further expanded topics with the corresponding gene family using a regular expression and a mapping for solid tumors as described by Oleynik et al. [9]. Additionally, we expanded corresponding topics with *colon* ↔ *colorectal* for both tasks.

2.3 Reference Standard

Since topics in 2019 differed from the previous TREC-PM editions, we created an internal gold standard to evaluate our experiments in the interim. Two annotators (a medical student and a co-author) assessed in total 454 biomedical abstracts and 403 clinical trials. Out of those, 172 abstracts and 65 trials were annotated by both of them, with an agreement rate (Cohen's kappa) of 75,22% (disagreement in 27 abstracts) and 80,90% (disagreement in 7 trials). We used the

[1] https://github.com/JULIELab/jcore-projects
[2] http://biocreative.sourceforge.net/biocreative_2_gm.html
[3] https://github.com/JULIELab/jcore-base/tree/master/jcore-elasticsearch-consumer
[4] https://www.lexigram.io
[5] https://www.ncbi.nlm.nih.gov/gene
[6] https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-multi-match-query.html#multi-match-types
[7] See, e.g., https://www.ncbi.nlm.nih.gov/pubmed/16521281.

internal gold standard to evaluate several experiments and also to train the LETOR algorithm (see Section 2.4).

2.4 Learning to Rank

We implemented a Learning to Rank (LETOR) approach [6] to rerank documents as an additional step after document retrieval (see Figure 3). Overall, we trained four LETOR models, two for the BA and CT task, respectively. For each task, one model was trained on our internal TREC-PM 2019 gold standard and the other on the union of the two previous official gold standards (from the 2017 and 2018 editions). We used the LambdaMART [1] implementation of RankLib[8] to train the LETOR models and rerank documents.

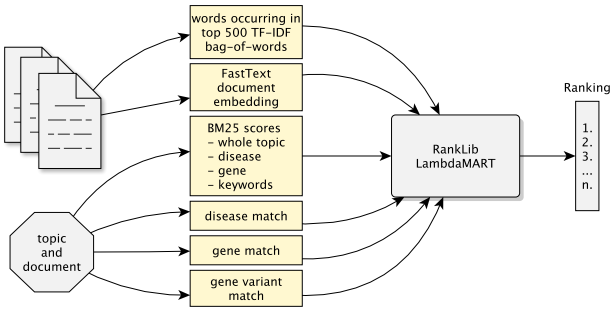
words occurring in top 500 TF-IDF bag-of-words
FastText document embedding
BM25 scores – whole topic – disease – gene – keywords
disease match
gene match
gene variant match
topic and document
RankLib LambdaMART
Ranking
1.
2.
3.
...
n.

Figure 3: Features used for training LETOR.

We explored four main classes of features:

Binary vocabulary features. We created binary features for the 500 words with highest TF-IDF from the gold standard of each task indicating the word presence.

Matches of the topic with the document. We created features reflecting how well a document matches the query topic by recording string matches of the topic's disease (with its synonyms and hypernyms), gene (with its synonyms), gene variant, and only the variant in the document. For each match type, we set the feature value to the number of matches, i.e., if the gene name of a topic matched three times, the feature "gene name match" was set to 3.

BM25 scores of the topic matched with document fields. We added features for different Okapi BM25 scores between the topic and the document. The calculated BM25 scores originated from: (1) the complete topic baseline query score, including disease, gene, their synonyms, and other relevance signals (as described in Section 2.2); (2) only the disease and its synonyms; (3) only the gene and its synonyms; (4) the optional relevance signal keywords also used in the baseline query.

FastText document embeddings of the document. Finally, we calculated the fastText document embedding [8] for the document.[9] The embeddings were trained on a PubMed subset containing gene mentions as identified by the Banner annotator described in Section 2.1 with a dimension of 300. All other parameters were left unset, resulting in the default settings of the fastText program.

