RSA DSc on TREC's 2018 Precision Medicine Track

Alexandros Bampoulidis(✉) and Mihai Lupu

Research Studios Austria, Studio Data Science
Vienna, Austria
{name.surname}@researchstudio.at

Abstract. In this paper, we describe our approach to TREC's 2018 Precision Medicine challenge. We describe how we developed a system that semantically enriches the text documents and the disease part of the topic and issues extensive and detailed boolean queries to the Information Retrieval system and we present its results.

1 Introduction

The TREC 2018 Precision Medicine (PM) track challenge, as in 2017 [1], is to retrieve the most relevant documents from a collection of literature articles' (LAs) abstracts and clinical trials' (CTs) descriptions, given a patient's form of cancer, demographic and genomic information.

Each document collection (LAs abstracts and CTs descriptions) corresponds to a subtask, although the topics that are to be queried to the Information Retrieval (IR) system are common for both. The LAs collection consists of 27 million abstracts from MEDLINE and the CTs collection consists of 241 thousand descriptions from ClinicalTrials.gov. Both collections are in XML format and each document includes at least a title.

The track defines two degrees of relevance: definite and partial. Both specify that a document's demographic must match the topic's one. Definite relevance specifies that the document's discussed form of cancer is the topic's exact or more specific form of cancer and that the document's discussed gene(s) match at least one of the topic's genes. Partial relevance specifies the same things as the definite one, except that the type of cancer can be of a more general form and the discussed gene(s) can be missing a variant or have a different variant of the gene.

2 Approach

In this section, we describe our approach to TREC's 2018 PM challenge. Specifically: how we preprocessed and indexed the data, how we processed the topics and retrieved the documents and, finally, the details of our submitted runs. Note that the whole process is automated.

2.1 Preprocessing

We defined two types of text corresponding to importance: *primary* and *secondary*. We concatenated the text of specific XML elements of the provided data to either *primary* or *secondary* and, then, annotated them with a GATE annotation pipeline that was especially developed within the KConnect project[1].

Figure 1 depicts how the annotation pipeline works: Given a text, it annotates terms with a class (Anatomy, Disease, Drug or Investigation) and a UMLS Concept Unique Identifier (CUI)[2]. Using this annotation pipeline we extracted all the *Disease CUIs* from the primary and secondary texts and concatenated them to the fields *primary_text_annotations* and *secondary_text_annotations*, respectively, with the CUIs being separated by a semicolon.

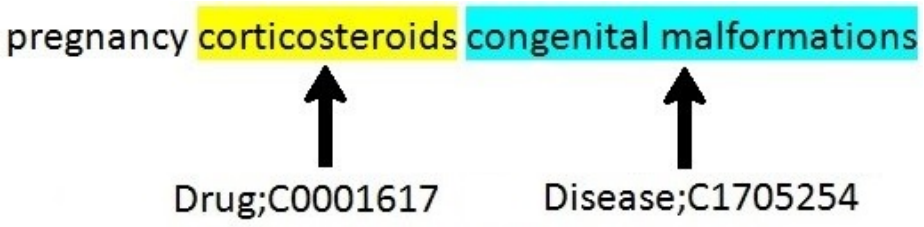

Fig. 1: GATE annotation pipeline

The primary text of the LAs consists only of the title and the secondary text consists only of the abstract. The primary text of the CTs consists of the elements: brief_title, official_title, condition, keyword and mesh_term. The secondary text of the CTs consists of the elements: brief_summary, detailed_description and arm_group_label. Additionally, we extracted the information from the gender, minimum_age and maximum_age elements of the CTs.

2.2 Indexing

After the preprocessing step, we indexed the two collections into elasticsearch[3] with the following fields:

- docid
- primary_text
- primary_text_annotations
- secondary_text
- secondary_text_annotations
- gender (CTs only)
- minimum_age (CTs only)
- maximum_age (CTs only)

[1] Horizon 2020 research and innovation programme, grant agreement No. 644753
[2] https://www.nlm.nih.gov/research/umls/new_users/online_learning/Meta_005.html
[3] https://www.elastic.co

2.3 Retrieval

Each topic consists of 3 fields: disease, gene and demographic (Fig. 2).

```
−<topic number="5">
    <disease>melanoma</disease>
    <gene>BRAF (V600E), PTEN loss of function</gene>
    <demographic>57-year-old male</demographic>
  </topic>
−<topic number="6">
    <disease>melanoma</disease>
    <gene>BRAF (V600E), NRAS (Q61R)</gene>
    <demographic>67-year-old male</demographic>
  </topic>
```

Fig. 2: TREC 2018 PM track topics

Disease We applied the GATE annotation pipeline to the disease and extracted its CUI. Then, we created three lists of CUIs: exact, more specific and more general. These lists contain disease CUIs that are related to the topic's one, as retrieved from NCBI's MedGen MGREL database[4] (Fig. 3). Specifically:

- Exact
  - *RELA* in *has_alias*, *alias_of*
- More specific
  - *RELA* in *has_alias*, *alias_of*
  - *RELA* = *isa*
  - *RELA* = "" and *REL* in *PAR*, *CHD*
- More general
  - *RELA* in *has_alias*, *alias_of*
  - *RELA* = *inverse_isa*

Gene We created two lists of genes: exact and missing/different variant. The former contains the text as it is specified in the topic, while the latter contains only the genes (e.g. BRAF, PTEN and NRAS in Fig. 2).

