ETH Zurich at TREC Precision Medicine 2017

Negar Foroutan Eghlidi, Jannick Griner, Nicolas Mesot,
Leandro von Werra and Carsten Eickhoff
Department of Computer Science
ETH Zurich, Switzerland
c.eickhoff@acm.org

Abstract

This paper describes ETH Zurich's submission to the TREC 2017 Precision Medicine (PM) track. We begin by performing literal query term matching, taking into account the likelihood of document relevance in terms of cancer types, genetic variants, and demographics. In a second, subsequent stage, we re-rank the most relevant results based on a range of deep neural gene embeddings that project literal genetic expressions into a semantics-preserving vector space using feed-forward networks trained on PubMed and NCBI information but also relying on generative adversarial methods to determine the likelihood of co-occurrence of various mutations within the same patient/article. Empirical results show that even without existing expert labels, the proposed method can introduce marginal improvements over competitive tf-idf baselines.

1 Introduction

Precision medicine is a modern field of study that aims to use genomic information in finding more effective treatments for patients. Due to the popularity of the new paradigm, the volume of annually published scholarly precision medicine articles has been growing rapidly in recent years. While this considerable amount of scientific research holds a rich and ever increasing well of knowledge, its sheer scale makes it intractable for manual inspection and mandates the development of dedicated automatic retrieval facilities.

In this paper, we present a modular patient-centric information retrieval system for use in precision oncology settings. Based on patient demographics as well as information regarding the type of tumor and its genetic composition, we rank scholarly articles as well as clinical trials with respect to their relevance for a range of reference patients.

Our ranking scheme combines evidence from four sources of information: (1) literal presence of keywords in the description of patients, cancer types and genetic variations. (2) dense semantics-preserving vector representations of key biomedical terms (3) functional neural embeddings of gene expressions and their patient context (4) generative adversarial models of gene mutation co-occurrence. After an initial retrieval run evidence for the usefulness of expansion candidates is aggregated across our various sources, resulting in a significantly more robust ranking of resources.

Lacking historic training data, in this first edition of the TREC Precision Medicine track, we manually annotated a sample of 1800 (1200 abstracts and 600 clinical trials) query-document pairs in order to train and optimize any supervised representation learning and fusion schemes. While these labels were created without direct involvement of domain experts, we see significant performance improvements over training-free exact matching methods.

26th Text Retrieval Conference (TREC) Notebook 2017.

2 Methodology

Our ranking scheme is centrally based on exact keyword matching (Section 2.1) and subsequently further refined by three neural-network components that estimate a range of compatibility scores between patient and document (Sections 2.2 - 2.4). Finally, we apply a number of score fusion methods in order to re-rank the original exact matching results (Section 2.5).

2.1 Term-based Exact Matching

We use patient descriptions (topics) to generate queries that we compare to documents to obtain a similarity score. The theoretical framework adopted for representing both the abstracts and trials is the vector space model as described by Salton *et al.* [7] where term weights corresponf to tf-idf scores. For the practical implementation, we use the open-source framework Lucene[1]. In order to account for varying field importance in structured topics, we apply static weighting factors as detailed in Table1. To further improve recall, we perform term-wise query expansion off all genome identifiers by appending their various synonyms as specified by the HUGO Gene Nomenclature Committee (HGNC)[2].

For scientific abstracts, we divide age-specific query terms into three categories (pediatric, adult and geriatric medicine). For each of these categories, we add the age indication terms proposed by Kaster *et al.* [5]. Clinical trial runs are enhanced by performing exact age-range matching on the age range provided by the clinical trial descriptions.

Table 1: Empirically optimal field weights.

Term	Scientific Abstracts	Clinical Trials
Gender	0.05	1
Age	0.05	1
Condition	1	1
Genes	1	0.2

2.2 Neural Context Embedding

In our second approach, we go beyond literal term matching by computing semantic vector representations of gene identifiers in patient descriptions and abstracts. We construct a vocabulary of gene symbols, biomedical concepts and frequent words observed in documents. We then use a neural probabilistic language model according to [6], to find continuous embeddings of these words. The similarity of a patient and a scientific abstract or clinical trial document can now be expressed as a distance function applied to their respective vector representations.

To identify relevant gene identifiers in the document collection, we again consult the HGNC list of gene symbols and their synonyms and apply simple heuristics based on symbol-synonym relations to eliminate ambiguous gene symbols such as CO2 or SARS that also have non-genomic interpretations. The biomedical concepts for our vocabulary are given by the Unified Medical Language System (UMLS)[3]. We train our embeddings on all abstracts found in PubMed[4] containing any of the identified relevant gene symbols.