2.5 Treatment Subtask

We participated in the treatment subtask using the officially provided treatment list extracted with MetaMapLite.[10] We filtered the list for the semantic types depicted in Table 2. Upon closer inspection, we noticed that several of the extracted concepts were either (a) not a real treatment (e.g., "duration", "basis", "medicine"), (b) not drug-related (e.g., "potassium", "yeast", "glucose"), or (c) amino acids (e.g., "leu", "leucine"). We thus experimented with filtering the treatments with a manually curated stop list of 230 entries. For each document of a topic result list, we removed treatments mentioned in higher-ranking positions (in order to maximize recall) and then, for each document, ranked remaining treatments by frequency (we kept only the top-3 most frequent). We lastly removed documents not matching any treatment.

Table 2: Semantic types used for treatment filtering.

Group	Code	Description
Procedures	T061	Therapeutic or Preventive Procedure
Chemicals & Drugs	T121	Pharmacologic Substance
Chemicals & Drugs	T200	Clinical Drug

In parallel, we also explored the existence of any valid treatment (whitelisted semantic type and not in the term stoplist) as a ranking signal during document retrieval, even if not a treatment run.

2.6 Clinical Trials Experiments

We experimented with two extra query variations for the CT task on top of the baseline retrieval strategy described in Section 2.2. First, in the run *jlctgenes*, we matched the topic gene not only with the document text, but also with a specific field filled only with gene names automatically extracted by the Banner gene tagger (see Section 2.1) following our previous successful experiments reported with biomedical abstracts. Second, in the run *jlctprec*, we refrained from matching all remaining documents to fill up the result list in order to improve precision.

3 RESULTS

The following list enumerates the elements used for the run names:

(1) jl: JULIE Lab,
(2) pm: PubMed,
(3) ct: Clinical Trials,
(4) tr: runs annotated with treatments,
(5) letor/ltr: learning to rank,[11]
(6) in: internal reference standard.

3.1 Biomedical Abstracts

Table 3 depicts the different strategies implemented for the five runs submitted for the BA task, as well as the corresponding results. Two runs included treatment annotations and therefore also include the corresponding metrics. Figure 4 depicts a visual overview of the official results, while detailed results per topic are shown in Figure 6 in the Appendix.

[8] https://sourceforge.net/p/lemur/wiki/RankLib/
[9] https://fasttext.cc/docs/en/crawl-vectors.html
[10] https://metamap.nlm.nih.gov/MetaMapLite.shtml
[11] Note that our officially submitted LETOR runs contained a code bug that prevented proper ranking. Meanwhile, this bug has been fixed and the updated results are provided here.

Table 3: Biomedical Articles: description and results of runs.

			jlpm		
Strategy	common2	letor	ltrin	trboost	trcommon
Baseline strategies	Y	Y	Y	Y	Y
Valid treatment exists	Y	N	N	Y	N
LETOR training data	-	2017/18	2019	-	-
treatments filter	Y	N	N	Y	N
# Treatments				**11,128**	9,443
Recall@10				**0.2857**	0.2698
F_1@10				**0.3118**	0.3019
Recall@25				**0.4603**	0.4469
F_1@25				**0.3793**	0.3716
infNDCG	**0.5783**	0.5740	0.2014	0.3876	0.3745
P@10	**0.6525**	0.6050	0.0650	0.5925	0.5775
R-Prec	**0.3572**	0.3527	0.1137	0.1639	0.1615

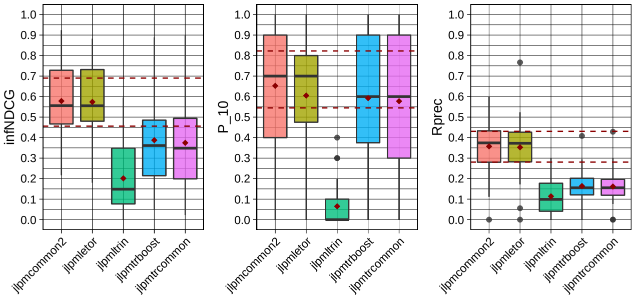

Figure 4: Biomedical abstracts: boxplots comparing our runs to the average best and median results.

The best performing run across all metrics was *jlpmcommon2*. This run closely resembles our top performing runs from last year with an additional check for existence of valid treatments as described in Section 2.5, including the treatment stop list.

