Deep Learning Approach for the Precision Medicine Track

Juan Pablo Consuegra-Ayala[1,3], Giovanni Stilo[2,4], Alessandro Celi[2,5], Amleto Di Salle[2,6]

[1] *University of Havana, School of Math and Computer Science*
[2] *University of L'Aquila, Dep. of Information Engineering, Computer Science and Mathematics*
[3] jpconsuegra@matcom.uh.cu [4] giovanni.stilo@univaq.it [5] alessandro.celi@univaq.it [6] amleto.disalle@univaq.it

The paper describes the system presented by the University of L'Aquila in collaboration with the University of Havana - team named UNIVAQ - to the TREC 2019 Precision Medicine Track. The proposed solution, maps any kind of documents - Scientific Abstract, Clinical trials, and Topics - into a multi-dimensional common general representation. Each document is described by five primitive features. The values of each feature are extracted from the original documents using deep learning and machine learning text processing based techniques. To recognize Genes and Diseases, we have trained our models using the PubTator annotated corpus. Instead, to derive demographics information, we have trained the employed deep learning models using the documents -obtained from the Relevance and Raw judgements of the past edition of TREC Precision Medicine / Clinical Decision Support Track 2018- considered "relevant" or "partially relevant". The results of the Track clearly show that applying a system (as our) made solely by a tagging based approach to the Precision Medicine task, is not sufficient to achieve the performances gained by other systems presented in the TREC Precision Medicine Track 2019.

1 Introduction

The precision medicine track of the 2019 REtrieval Conference (TREC) addresses the challenge of helps doctors. The track is oriented to retrieve useful information regarding the treatment of several cancer diseases and their patients. Precision Medicine (PM)is the approach that customizes healthcare by taking into account the characteristics of each person, such as genes, environment, and lifestyle. Note that the traditional approach formulates treatments and prevention strategies by taking into account solely cases that are popular. The Precision Medicine approach, allows doctors and researchers to more accurately predict the treatments that are more effective for a particular group of people or individuals. However, the vast literature available in medicine makes it difficult to find quickly the optimal treatment for the considered patient. The NIST in the TREC Precision Medicine track promotes a challenge to better understand problems related to this domain, and technologies that help doctors to retrieve the medical information, more effectively.

The 2019 TREC Precision Medicine track -as an extension of the last edition- ask participants to rank a predefined collection of documents according to a set of *Topics* (queries). Two collections of documents are provided in the PM Track. The first collection is composed by *Scientific Abstract*, which could be relevant to identify patient's treatments, and the second one contains *Clinical Trials*, for which the patient may be eligible.

In this paper, we describe the system proposed by the University of L'Aquila in collaboration with the University of Havana for the 2019 TREC Precision Medicine track. The proposed solution is composed of the following steps:

1. first, we map Scientific Abstract, Clinical trials and Topics into a common general representation;
2. then, each document is scored against the query using the proposed matching model (based on five primitive scores);
3. the five scores - for each document - are combined together to produce a final score;

4. documents are finally ranked according to their final score.

The paper is organized as follows. Section 2 describes the common general representation of documents and topics. Section 3 presents how it is possible to transform each type of document into its general representation. Section 4 shows how to score each document against the topic and how to produce the final ranking. Finally, Section 6 summarise the acquired knowledge, during the participation to the Precision Medicine Track.

2 General Representation

In this section, we describe the general representation used for all documents, i.e. Topics, Clinical Trials and Scientific Abstracts. The representation is composed by five distinctive features: *genes* and *diseases*, *gender* and *age* information, and *precision medicine (PM)*. As general approach, each features is expressed by a score that tells how much, the considered value, is present/discussed in the document.

The first two features of the representation are the *genes* and *diseases*. The *genes* feature is a g-dimensional vector, where g corresponds to the size of the vocabulary of genes founded across the whole collection of documents. Analogously, the *diseases* feature is a d-dimensional vector, where d is the size of the diseases' vocabulary. Each component of those vectors holds the estimated probability of the given document to discuss the corresponding term (disease or genes).

Then the vocabulary of genes and diseases is mapped to a semantic vector space. To do so, first we built a graph, based on the DisGeNet (Piñero et al., 2019) dataset. The gene and disease are the nodes of the graph. And there is an edge between a pair of nodes if exist an association among them in the DisGeNet[1] (the dataset holds directly gene-disease associations). Then we have decided to use `node2vec` (Grover and Leskovec, 2016) to embeds the nodes of a graph into a 64-dimensional space. [2]

The third feature of the representation is the *gender* information.The *gender* feature is a 2-dimensional vector. Each component of this vector corresponds to the estimated probability of the given document to be related to *male* and *female* population.