In a first step we replace all UMLS expressions in the abstracts by their UMLS concept identifiers and apply the CBOW and skip-gram Word2Vec models to find continuous vector representations of all words in the vocabulary. To optimize model parameters, we evaluate the resulting embeddings based on the cosine distances between gene symbol synonym pairs found in the HGNC gene list. We find the best performance for a skip-gram model, using a vocabulary of 150'000 words and a small window size of 1 word before and after the target word and an embedding dimensionality of 250.

[1] https://lucene.apache.org/

[2] https://www.genenames.org/

[3] https://www.nlm.nih.gov/research/umls/

[4] https://www.ncbi.nlm.nih.gov/pubmed/

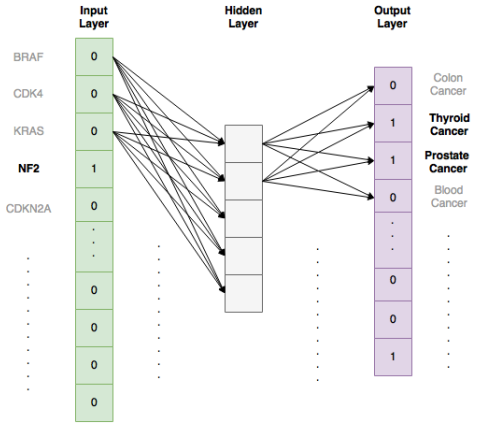

Figure 1: Feed-forward network architecture for functional gene embeddings.

Relevance scores for each topic-article pair are calculated based on their vector representations and a context-based similarity score s_c. We consider the three most similar words (most proximal embeddings) $d_{i,1}, d_{i,2}, d_{i,3}$ for each gene symbol g_t in a patient's description and give them ordinally decreasing importance weights w_j. We apply the commonly used term $1+log(k_{ij})$ for term frequency $tf(d_{ij})$ to account for multiple occurrences of the term d_{ij} in the document. The final score is the inverted sum of cosine distances, normalized per topic for later fusion (see Section 2.5).

$$s_c(T, D) = \sum_{g_t \in T} \sum_{j=1}^{3} w_j * (1 + log(tf(d_{ij}))) * (1 - cos(g_t, d_{ij}))$$

2.3 Neural Gene Embedding

In an alternative neural embedding scheme, we aim to model bio-medical properties of genes and mutations trying to establish a Euclidean space in which functionally similar genes are projected to proximal locations. This functional similarity is defined based on the common cancer types that both genes are associated with, making genes that are related to the same cancer types more similar.

We use the human genome portion of the National Center for Biotechnology Information (NCBI) dataset [1] to establish. For each gene, this collection includes chromosomal localization, gene products and their attributes (*e.g.*, protein interactions), associated markers, phenotypes, and interactions. Additionally, for each recorded gene in this dataset, there is a "*summary text*" field, which contains a summary of all the information about that gene. In this summary, all the cancer types a gene is associated with (if any) are listed. We select all genes that contain at least one human cancer type in their summary text. This procedure results in a total of 1347 distinct genes and 109 cancer types.

We employ a single-layer feed-forward neural network to embed the genes into a Euclidean space. The input of this network is a one-hot vector that uniquely represents a gene. The output of the network is a vector in which each component corresponds to a specific cancer type. If an input gene is associated with a cancer type, the corresponding element in the output vector indicating that cancer type is set to one. Figure 1 shows the architecture of our neural network.

The input layer size is equal to the number of genes (1347) while the output layer corresponds to the number of cancer types (109). The hidden layer size is chosen experimentally (50). The maximum cross-validation accuracy of predicting cancer types is 60%. After training the network with these characteristics, we use the gene representation to rank the scholarly articles. For each topic-document pair, we calculate the overall functional relevancy score s_f as the aggregated cosine similarity between all genes g_t mentioned in the topic and all genes g_d mentioned in the document. We consider three aggregation functions $a = \{mean, max, mean\ max\}$

$$s_f(T, D) = a(cos(g_t, g_d)) \forall g_t \in T, g_d \in D$$

2.4 Adversarial Training for Gene Pair Ranking

In order to measure the compatibility of a patient's gene mutations with mutation mentions in a document, we use a generative adversarial net (GAN). A GAN consists of two networks trained with respect to different objectives: the generator network aims at copying the original data distribution, whereas the discriminator network is fed real and generated data alternately and is tasked to identify the real data point. In optimal settings, the generator's ability to output pseudo-realistic data increases in parallel to the discriminator's power to distinguish real from generated data. For the TREC Precision Medicine Challenge, we use the GAN's discriminator part to determine, if a patient's gene mutations are compatible with a gene mentioned in a document.