The runs *jlpmtrboost* and *jlpmtrcommon* included treatment information and therefore documents were reranked as described in Section 2.5. While the former run is similar to the baseline run *jlpmcommon2*, the latter did not include a check for a valid treatment during retrieval using the stop list. The results also show that treatment re-ranking had a negative impact on the outcome of overall metrics, except for P@10, which reflects a smaller impact. Moreover, the additional check for a valid treatment during retrieval (in the run *jlpmtrboost*) improved not only overall metrics (e.g., +0.0150 P@10), but also treatment metrics (e.g., +0.0159 Recall@10). Additional experiments are required to test whether the same effect would be observed with a regular run.

Finally, the runs *jlpmletor* and *jlpmltrin* also used the baseline retrieval strategies to obtain documents from ES, on top of which LETOR was applied to re-rank documents. The LETOR model used for the *jlpmletor* run was trained on the union of TREC-PM 2017 and 2018 gold standards. The *jlpmltrin* run used a model trained on the internal reference data (see Section 2.3). The *jlpmletor* run performed similarly to *jlpmcommon2* albeit with a smaller variance across topic scores, while the *jlpmtrin* run exhibited the worst score of our BA runs.

3.2 Clinical Trials

Table 4 shows the retrieval and re-ranking features applied to the CT runs, as well as the official evaluation results. As described in Section 2.6, we experimented with small query variations and LETOR on top of the baseline retrieval strategy. Figure 5 compares CT results using boxplots across all topics, while detailed results per topic are shown in Figure 7 in the Appendix.

Table 4: Clinical Trials: description and results of runs.

			jlct		
Strategy	genes	letor	ltrin	phrase	prec
Baseline strategies	Y	Y	Y	Y	Y
Match all	Y	Y	Y	Y	N
Match extracted genes	Y	N	N	N	N
LETOR training data	-	2017/18	2019	-	-
infNDCG	**0.6451**	0.6243	0.2150	0.6244	0.6245
P@10	**0.5474**	0.5421	0.0553	**0.5474**	0.5447
R-Prec	0.4814	0.4605	0.1318	0.4799	**0.4820**

Run *jlctphrase* closely resembles our top performing run from last year and is thus considered our baseline here. It includes an exact (phrase) match on the disease topic for optimal precision and ranks best for P@10, in a tie with run *jlctgenes*.

The *jlctprec* run is similar to the above, but omitted a clause to retrieve all remaining documents (see Section 2.6). Even though the effect of this precision-optimization is minor, it is visible in the results as an increase of 0.0021 in R-Prec. Compared to the baseline, the *jlctgenes* run matches genes automatically extracted from text and obtained the best metrics across all runs.

The LETOR runs *jlctletor* and *jlctltrin* reveal decreased evaluation scores similarly to the BA task, in which the internal 2019 reference standard led to worse results than the union of previous official annotations.

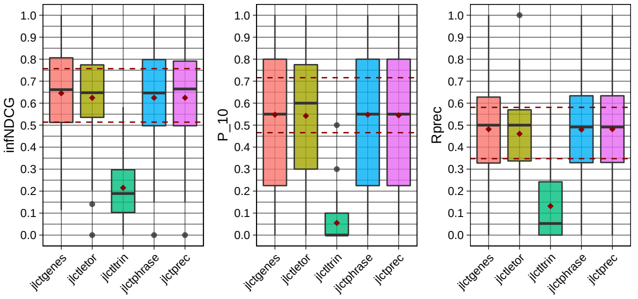

Figure 5: Clinical Trials: boxplots comparing our runs to the average best and median results.

4 DISCUSSION

Our optimal approaches described before led to the best metrics across all participating teams — except P@10 for CT, in which we ranked second. Nonetheless, future work is required to overcome some issues found, especially regarding the treatment subtask.

