Learning to Rank Claim-Evidence Pairs to Assist Scientific-Based Argumentation

José María González Pinto[1[0000-0002-2908-3466]], Serkan Celik[2] and Wolf-Tilo Balke [1[0000-0002-5443-1215]]

[1,2] IFIS TU-Braunschweig, Mühlenpfordstrasse 23, 38106 Braunschweig, Germany
{pinto, balke}@ifis.cs.tu-bs.de, [2]s.celik@tu-bs.de

Abstract. We consider the novel problem of learning to rank claim-evidence pairs to ease the task of *scientific argumentation*. Researchers face daily scientific argumentation when writing research papers or project proposals. Once confronted with a sentence that requires a citation, they struggle to find the manuscript that can support it. In this work, we call such sentences claims – a natural language sentence – that needs a citation to be credible. *Evidence* in our work refers to a paper that provides *credibility* to its corresponding claim. We tackle the scientific domain where the task of matching claim-evidence pairs is hindered by complex *terminology variations* to express the same concept and also by the *unknown* characteristics beyond *content* that makes a paper worth to be cited. The former calls for a suitable *representation* capable of dealing with the challenge of content-based matching considering domain knowledge, whereas the latter implies a need to propose *semantic features* of suitable characteristics to guide the learning task. To this end, we test the scope and limitation of a deep learning model tailored to the task. Our experiments reveal what specific attributes can guide the learning task, the impact of using domain knowledge in the form of concepts and also the assessment of which metadata of a document, e.g., 'background', 'conclusion', 'method', 'objective', or 'results' should be considered to achieve better results.

Keywords: learning to rank; scientific claims; scientific arguments;

1 Introduction

Our work considers the problem of claim-evidence matching as a learning to rank problem whose goal is to find *evidence* in the form of a paper to make a natural language sentence, hereafter a *claim*, credible. Our notion of claim-evidence pairs draws motivation from the Argumentation Mining community. In particular, we build on the argument model established in Computational Linguistics [1]. An argument refers to a topic and consists of two components: a claim defined as a 'general concise statement that directly supports or contests a topic' while evidence is 'a text segment that directly supports a claim in the context of a given topic'. Consider for example our user Anna,

a post-doc researcher, currently writing a research proposal regarding the role of 'Lycopene' in human health. She writes the following claim: *'research has been not sufficient to establish whether lycopene consumption has a positive or a negative effect on human health'*. Anna considers submitting her claim as a query to a curated Digital Library such as PubMed to rely on high-quality content. However, among the possible papers that exist in the Digital collection, which specific paper would be needed here for Anna? Beyond the content matching capabilities of a modern retrieval system behind PubMed, what other attributes are needed? Perhaps Anna would need a paper that is highly cited, or that is published in a high-impact Journal, or a paper by a highly cited author? Is it possible to accurately learn a model that can assist users such as Anna in the difficult task of *scientific argumentation*?

In general, scientific argumentation is a complex information need and a hard-task: we will show in our experiments (Section 4) that treating a claim as query terms to measure the relevance of those query terms occurring in documents leads to an inadequate solution. Hence, we promote a learning to rank strategy tailored to the specific needs of claim-evidence pairs targeting as our main use case biomedical paper citations.

Moreover, we promote the citations on the Wikipedia archive as a *valuable* source for our novel problem because the crucial role that *citations* play in Wikipedia, i.e., valid citations provide *credibility* to Wikipedia's content. Moreover, credibility makes the task of claim-evidence ranking a challenge. How can we account for *credibility*?

To answer the question, perhaps we can gain insights from Wikipedia itself. For instance, consider Wikipedia citing sources page[1] that states: 'Wikipedia's verifiability policy requires inline citations for any material challenged or likely to be challenged and for all quotations, anywhere in article space'. Moreover, in Wikipedia's verifiability page[2] we find that 'verifiability means that other people using the encyclopedia can check that the information comes from a reliable source. Verifiability, no original research and neutral point of view are Wikipedia's core content policies'.