The fourth feature of the representation is *age* information. The *age* feature is an a-dimensional vector, where a corresponds to the number of possible age spans found across the whole collection of documents. Each age span has the form of <t_s, t_e> where t_s and t_e, respectively stand for starting and ending age. Each component of the vector holds the estimated probability of the given document to be related with the corresponding age span.

The last feature is the *precision medicine (PM)* score. Documents provided in the Track can be classified into one of the following categories:

Human PM: The abstract/trial (1) relates to humans, (2) involves some form of cancer, (3) focuses on treatment, prevention, or prognosis of cancer, and (4) relates in some way to at least one of the genes in the topic.

Animal PM: Identical to Human PM requirements (2)-(4), except for animal research.

Not PM: Everything else. An example includes "basic science" that focuses on understanding underlying genomic principles (e.g., pathways), but provides no evidence for treatment.

We decide to consider relevant for the Track only Human PM and Animal PM. The *PM* feature of the general representation holds a scalar value that represents the estimated probability of the given document to be related to precision medicine, either Human PM or Animal PM.

Table 1 summarise the features of the general representation. The presented representation of table 1 corresponds to a sample Topic as described in Section 3.

Table 1: *General representation of a sample Topic*

```
-------------------------------------------
  <topic number-"1">
    <disease>melanoma</disease>
    <gene>BRAF (V600E)</gene>
    <demographic>64-year-old male</dem>
  </topic>
-------------------------------------------
```

Feature			Value			
Diseases	ds_0	...	*melanoma*		...	ds_d
$p_{ds}(d_i)$	0	...	1		...	0
Genes	g_0	...	*BRAF*	*V600E*	...	g_g
$p_g(d_i)$	0	...	1	1	...	0
Gender			*male*	*female*		
$p_g(d_i)$			1	0		
Age	a_0	...	`<64-64 years>`		...	a_a
$p_a(d_i)$	0	...	1		...	0
PM			*Human/Animal PM*			
$p_{pm}(d_i)$			1			

3 Document Tagging

In this section, we describe how each type of document is transformed into the general representation; i.e. we explain how we extract each feature from every document. In some cases, the features are explicitly stated

[1] www.disgenet.org

[2] The implementation used was freely available at https://github.com/eliorc/node2vec. For the training phase, we have generated for each node, 100 random walks of length 30, and then, we used a window size equal to 7.

in the document and a simple set of extraction rules are sufficient to get them. In other cases, the features must be inferred from the document using text processing techniques that we will explain later on.

The *Topics* documents (queries) explicitly state all the features. Therefore, the features extracting process is straight forward for them. On the other hand, the *Clinical Trials* and *Scientific Abstracts* documents do not explicitly state all the features. Consequently, we decided to use a set of text processing techniques to extract the features needed. The procedures used to extract information, from Clinical Trials and Scientific Abstracts, are similar but not identical. Genes and Diseases information is extracted using the same model but applied on different textual fields. Demographics information is handled in an entirely different manner between the different types of documents.

The following subsections describe in the details how these features are extracted from each type of document.

3.1 Genes and Diseases

In this subsection, we describe how we have extracted the *genes* and *diseases* features from the different types of documents - Topics, Clinical Trials and Scientific Abstracts.

Diseases and genes information, for the Topics documents, as previously stated, can be easily extracted. The disease is explicitly available through the field `<disease>DISEASE</disease>`. The corresponding value in the *diseases* feature vector , of the extracted disease, is then set to 1 . Diseases might contain multiple words. Then, for each possible non-empty sub-selection of words, the corresponding value in feature vector is set to a number $v \in [0, 1]$ that is proportional to the number of words. The gene and its variant is available in the Topic with the format `<gene>GENE (VARIANT)</gene>`. [3] In the same way, the values that belong to genes and its variants are set to 1 in the *genes* feature vector.

The other types of documents provided in the Track do not explicitly state genes and diseases. But genes and diseases can be extracted from the text fields - e.g. title, and abstract.