With respect to the treatment subtask, we would like to further explore techniques to refine the result list. Since only three treatments are accepted per document, we would like to better prioritize them using both a local and a global strategy. In a local context, we would like to explore syntactic features to, e.g., prioritize longer, more specific, candidates such as "monoclonal antibodies" instead of "adjuvant". Conversely, in a global context, we would like to try to optimize the result list using, e.g., LETOR methods to reorder the list. An automatic "treatment tagger" taking into account semantic information like word embeddings could be helpful to make the manual filtering of treatment terms obsolete. This would save manual labor and, hopefully, generalize to terms not in the list. We finally believe treatment runs could be further refined in ways different than regular runs, e.g., by boosting documents about treatment, cross-referencing DGIdb data on drugs,[12] and looking further down on the result list for potential matches.

Moreover, our LETOR approaches surprisingly decreased evaluation scores for both tasks. Since one LETOR feature is the BM25 score of the underlying run, this result comes completely unexpected. Future work is needed to measure how many documents need to be annotated so that an internal reference standard can produce better results than a baseline run.

5 CONCLUSION

In our previous appearance at TREC-PM, we showed that *dis_max* queries proved useful to expand queries without a drop in precision and successfully associated it with ranking signals related to precision medicine. Our current work further expanded that with Learning to Rank and a baseline strategy for treatment ranking, as well as additional minor query enhancements. Our LETOR approach surprisingly worsened all performance scores in both tasks, especially when using the internal reference standard.

In the biomedical abstracts task, checking for valid treatments during retrieval improved treatment runs, but it is unclear whether the same effect would be seen on regular runs. In the clinical trials task, matching all documents as a failover slightly worsened results, whereas matching extracted genes had a positive effect. The latter corroborates our conclusions from last year, where we showed a similar effect in the BA task. However, we still perform comparatively worse when evaluated by P@10 in this task, which opens possibilities for further experiments and improvements.

ACKNOWLEDGMENTS

We thank Juan Carlos Niño Rodriguez for the internal gold standard construction.

The contributions of the first and third author were funded by the German Bundesministerium für Bildung und Forschung (BMBF) under grant no. 01ZZ1803G.

REFERENCES

[1] Christopher J. C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: an overview.* Technical Report Technical Report MSR-TR-2010-82. Microsoft Research.

[2] Francis S Collins and Harold Varmus. 2015. A new initiative on precision medicine. *New England Journal of Medicine* 372, 9 (2015), 793–795.

[3] U.S. National Research Council. 2011. *Toward Precision Medicine: Building a Knowledge Network for Biomedical Research and a New Taxonomy of Disease.* The National Academies Press, Washington, DC. https://doi.org/10.17226/13284

[4] Erik Faessler and Udo Hahn. 2018. Annotation data mmanagement with JeDIS. In *DocEng 2018 — Proceedings of the 18th ACM Symposium on Document Engineering. Halifax, Nova Scotia, Canada, August 28-31, 2018.* Association for Computing Machinery (ACM), New York/NY, 4pp.

[5] Lewis J Frey, Elmer V Bernstam, and Joshua C Denny. 2016. Precision Medicine Informatics.

[6] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[7] Pablo López-García, Michel Oleynik, Zdenko Kasáč, and Stefan Schulz. 2017. TREC 2017 Precision Medicine - Medical University of Graz. In *TREC 2017 — Proceedings of the 26th Text REtrieval Conference. Gaithersburg, Maryland, USA, November 15-17, 2017 (NIST Special Publication).* National Institute of Standards and Technology (NIST), Gaithersburg/MD, 12pp.

[8] Tomáš Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *LREC 2018 — Proceedings of the 11th International Conference on Language Resources and Evaluation. Miyazaki, Japan, May 7-12, 2018*, Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan E. J. M. Odijk, Stelios Piperidis, and Takenobu Tokunaga (Eds.). European Language Resources Association (ELRA), Paris, 52–55.

[9] Michel Oleynik, Erik Faessler, Ariane Morassi Sasso, Arpita Kappattanavar, Benjamin Bergner, Harry Freitas da Cruz, Jan-Philipp Sachs, Suparno Datta, and Erwin Böttinger. 2018. HPI-DHC at TREC 2018 Precision Medicine Track. In *TREC 2018 — Proceedings of the 27h Text REtrieval Conference. Gaithersburg, Maryland, USA, November 14-16, 2018 (NIST Special Publication).* National Institute of Standards and Technology (NIST), Gaithersburg/MD, 9pp.