We use the co-occurrence of gene mutations in PubMed abstracts, provided by the TREC precision medicine track, as training data [8]. The underlying assumption using this approach is, that genes often occurring together also share similarities. We filter 22M abstracts for 614 genes known to be in connection with the formation of cancer according to COSMIC [3]. For each abstract, we extract pairs of mutated genes from the list of mentioned mutations (*i.e.*, a document with 4 mutations yields 6 pairs). In total, we retrieve 21k unique gene pairs for further use as GAN training data.

While GANs demonstrate impressive results when generating images, it is widely known that they suffer from poorly defined gradients at the generator/discriminator interface when tasked with discrete variables, such as two-hot vectors representing gene pairs [4]. We use an approach proposed by Choi *et al.*, named medGAN [2]. This architecture utilizes the decoder part of a pre-trained autoencoder to translate the real valued generator output vector to a discrete vector. In order to prevent mode-collapse, the generator produces a batch of gene pairs, which are then separately fed to the discriminator along with the average vector of the batch. However, this leaves the discriminator with the possibility to distinguish solely on the basis of the average batch vector, which is neither available nor relevant when looking at a single gene pair. To tackle this issue, we introduce a second training mode which runs in parallel to the batch training. We call it single mode, providing the discriminator with a zero vector instead of the mean batch vector. Using this architecture, we train the GAN with the gene pairs in the form of two-hot vectors as training data.

Finally, we use the trained GAN to construct a metric to rank documents according to their gene compatibility. For each patient gene g_t and gene mention in a given document g_d, we calculate the discriminator value for $Disc(g_t, g_d)$. For likely real pairs the discriminator outputs 1 and 0 for likely unrelated data, respectively. We combine the values of $Disc(.)$ across all gene mentions in a document by applying an aggregation function $a = \{min, max, mean\}$.

$$s_g(T, D) = a(Disc(g_t, g_d)) \forall g_t \in T, g_d \in D$$

2.5 Score Fusion

The previously described constituent scores (tfidf(.), $s_c(.)$, $s_f(.)$ and $s_g(.)$) are fused in 5 different ways based on our manually annotated topic-document pairs. To ensure computational feasibility, the scores of all neural approaches ($s_c(.)$, $s_f(.)$ and $s_g(.)$) are based on the 1000 top-ranked documents for each topic retrieved by term-based exact matching. In a first step, we combine all constituent scores into a feature vector v, which has 8 components explained in detail in Table 2.

Table 2: Feature vector v description.

Entry	Method	Value
v_1	Term-based Exact Matching	$tfidf(T, D)$
v_2	Neural Context Embedding	$s_c(T, D)$
v_3	Functional Gene Embedding	$mean(s_f(T, D))$
v_4	Functional Gene Embedding	$max(s_f(T, D))$
v_5	Functional Gene Embedding	$meanmax(s_f(T, D))$
v_6	GAN Gene Pair Ranking	$mean(s_g(T, D))$
v_7	GAN Gene Pair Ranking	$min(s_g(T, D))$
v_8	GAN Gene Pair Ranking	$max(s_g(T, D))$

We present five ways to fuse the score vector into a single value further used to rank the list of documents. These methods of fusing scores make up the runs submitted for the TREC Precision Medicine Track for both abstracts and trials.

1. Weighted Sum (nDCG) This method is based on the weighted sum of each score vector. The list of documents is ranked according to the weighted sum and evaluated using the nDCG metric. We optimize mixture weights for ideal nDCG on our manually annotated sample.

2. Pure Term-based Exact Matching This is technically not a run fusion, since it only takes into account the scores from Term-based Exact Matching. We chose this approach to comply with the TREC PM 2017 guidelines, which state that exact matches are required in a document to constitute relevance.

3. Neural Fusion The neural approach trains a feed-forward neural network with one fully connected hidden layer (16 neurons) to learn ranking the documents based on the combined score vector. The network is trained to map the input vector onto 1 if the underlying document is relevant and 0 otherwise. Again, our annotated documents serve as ground truth data. We use dropout to avoid over fitting, which is likely due to the limited amount of training data available.

4. Weighted Sum (qrel) In contrast to the nDCG method, this approach aims at mapping the score vector onto annotated document labels via linear regression.