For Scientific Abstracts documents, genes and diseases are extracted from the `title` and the `abstract` fields. In the case of Clinical Trials, the annotations are derived from the text parts `title` and `eligibility-criteria`. The `eligibility-criteria` is a textual field divided into two sub-parts: *Inclusion Criteria* and *Exclusion Criteria*. Since the primary intention of the retrieving trials is to address relevant Clinical Trials for which the patients are eligible, we decide to extract diseases and genes solely from the *Inclusion Criteria* sub-part.

Two deep learning models were employed to extracts information about diseases and genes from the plain text. Both models share the same architecture but are trained on different tasks. The proposed architecture has been previously successfully applied in the literature to extract entities from plain text (Piad-Morffis et al., 2019; Mederos-Alvarado et al., 2019). We used an annotated corpus provided by *PubTator* as a training set (Wei, Kao, and Lu, 2013). Hereafter, we depict the used architecture and we will provide details of the training phase.

3.1.1 Model Architecture

The models used to extracts diseases and genes want to solve a sequence labelling problem. Given a sequence of words, the model predicts the best label to be assigned to each word. Genes and diseases can be multi-word so the models use a labelling system that allows to performs this encoding in one iteration. Labels used by our system are the follows: *[B]egin* to denote the beginning of an annotation; *[M]iddle* to denote the continuation of an annotation; *[E]nd* to denote the end of an annotation; *[W]ord* to report a single word annotation; and *[O]ther* to indicate that is not possible to annotate the considered word.

Figure 1 summarise the architecture of the proposed model and shows an example of its application on a single sentence. The model is made of five stacked layers connected in a sequential manner. The first layer - used as look-up table *(A)* - transform the characters of each word into an embedding vector. Then, a Bi-LSTM layer (Hochreiter and Schmidhuber, 1997) process the sequence of embedded characters in order to produce a single representation of the whole word *(B)*. The sequence of the words - sentence - is then, processed by another Bi-LSTM layer, that produces the local sentence representation *(C)*. Finally, a fully-connected layer *(D)* with softmax activation, predicts the most probable label (*BMEWO*) that can be assigned to each word.

The sequence of labels(*BMEWO*) built by each model, is then used to identify the named genes or the list of diseases, present in each document.

3.1.2 Training Details for Gene and Diseases

In order to train the presented model we used an annotated corpus provided by PubTator. The corpus is composed by a collection of $28\,581\,465$ articles from *PubMed* (contains title and abstract). For each article, a list of annotated entities, and their span in the text, is given. Entities belongs to one of the following classes: `Gene`, `Chemical`, `Disease`, `Species` and `Mutation`. For our purpose, only the `Disease`, and the `Gene` entities were used to train each model.

Since there is a frequent overlap between gene and disease in the *PubTator* corpus, we decide to train two individually models instead of just a single one. Furthermore, since genes and diseases do not share the

[3] The previous format, due to some syntax error, is not always consistent, so a set of hand-crafted rules were taken into account to produce the list of genes.

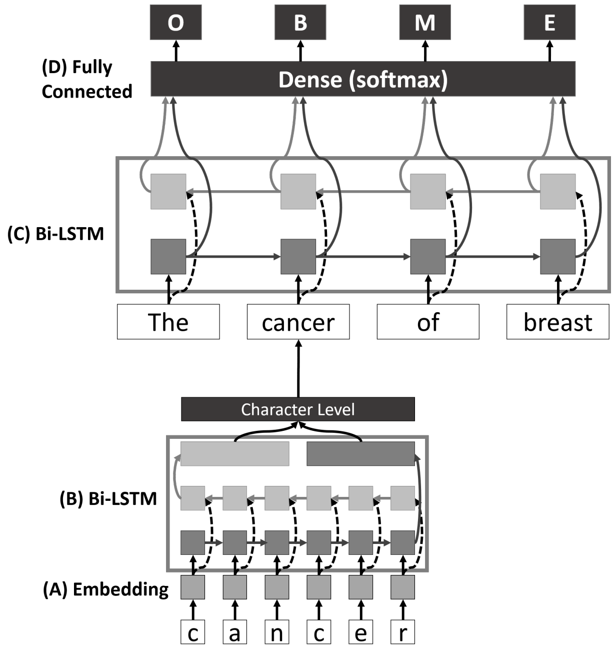

Figure 1: *Sample architecture of the gene and disease extraction models.*

same morphology, it makes sense to think that the features that are relevant for detecting one class might not necessarily be the same for detecting the other class. The intuition is that, two individual models might achieve better performances. Note that even if, the gene-gene and disease-disease overlapping are not handled by our models, there are few occurrences of this issue in the corpus.