[12] http://www.dgidb.org/

A RESULTS PER TOPIC

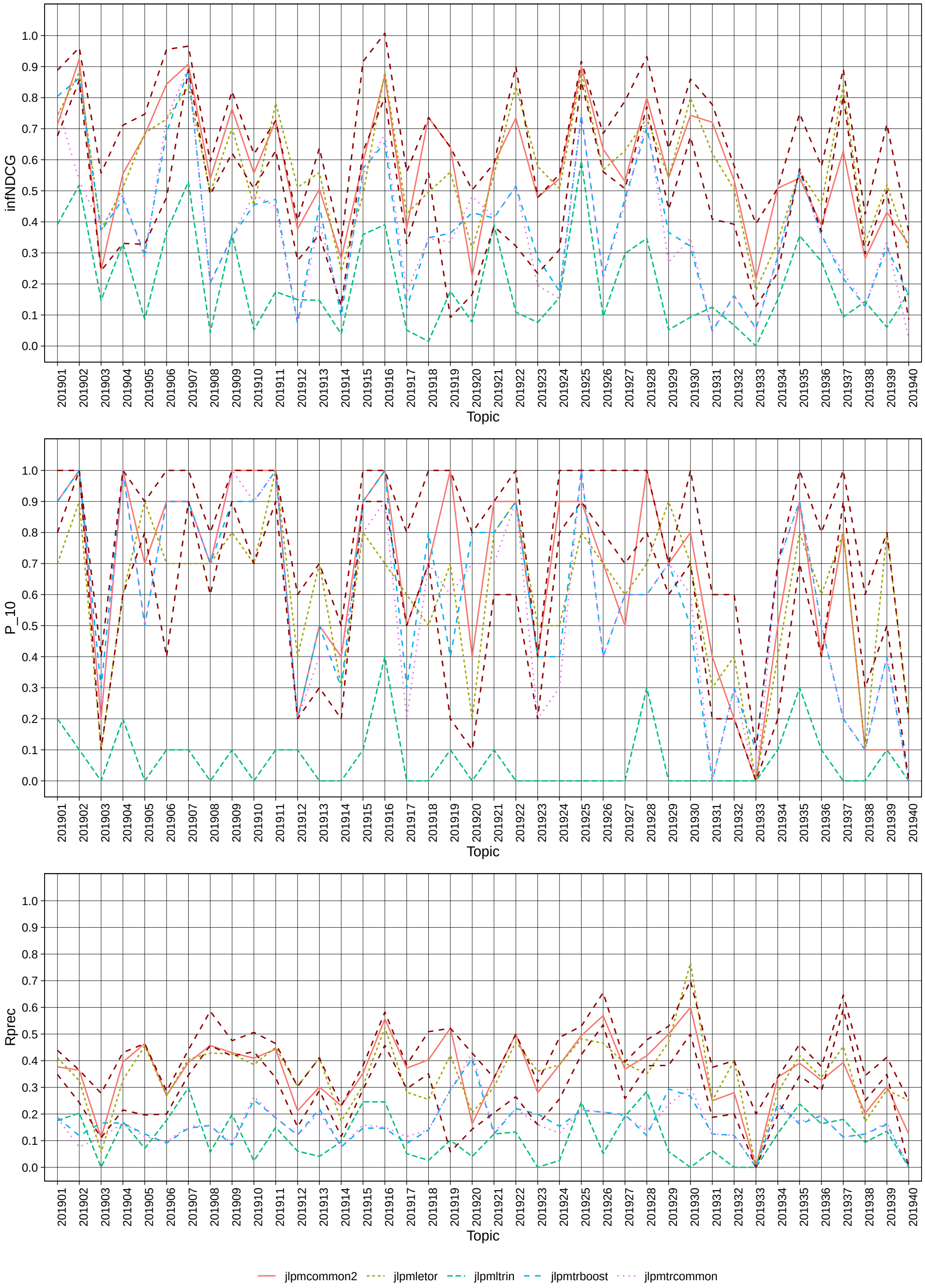

Figure 6: Biomedical Abstracts: metrics per topic for the submitted runs.