[4] ftp://ftp.ncbi.nlm.nih.gov/pub/medgen/MGREL.RRF.gz

	CUI1	AUI1	STYPE1	REL	CUI2	AUI2	RELA
	C0699790	Filter	Filter	Filter	Filter	Filter	Filter
1	C0699790	A11933211	AUI	PAR	C0027651	A11965596	
2	C0699790	A11948183	AUI	PAR	C0027651	A11965596	
3	C0699790	AN0163863	AUI	CHD	CN029768	AN0081272	
4	C0699790	A7591371	SCUI	RO	C0005859	A7569165	disease_may_have_associated_disease
5	C0699790	A7591371	SCUI	RO	C1332442	A20244448	disease_may_have_associated_disease
6	C0699790	A7591371	SCUI	RO	C1708349	A10806850	disease_may_have_associated_disease
7	C0699790	A24581942	AUI	RQ	C1527249	A12035993	has_alias
8	C0699790	A11933211	AUI	RO	C0032580	A20267552	has_manifestation
9	C0699790	A11948183	AUI	RO	C1321489	A12005602	has_manifestation
10	C0699790	A11948183	AUI	RO	C1835398	A12005824	has_manifestation
11	C0699790	A11948183	AUI	RO	C2676137	A23784221	has_manifestation
12	C0699790	A7591371	SCUI	PAR	C0007102	A7605209	inverse_isa
13	C0699790	A7591371	SCUI	PAR	C0009402	A7569694	inverse_isa
14	C0699790	A7591371	SCUI	RO	C1707292	A10800262	is_finding_of_disease
15	C0699790	A7591371	SCUI	RO	C1707933	A10804126	is_finding_of_disease
16	C0699790	A7591371	SCUI	RO	C1332480	A7625956	is_not_finding_of_disease
17	C0699790	A7591371	SCUI	CHD	C0149640	A7634696	isa

Fig. 3: NCBI's MedGen MGREL database

Demographic In the case of the CTs, we extracted the gender and the age of the patient with simple string processing.

After extracting all the information and creating the lists of disease CUIs and genes, we created all the possible query types of the form:

$$(gender\hat{}age\hat{})*\_text\_anno{:}disease\_*\hat{}*\_text{:}gene\_*,$$

where

- gender (CTs only): either male or female
- age (CTs only): topic's age between minimum_age and maximum_age
- *_text_anno: either primary_text_anno (pr) or secondary_text_anno (se)
- disease_*: either disease_exact (ex), disease_specific (sp) or disease_general (ge) list of CUIs
- *_text: either primary_text (pr) or secondary_text (se)
- gene_*: either gene_exact (ex) or gene_missing_different_variant (md) list of genes

Then, we created different rankings of the query types that were to be issued to the index (in total 34 query types, including the case of no disease CUIs and no genes). After conducting extensive experiments on the collections with different rankings and using the challenge's 2017 topics, we submitted the best performing ones shown at Tables 1 and 2. The queries were issued in the order displayed at Tables 1 and 2 until the IR system has retrieved 1000 documents. Each retrieved document that was not retrieved by the preceding queries was stacked in a list and was scored from 1.1 (1st retrieved document) decreasing

by 0.001 until the 1000th document was retrieved. The retrieved documents of each query were ranked by elasticsearch's default ranking system.

A simple example of this procedure: If *pr_ex_pr_ex* retrieves documents (A, B, C), then the list of stacked documents would be [(A, 1.1), (B, 1.099), (C, 1.098)]. Then, if *pr_sp_pr_ex* retrieves documents (B, E, C, D, A), then the list of stacked documents would be [(A, 1.1), (B, 1.099), (C, 1.098), (E, 1.097), (D, 1.096)], and so on.