Given these observations, we investigate whether it is possible to learn from existing citations of biomedicine in Wikipedia, content and non-content-based patterns, to automatically rank research papers to ease scientific-based argumentation. In summary, the questions that guided our research were the following:

1. How different is claim-evidence ranking from content-based ranking?
2. How relevant is domain knowledge, given that we are targeting the biomedical field?
3. Can we gain performance improvement if the models are aware of rhetorical parts of documents? By rhetorical parts in this work, we mean metadata such as 'objective', 'methods', 'conclusions', 'background' or 'methods'.

In a nutshell, our goal is to design and implement a learning to rank model based on the novel task of learning to rank claim-evidence pairs. To this end, we characterize current citations to device semantic properties (Section 3.3 and 3.4), introduce a proper representation of the claim-evidence pairs (Section 3.2) and test different versions of our proposed model (Section 4) to assess its scope and limitations. In Section 2 we

[1] https://en.wikipedia.org/wiki/Wikipedia:Citing_sources
[2] https://en.wikipedia.org/wiki/Wikipedia:Verifiability

provide relevant related work, and finally, in Section 5, we conclude with a summary and outlook of our findings.

2 Related Work

In this section, we discuss related work on research efforts using models based on deep learning techniques in information retrieval and recent efforts to account for Argumentative machines: information systems aiming at helping users to retrieve arguments from corpora.

Information Retrieval like many other fields where machine learning is a core part of solving problems, has seen many efforts based on the Renaissance of neural networks. In particular, deep neural architectures. We refer the reader to a recent overview of various neural ranking models [17]. As first proposed in [5] neural ranking approaches can be classified as early and late combination models. Some examples of the first category are DRMM [9] that uses histogram analysis between query and document contents to model their interaction. The idea behind DRMM is to include matching signals between the query and document pairs using word embeddings as a fundamental block of transfer learning. Afterward, DRMM uses a deep learning architecture to learn latent patterns to compute the relevance score. Another example is the work of [15] that instead of representing query and document pairs as vectors, they model sequence to sequence interactions of objects. In other words, the model tries to accurately detect which parts of the pairs have an impact on the ranking.

An example of the second category of models is the work of [21] that uses Convolutional Neural Networks first to learn a latent representation of query and document pairs. Then using max-pooling to retain the most relevant features of the model, a relevance score between query and document pairs is computed using the cosine similarity.

Another instance of this category is the recent work of [23] that differs fundamentally to previous models in one relevant aspect: they proposed to learn sparse latent representations for each query and document pair. The latent sparsity representation of the proposed model showed that sparsity in neural information retrieval systems could have a positive impact. All these models demonstrated how to model meaningful interactions between query and document pairs.

One particular work that also accommodates for additional features beyond semantic matching between query and document pairs is the work of [20]. The model that the authors proposed used a Convolutional Neural Network representation of query and document pairs to model their interactions. In addition to that, they proposed to include at the end of the deep learning architecture the inclusion of features that account for relatedness between the query and document pairs. For instance, they considered word overlap and IDF-weighted word overlap between all words and only non-stop-words. These are lexical features that boost the learning to rank task as it was shown in the paper.

In contrast in our work, we proposed to include claim-evidence features that can help our claim-evidence proposed task. In particular, to include prestige and domain

knowledge features. Prestige features are particularly relevant for our problem because they allow us to model 'credibility' when assessing a match between claim-evidence pairs. The inclusion of domain knowledge allows us to answer one of our research questions to assess the impact of medical concepts in our learning to rank task.

Research efforts to perform claim-oriented document retrieval includes the work of [19], where researchers introduced a specific retrieval task focused on controversial topics. This pioneering work aims at Argumentative machines: enabling information systems to have a notion of arguments that need the support of one or more relevant claims [13]. Claims, as defined by [13], are concise statements that directly support or contest a discussed topic. Their work focused on the retrieval of controversial topics using Wikipedia. To do so, they relied on state-of-the-art retrieval models and developed a set of features in the form of a 'controversial lexicon' to re-rank their models.