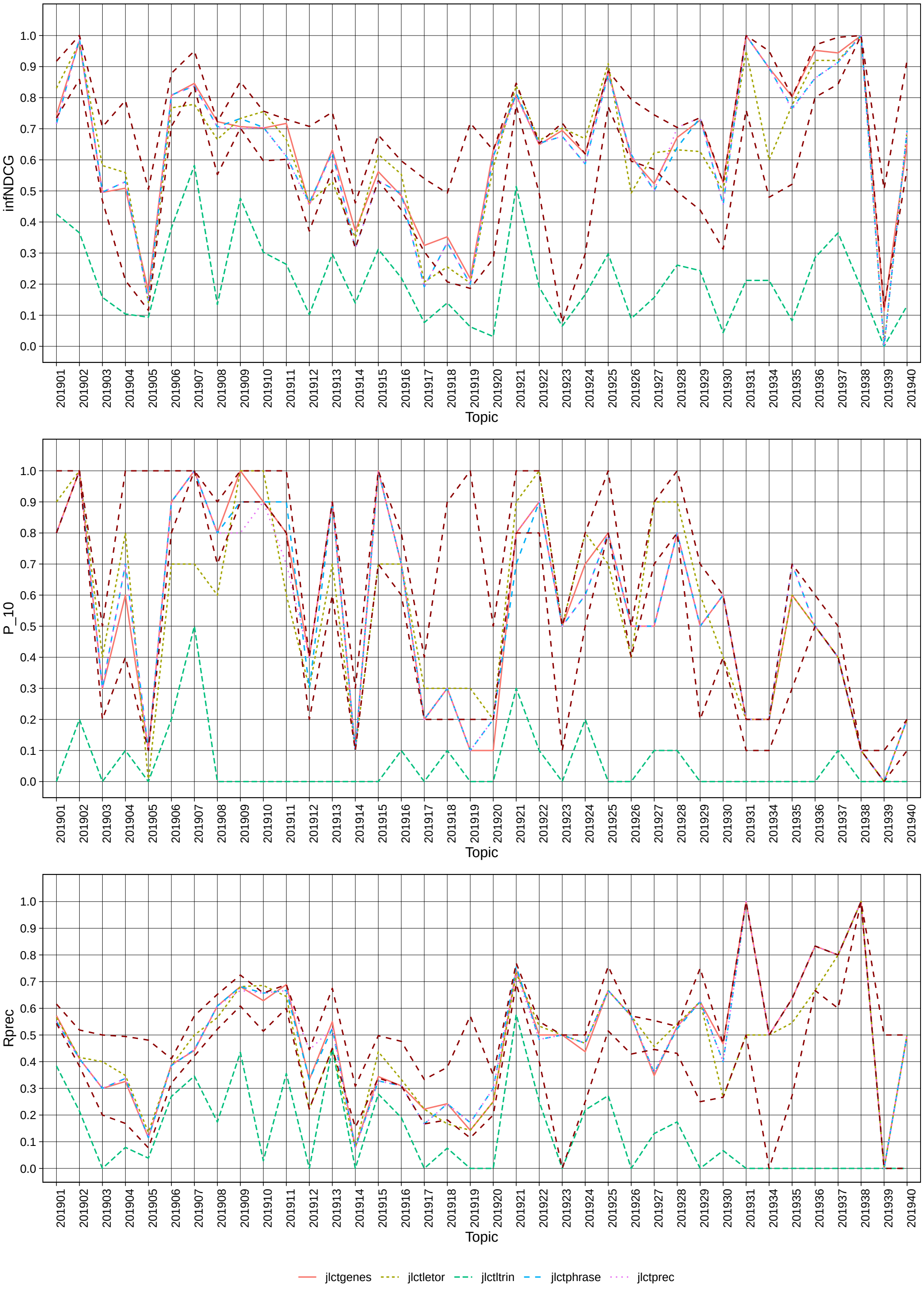

Figure 7: Clinical Trials: metrics per topic for the submitted runs.