For replicability purpose, we will give some additional details of the adopted architecture. The maximum length of each word is set to 15 characters. Words with less than 15 chars are padded to the right. Words with more than 15 chars are truncated.

A k-fold cross-validation was performed using k = 5 on both models (gene and disease). We used 10 000 randomly sampled documents in the training phase. The gene extraction model achieve a mean accuracy of 97.93% with a standard deviation of 0.08%. The disease extraction model achieve a mean accuracy of 96.17% with standard deviation of 0.04%.

3.2 Demographics

In this section, we will give details about the extraction of the demographics.

For Topics, the gender and the age can be extracted directly from the original document. The gender and age are directly available in the field `<demographic> AGE-year-old GENDER </demographic>`. The corresponding value of the gender discovered in the document is the set, in the 2-dimensional *gender*, to 1. Similarly, the value of the corresponding *age* span discovered in the document is set to 1.

For the Clinical Trials documents, the demographics are already available in the following fields: `eligibility`: (1) `gender`, (2) `minimum_age`, and (3) `maximum_age`. As before, the corresponding *gender* value of the 2-dimensional vector is set to 1.

According to the `<minimum_age, maximum_age>` the corresponding *age* span is enabled using the 1 value. Note that, minimum and maximum age is not always reported in the documents using the year value, but also days or months values were used, then, it was necessary to use a standardization process in order to have a common representation.

Instead, Scientific Abstracts do not explicitly state their demographics information. Therefore, to deduce them, a deep learning models were employed to classify the document into a fixed set of demographic classes. Especially, two models were trained to estimate the gender (male, female), and four others to estimate the age span (infant, adult, etc.). To simplify the demographic information, we have chosen to use a set of 6 age classes identified in *Proteccion social: Ciclo de Vida*. Table 2 summarise the adopted ages span for each class.

Table 2: *Ages span of each demographics' class*

Class	Age Span	
Early Childhood	0 to 5	years
Childhood	6 to 11	years
Adolescence	12 to 18	years
Youth	19 to 26	years
Adulthood	27 to 59	years
Seniors	60 to ∞^+	years

3.2.1 Model Architecture

Given a sequence of words, the model predict if the document matches one of the corresponding gender or age category. The input text is built from the concatenation of the `title` and `abstract` of each document. Two embeddings based representations - character level and word-level - are used as input of the model. For word-level representation, we have used a pre-trained Glove (Pennington, Socher, and Manning, 2014) embedding of dimension 100.

Figure 2 summarise the architecture of the demographic models and shows the application of the model on a sample text sequence. All models are made of a two-branch; each branch is composed by a stack of layers. One branch computes the character level representation of the words through a embedding lookup table and a Bi-LSTM layer for character level encoding *(A-B)*. The other branch computes the word-level representation of each word through a pre-trained word embedding layer *(C)*. The outputs of the two branches are concatenated to obtain the final word representation.

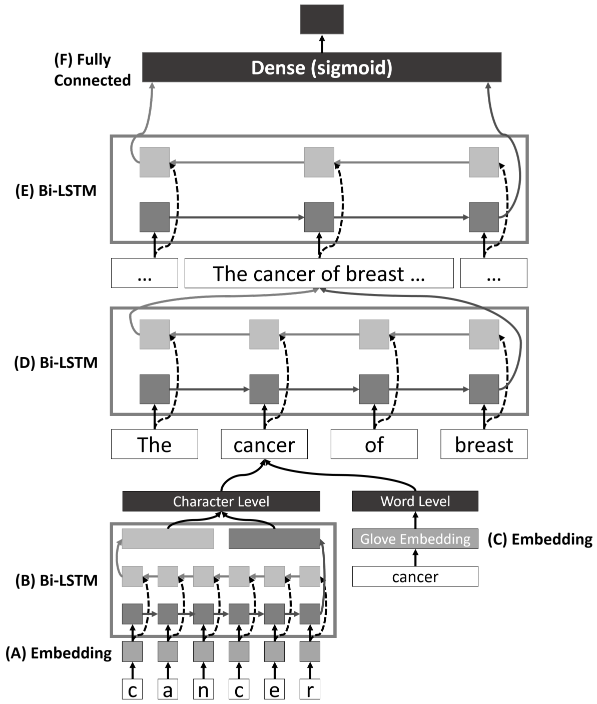

Figure 2: *Sample architecture of the demographic and PM models.*

A Bi-LSTM layer process the sequence of word representations to compute a single vector that encodes the whole sentence *(D)*. Another Bi-LSTM layer process the sequence of sentence representations to compute a single vector that encodes the whole document *(E)*. Finally, a fully-connected layer with *sigmoid* activation estimates the probability of the given document to belong to the corresponding demographic class *(F)*.