3 Approach and Problem Formulation

In this section, we provide definitions to accomplish our goal: learning to rank claim-evidence pairs; then, we describe the model that we used in our attempt to answer our research questions. Finally, we describe in details the features used to help our learning models.

3.1 Problem Formulation

We follow the terminology introduced in [14] and thus refer the reader for a detailed exposition of the different ranking models that have been proposed and studied. Learning to Rank is a task to automatically construct a ranking model using training data such that the model can determine the degree of relevance or importance of a set of objects for a given query. Learning to Rank relies on machine learning algorithms to accomplish its goal.

We denote $\mathcal{Q}$ the set of queries and $\mathcal{D}$ the set of documents. We are given a set of retrieved lists where each $q_i \in \mathcal{Q}$ has its own list of candidate documents $\mathcal{D}_i = \{d_{i1}, \dots d_{in}\}$ and for each document $\in \mathcal{D}_i$ a relevancy judgement is also given. Documents that are relevant for the query q_i have judgements equal to 1 or 0 otherwise. The goal is the following: build a model that for each query q_i and its candidate list $\mathcal{D}_i$ delivers a ranking $\mathcal{R}$ such that relevant documents appear at the top of the list.

More formally, because learning to rank is a supervised learning task, we have training and testing sets to measure the success of a ranking model. Thus a training set consists of n training queries $q_i (\, i = 1..n) \in \mathcal{Q}$, their associated documents represented by feature vectors $x^{(i)} = \{x_j^{(i)}\}_{j=1}^{m^{(i)}}$ where $m^{(i)}$ is the number of documents associated with the query q_i and the corresponding relevant judgments. Then, a specific learning algorithm is used to learn the ranking model such that the output can predict the ground truth in the training set as accurately as possible. The learned model is then applied to new unseen queries to rank the relevance of the documents. Finally, the

hypothesis space that defines the function mapping the input space to the output space in learning to rank is as follows:

$$h(w, \psi(q_i, D_i)) \rightarrow \mathcal{R}$$

where $\psi(\cdot)$ plays a key role in the task because it models the query-document matching pairs in the feature space and learns a suitable weighted vector w during the training phase. In what follows we define what we mean by claim-evidence pairs and how these terms fit the general setting of query-document pairs.

Definition 1. *Claim*: a claim in this work is a natural language sentence in Wikipedia that needs or has a specific paper as a citation. Each claim represents a query q_i in our setting.

Definition 2. *Evidence*: evidence in this work is a paper that could be paired with a claim to provide credibility. Thus, each evidence represents a document in $\mathcal{D}$.

Learning to Rank Approach. In our work we focus on a pointwise approach to provide insights on two core aspects of the problem a) what type of representation is needed: content and non-content semantic features b) what specific rhetoric part of a document better suits the ranking task: 'background', 'conclusion', 'methods', 'objective' or 'results'. To that end, we implemented and tested deep learning models with our proposed features. In pointwise, the training instances are triples of the form (q_i, d_{ij}, r_{ij}) and one can train a classifier to achieve the task.

3.2 Model

In this section, we describe the model used for our learning to rank task. To learn a content match between claim-evidence pairs, we apply a learning to rank model that uses a convolutional deep learning architecture [20]. The model is an instance of a feed-forward multi-input model commonly used in Deep Learning. In particular, the model accounts for the two main inputs that correspond to our task: claim and evidence pairs. Moreover, the model also accommodates for the inclusion of a set of 'non-trainable attributes' that guide the learning to rank process. In what follows we describe the five core components of the model.