Literature Articles Runs				
Run 1	Run 2	Run 3	Run 4	Run 5
pr_ex_pr_ex	pr_ex_pr_ex	pr_ex_pr_ex	pr_ex_pr_ex	pr_ex_pr_ex
pr_sp_pr_ex	pr_sp_pr_ex	pr_sp_pr_ex	pr_sp_pr_ex	pr_sp_pr_ex
pr_ge_pr_ex	pr_ge_pr_ex	pr_ge_pr_ex	pr_ge_pr_ex	pr_ge_pr_ex
pr_ex_pr_md	pr_ex_se_ex	pr_ex_pr_md	pr_ex_se_ex	pr_ex_se_ex
pr_sp_pr_md	pr_sp_se_ex	pr_sp_pr_md	pr_sp_se_ex	pr_sp_se_ex
pr_ge_pr_md	pr_ge_se_ex	pr_ge_pr_md	pr_ge_se_ex	pr_ge_se_ex
pr_ex_se_ex	se_ex_se_ex	pr_ex_se_ex	pr_ex_pr_md	pr_ex_pr_md
pr_sp_se_ex	se_sp_se_ex	pr_sp_se_ex	pr_sp_pr_md	pr_sp_pr_md
pr_ge_se_ex	se_ge_se_ex	pr_ge_se_ex	pr_ge_pr_md	pr_ge_pr_md
pr_ex_se_md	pr_ex_pr_md	se_ex_se_ex	se_ex_se_ex	pr_ex_pr_md
pr_sp_se_md	pr_sp_pr_md	se_sp_se_ex	se_sp_se_ex	pr_sp_pr_md
pr_ge_se_md	pr_ge_pr_md	se_ge_se_ex	se_ge_se_ex	pr_ge_pr_md

Table 1: Rank of the first 12 queries for the LAs runs. The first four letters refer to the disease CUIs part of the query and the last four letter refer to the genes part of the query.

3 Results

The performance of our runs is presented in Table 3. There are minor differences in performance across the runs, with run 5 of both LAs and CTs performing the best in most of the challenge's evaluation metrics. Note that despite the fact that the first 3 queries issued to the index are the same across all runs, they do not retrieve more than 5 documents in all topics, as it is evident from Table 3.

4 Conclusion

In this paper, we describe our approach to TREC's 2018 Precision Medicine challenge. Our approach consists of splitting the text into two categories corresponding to importance, semantically enriching the documents and the disease

Clinical Trials Runs				
Run 1	Run 2	Run 3	Run 4	Run 5
pr_ex_pr_ex	pr_ex_pr_ex	pr_ex_pr_ex	pr_ex_pr_ex	pr_ex_pr_ex
pr_sp_pr_ex	pr_sp_pr_ex	pr_sp_pr_ex	pr_sp_pr_ex	pr_sp_pr_ex
pr_ge_pr_ex	pr_ge_pr_ex	pr_ge_pr_ex	pr_ge_pr_ex	pr_ge_pr_ex
pr_ex_pr_md	pr_ex_pr_md	pr_ex_pr_md	pr_ex_se_ex	pr_ex_se_ex
pr_sp_pr_md	pr_sp_pr_md	pr_sp_pr_md	pr_sp_se_ex	pr_sp_se_ex
pr_ge_pr_md	pr_ge_pr_md	pr_ge_pr_md	pr_ge_se_ex	pr_ge_se_ex
se_ex_pr_ex	se_ex_se_ex	pr_ex_se_ex	se_ex_se_ex	pr_ex_pr_md
se_sp_pr_ex	se_sp_se_ex	pr_sp_se_ex	se_sp_se_ex	pr_sp_pr_md
se_ge_pr_ex	se_ge_se_ex	pr_ge_se_ex	se_ge_se_ex	pr_ge_pr_md
se_ex_pr_md	se_ex_se_md	pr_ex_se_md	pr_ex_pr_md	se_ex_se_ex
se_sp_pr_md	se_sp_se_md	pr_sp_se_md	pr_sp_pr_md	se_sp_se_ex
se_ge_pr_md	se_ge_se_md	pr_ge_se_md	pr_ge_pr_md	se_ge_se_ex

Table 2: Rank of the first 12 queries for the CTs runs. The first four letters refer to the disease CUIs part of the query and the last four letter refer to the genes part of the query.

part of the topic, issuing multiple detailed queries to the IR system, stacking the retrieved documents and assigning them a symbolic score.

References

1. Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, William R. Hersh, Steven Bedrick, Alexander J. Lazar, and Shubham Pant. Overview of the trec 2017 precision medicine track. 2017.

LAs/CTs Results					
Run	iNDCG	Rprec	P@5	P@10	P@15
LAs Run 1	0.4568	0.2862	0.5640	0.5440	0.4960
LAs Run 2	0.4709	0.2916	0.6160	0.5780	0.5267
LAs Run 3	0.4467	0.2850	0.5640	0.5440	0.4960
LAs Run 4	0.4755	0.2937	0.6200	0.5780	0.5213
LAs Run 5	0.4855	0.2949	0.6200	0.5780	0.5213
CTs Run 1	0.4691	0.3706	0.6040	0.5440	0.4720
CTs Run 2	0.4713	0.3673	0.6040	0.5460	0.4760
CTs Run 3	0.4710	0.3700	0.6040	0.5480	0.4760
CTs Run 4	0.4729	0.3704	0.5960	0.5420	0.4787
CTs Run 5	0.4743	0.3721	0.6040	0.5460	0.4853

Table 3: Inferred NDCG, R-prec, P@5, P@10 and P@15 of our submitted runs.