All layers, except the last one, are shared among the models and are trained jointly.

3.2.2 Model Training

To train the classifiers presented above, we used the Relevance and the Raw judgements from the past edition of *TREC Precision Medicine / Clinical Decision Support Track - 2018*. To build a training set, we used those documents evaluated as *partially relevant* and *definitely relevant* in the Relevance judgements. We obtain 5 552 positive examples, and we consider 678 documents from the Raw judgements, as negative examples. Unfortunately, only four over six considered age classes, were found in the training set. Moreover, to mitigate the unbalanced nature of the dataset, we employed a class based weighting scheme in the training phase. The weighting scheme gives different weights to the model's loss according to the cardinality of each class. The complete statistics for each class (age/gender) - divided into positive and negative instances - are reported in table 3.

Table 3: *Statistics of the training set used for Age and Gender identification*

Age class	Total	Positives	Negatives
Early Childhood	361	304	57
Childhood	0	0	0
Adolescence	286	211	75
Youth	0	0	0
Adulthood	2 529	2 266	263
Seniors	3 054	2 771	283
Total	6 230	5 552	678

Gender	Total	Positives	Negatives
Female	2 601	2 308	293
Male	3 629	3 244	385
Total	6 230	5 552	678

For replicability purpose, we will give also some additional details of the adopted architecture. The maximum number of characters the model allows per word are 15. Words with less than 15 chars are padded to the right. Words with more than 15 chars are truncated. The number of words per sentence is independently setted up to the highest number of words among all the sentences of each document. Sentences with fewer words, are padded to the right.

3.3 Precision Medicine (PM)

In this section, we describe how to compute the *PM* score for the different types of documents, i.e. Topics, Clinical Trials and Scientific Abstracts. All the Topics demands to retrieve Precision Medicine documents . Therefore, the *PM* feature of every Topic is always set to 1.

In the case of Clinical Trials and Scientific Abstract, two independent models were trained to predict the probability of a given document to be relate with the precision medicine task. Both models use the same architecture described in Section 3.2.1 and Figure 2. To train the models we used the *Raw judgments for Scientific Abstracts 2018* and *Raw judgments for Clinical Trials 2018*, from the 2018 edition of the Track.

4 Document Ranking

In this section, we describe how it is possible to rank the documents according to the Topic using the introduced General Representation. Once that we have transformed each document into the general representation, we need to score the similarity of each document to the Topics.

To compute the final similarity score we decide to use five sub scores (between 0 and 1). This five scores are then combined into a single scalar value. The following sections describe how to compute each score and how they are combined together.

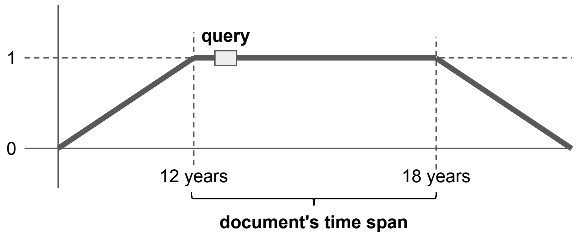

Figure 3: *Fussy match between the age range of a document and the query.*

4.1 Gene and Disease Score

The gene and disease similarity is computed as the cosine similarity between their embedded terms. The disease score is computed as the average similarity between the top K most similar pairs of diseases among those from the document and the query. Analogously, the gene score is computed using the annotated genes set instead of the diseases. The parameter K is used to control the specificity of the score.

4.2 Age Score

The age score between a given document and the query is computed as follows. The non-zero $<t_s, t_e>$ time spans are recovered from the document and from the topic general representation. To match the age demographics, we explored two alternatives: the binary and the fuzzy match. The binary match assign 1 as the age score if the query's age is between the document's minimum and maximum age, and 0 otherwise. The fuzzy match extends the binary score to a linear decay that starts from 1. Figure 3 illustrates the fuzzy match between a document's age range and the query's one. Both the binary and fuzzy match are weighted according the scalar value associated with each time span in the document and Topic general representation.

4.3 Gender Score

The gender score between a given document and the query is computed as the dot product between their *gender* feature vectors.