5. Neural Embeddings and GAN Values Exclusively The last method is identical to the 4th fusion method with the exception of not considering any *tfidf* scores. As previously mentioned, the documents are already pre-selected by the exact matching method and this fusion therefore tests if such pre-selection of exact terms suffices to rank documents with neural embedding scores and GAN valuations alone.

We present a summary of the fusion process and the resulting unofficial nDCG scores in Table 3. We can observe that the weighted sum optimized for nDCG score performs marginally better than the pure exact matching score. This is likely due to the bias towards the exact matching score in the annotated documents due to the annotation of only the best exact matching documents.

Table 3: nDCG scores after score fusion.

Fusion Method	**Type**	**nDCG**
Weighted Sum (nDCG)	abs.	0.6642
Exact Matching	abs.	0.6605
Neural Fusion	abs.	0.5382
Weighted Sum (qrel)	abs.	0.4770
Exclusive Gene	abs.	0.2927
Weighted Sum (nDCG)	trial	0.9654
Exact Matching	trial	0.9647
Neural Fusion	trial	0.8316
Weighted Sum (qrel)	trial	0.9187
Exclusive Gene	trial	0.4024

3 Results

We submitted ten official TREC Precision Medicine 2017 runs for evaluation, five for scientific abstract retrieval and five for clinical trial retrieval. Table 4 lists the official performance of each run in terms of inferred nDCG, precision at 10 retrieved documents and R-precision. At this time, no official relevance judgments have been released.

For both tasks, literal matching represents a strong baseline that can only be marginally improved by including functional and contextual information. The exclusion of explicit matching information results in significant performance drops. We believe that this is due to the particular judging criteria of this track that require exact matches in terms of gene expressions and cancer types in order to

Table 4: Experimental results.

Fusion Method	Run Identifier	Type	infnDCG	P@10	R-Prec
Weighted Sum (nDCG)	ws	abs.	0.2663	0.3414	0.1941
Term-based	luc	abs.	0.2632	0.3379	0.1895
Neural Fusion	nn	abs.	0.2437	0.3034	0.1660
Weighted Sum (qrels)	wsq	abs.	0.2450	0.3034	0.1627
Neural + GAN	gws	abs.	0.1453	0.1517	0.1066
Weighted Sum (nDCG)	ws	trial	?	0.0571	0.0439
Term-based	luc	trial	?	0.0607	0.0453
Neural Fusion	nn	trial	?	0.0000	0.0019
Weighted Sum (qrels)	wsb	trial	?	0.0286	0.0257
Neural + GAN	gwsq	trial	?	0.0036	0.0035

constitute relevance. We can further observe that heavily parametrized methods such as neural fusion, due to lack of training data do not gain a competitive advantage over straight-forward linear fusion methods.

4 Conclusion

In this paper, we provide an overview of ETH Zurich's contribution to the TREC 2017 Precision Medicine track. Using a wide array of neural network approaches, we aim to capture contextual, functional and pairwise genetic information expressed in topics and documents. While immediate performance gains were rather limited, we see considerable potential for future improvement when using this year's official relevance judgments to train supervised models.

5 Acknowledgements

This research is funded by the Swiss National Science Foundation (SNSF) Ambizione Program under grant agreement no. 174025.

References

[1] Gene [internet]. bethesda (md): National library of medicine (us), national center for biotechnology information, 2004. [Cited 2017 Oct 15]. Available from: https://www.ncbi.nlm.nih.gov/gene/.

[2] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete electronic health records using generative adversarial networks. *arXiv preprint arXiv:1703.06490*, 2017.

[3] Simon A. Forbes, David Beare, Harry Boutselakis, Sally Bamford, Nidhi Bindal, John Tate, Charlotte G. Cole, Sari Ward, Elisabeth Dawson, Laura Ponting, Raymund Stefancsik, Bhavana Harsha, Chai Yin Kok, Mingming Jia, Harry Jubb, Zbyslaw Sondka, Sam Thompson, Tisham De, and Peter J. Campbell. Cosmic: somatic cancer genetics at high-resolution. *Nucleic Acids Research*, 45(D1):D777–D783, 2017.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] M. Kastner, W. Nancy, W. Cindy, M. Kathleen Ann, and H. Brian. Age-specific search strategies for medline. *J Med Internet Res*, 8(4).

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

[7] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[8] TREC. Precision medicine track, 2017. Available at: http://www.trec-cds.org/2017.html [Online; accessed 2017 Oct 18].