4.4 Precision Medicine Score

The PM score is computed as the product between their *PM* feature. Since the query *PM* feature is always set to 1 in the PM Task, we take directly only those documents that have the *PM* feature setted to 1.

4.5 Ranking

All candidate documents are ranked according to a single scalar value, i.e. a linear combination of the `gene-score`, `disease-score`, `gender-score`, `age-score`, and `PM-score` (or a subset of these according to the run). To maximize the classification accuracy of the documents into the relevant/non-relevant, we had automatically assigned the weights of the linear combination using logistic regression classifier trained on the 2018 relevance judgments.

The final score was, therefore, computed by applying also a sigmoid function to the linear combination of the features.

5 Run Configuration

Table 4 shows the configuration used in each run. A total of five runs was tested for Clinical Trials and only two for Scientific Abstracts.

The first three runs, of the Clinical Trials, use all the five features. Each run explore different combinations of the age matching approach, and different values of K parameter. The remaining two runs, made on the Clinical Trials, ignore the demographic information, and one over two ignore as well the PM information.

All the runs on Scientific Abstracts were performed using the following default configuration parameters: all features, binary matching, and parameter $K = 1$. Due to some processing limitations, only two subsets of the whole collection of documents were considered. The `default100k` and `default1m` runs worked on a random sample of $100\,000$ documents and one million documents respectively.

6 Conclusion

The presented work summarise the retrieval system proposed by the University of L'Aquila in collaboration with the University of Havana for the TREC 2019 Precision Medicine Track. The track is composed of two tasks:

- retrieve *Scientific Abstracts* that contains those treatments that could be useful for patients;
- find those *Clinical Trials* for which the patient may be eligible.

The proposed system - employed on both the tasks of the PM Track - is it made of the four steps previously described. Due to some computational issues, which could not be resolved before the deadline, we were able to submit only, five runs for the *Clinical Trials* related task, and two runs for the *Scientific Abstracts* related task. Concerning the clinical trial, our best run is **tk3onlygnds**, which uses only diseases and genes features without considering the Demographics and the *PM* information. As stated above for the *Scientific Abstracts* related task, we were not able to process the whole collection, and we applied our method only on two subsets of respectively 100,000 and 1 million documents. To summarise, it was not possible for us to clearly understand how our model performs against

Table 4: *Runs description*

RUN		FEATURES					PARAMS	
Name	**Doc.**	**Diseases**	**Genes**	**PM**	**Gender**	**Age**	**Age Matching**	**K**
`tk1allbinary`	trial	×	×	×	×	×	binary	1
`tk1allfuzzy`	trial	×	×	×	×	×	fuzzy	1
`tk3allfuzzy`	trial	×	×	×	×	×	fuzzy	3
`tk3nodemogr`	trial	×	×	×			-	3
`tk3onlygnds`	trial	×	×				-	3
`default1m`	abstract	×	×	×	×	×	binary	1
`default100k`	abstract	×	×	×	×	×	binary	1

the other participants of the TREC Precision Medicine Track. However, considering only the performances achieved in the *Scientific Abstracts* related task, we can note that a system (as our) made solely by a tagging based approach, is not competitive enough. For the reasons mentioned above, we plan to extends our approach by also incorporating classic retrieval models.

Bibliography

Grover, Aditya and Jure Leskovec (2016). “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 855–864.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural computation* 9.8, pp. 1735–1780.

Mederos-Alvarado, Jorge et al. (2019). “UH-MAJA-KD at eHealth-KD Challenge 2019: Deep Learning Models for Knowledge Discovery in Spanish eHealth Documents”. In: *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019)*. CEUR Workshop Proceedings.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). “Glove: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: `10.3115/v1/D14-1162`. URL: `https://www.aclweb.org/anthology/D14-1162`.

Piad-Morffis, Alejandro et al. (2019). “Overview of the ehealth knowledge discovery challenge at iberlef 2019”. In: *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS. org*.

Piñero, Janet et al. (2019). “The DisGeNET knowledge platform for disease genomics: 2019 update”. In: *Nucleic acids research*.

Proteccion Social, Ministerio de Salud y. *Proteccion social: Ciclo de Vida*. `https://www.minsalud.gov.co/proteccionsocial/Paginas/cicloVida.aspx`. Accessed: 2019-06-16.

Wei, Chih-Hsuan, Hung-Yu Kao, and Zhiyong Lu (2013). “PubTator: a web-based text mining tool for assisting biocuration”. In: *Nucleic acids research* 41.W1, W518–W522.