Claim-Evidence Matching. We represent each claim and evidence as sequences. Their resulting vector representations are x_c and x_e . These sequences are learned using Convolutional Neural Networks [10, 12]. The goal of having x_c and x_e is to compute a claim-evidence similarity score. One way to perform such a computation is the one introduced by [4] that defines the similarity between x_c and x_e as follows:

$$sim(x_c, x_e) = x_c^T M x_e$$

The role of the similarity matrix $M \in \mathbb{R}^{dxd}$ is to find the closest document to the input query x_c. The model learns during training the similarity matrix M as another parameter.

Non-trainable Attributes. The idea behind 'non-trainable attributes' is to guide the learning task to find an optimal hypothesis. What we want to do is to accommodate for situations where the model needs to learn attributes beyond claim-evidence matching. In our case, we introduce the idea of 'prestige' relevant to our task. Thus, we hypothesized that a paper to be cited needs to have a certain degree of 'prestige'.

Join Layer. The 'Join layer' concatenates the latent representations of x_c and x_e, the similarity score and the 'non-trainable attributes'. Up to this point, the model has not yet computed interactions between the different attributes. To accomplish the computation of the interactions, the model uses a hidden layer.

Hidden Layer. The hidden layer computes the following transformation:

$$\alpha(w_h \cdot x + b)$$

Where w_h is the weight vector of the hidden layer and α is a non-linearity function to explore more complex hypotheses. What this layer attempts is to capture possible interactions between the attributes (latent and 'non-trainable') previously computed.

Softmax Layer. Finally, the model is ready to compute a fully connected softmax layer. Formally, the model computes the probability distribution over the labels:

$$p(y = j|x) = \frac{e^{x^T \theta_j}}{\sum_{k=1}^{K} e^{x^T \theta_k}}$$

θ_k is the weighted vector of the $k - th$ class.

In our work we will show how we exploit and contrast the use of two different representations of the claim-evidence pairs: firstly, as sequences of tokens. Secondly, we also model them as a sequence of medical concepts using the UMLS. Furthermore, we incorporate in our models the notion of 'prestige' (details in Section 3.4) to measure the credibility of a paper that may influence the decision of being included as evidence given a claim.

In what follows, we provide the details of the domain knowledge that we used in our models and the prestige features that we considered.

3.3 Domain Knowledge

We rely on the Unified Medical Language System (UMLS) to incorporate into our models a representative set of biomedical entities. We used SemMedDB [11] database that uses SemRep [18] a specialized tool that given a natural language sentence returns a triple representation with the unique CUI ids and concepts from the UMLS. In particular, for this task, we used the table named PREDICATION that contains for each document, a sentence id and all the triples that SemRep identified.

Consider the following example to clarify what domain knowledge means in our case (taken from the SemRep web site)

Input: natural language sentence, for instance: "We used hemofiltration to treat a patient with digoxin overdose that was complicated by refractory hyperkalemia".

Output:

Hemofiltration-TREATS-PatientsDigoxin

overdose-PROCESS_OF-Patients

hyperkalemia-COMPLICATES-Digoxin overdose

Hemofiltration-TREATS(INFER)-Digoxin overdose

The subject and object arguments of each predication are concepts from the UMLS Metathesaurus and their relationship shown in uppercase is a relation from the UMLS

Semantic Network. Thus, for each claim and each evidence in our dataset, we extracted these concepts to measure their impact in our task.

3.4 Prestige Representation

In this section, we describe the prestige attributes that we used given our particular setting. We hypothesized that our claim-evidence rank needs a notion of relevance beyond content-matching. Thus, we will consider the following attributes:

- Impact Factor (IF): The "Impact Factor" (IF) is the primary indicator of the scientific importance of journals [8]. IF is calculated annually by the Institute for Scientific Information (ISI). Despite its widespread acceptance in the scientific world, IF has been criticized recently on many accounts as mentioned in [6]: lack of quality assessment of the citations, the influence of self-citation, or English language bias.
- Eigenfactor score (ES) is another index of scientific journal impact which uses a similar algorithm like Google's PageRank [3]. For calculating ES an iterative method is used, and journals are considered to be influential if they are cited more often by other prestigious journals [3].
- The Normalized Eigenfactor Score is the Eigenfactor score normalized, by rescaling the total number of journals in the JCR each year, so that the average journal has a score of 1. Journals can then be compared and influence measured by their score relative to 1. For example, if a journal has a Normalized Eigenfactor Score of 5, that journal is considered to be five times as influential as the average journal in the JCR.
- SCImago journal rank indicator (SJR) is another index which uses a similar method as the ES. However, this index is based on SCOPUS database which has much broader indexed journals compared to ISI [6].

3.5 Papers Rhetoric's

To account for specific metadata of a biomedical paper, we rely on SemMedDB database. Particularly relevant for our purposes is the table named SENTENCE that contains for each paper in MEDLINE: the pmid identifier for a paper, the sentence and the rhetorical category that corresponds to a given sentence. There are five of these rhetorical categories in the data: 'methods', 'conclusions', 'results', 'background', 'objective'.

Unfortunately, not all the documents contain these valuable metadata. Thus, to account for this aspect of the papers, we had to introduce a machine learning task to learn a function that can perform the mapping: from a natural language sentence to one of the rhetorical categories. We present the results of this classification task in Section 4.2.

3.6 Embeddings

In this section, we describe our primary strategy to enhance our model's performance: embeddings. In particular, we use word embeddings in two different ways.

Firstly, to improve our tokenized representations of our collection of claim-evidence. Secondly, to improve our UMLS-based claim-evidence representation.

We relied on distributional representations of words as our first step to account for deep semantics. To do so, we trained a word2vec [16] model on PubMed Central to account for a vocabulary size of 4,361,948, dimension size of 300, using a windows size of five and a minimum count of five as the parameters of the model. To train our word2vec model we used a random sample of 200 Million sentences from PubMed Central. The model trained here is the Skip-gram model using Hierarchical softmax.

To represent the medical concepts with a similar semantic representation, we used the model trained by [22]. The model that we used here is the one trained on OHSUMED: a collection of 348,566 MEDLINE medical journal abstracts used in TREC 2000 Filtering Track. To accomplish the task, researchers first transformed the free-text representation into UMLS concept identifiers. After that, they trained word2vec by Mikolov using hierarchical softmax, see [22] for more details.

4 Experimental Setup

In this section, we will empirically evaluate learning to rank models to answer our research questions. First, we will describe the datasets used in our experiments. Next, we will discuss the results of the models used to perform the automatic rhetoric classification of sentences. Then, we will discuss the performance of the ranking models.

4.1 Data Description

In this section, we will describe the dataset used in our experiments. We used the Wikipedia dump from the work of Fetahu B. et al. [7]. We limited our work to the citations that correspond to biomedical papers. To generate our data, we considered as ground truth the citations that currently exist on Wikipedia and divided the data in training, validation, and testing sets randomly.

We had to omit some of the claim-evidence pairs because we could not retrieve 'prestige' attributes of some of them. We show in Table 1 a summary of the dataset used for our experiments. To generate our claim-evidence pairs with relevant and non-relevant samples, we proceeded as follows: we indexed the abstracts of the entire PubMed up to the date of the Wikipedia dataset using Solr (hereafter SolrBM25). Then, we submitted each claim as a query to our index and retrieved up to the top ten (training and validation datasets) and 20 (testing dataset) documents in addition to the ground truth from Wikipedia. Thus, each model is trained assuming that the notion of 'relevance' from our SolrBM25 can be improved by learning a valuable latent space to re-rank the result set.

Preprocessing. As part of the preprocessing of the data, we need to consider that most of the claims and documents differ in their lengths, posing a problem to most algorithms since they are not able to work on variable length but fixed-length sequences. One way to solve this problem is to use padding. Padding means inserting a constant in the beginning or after the sequence. Thus, for instance, all the claims are

padded to the length of the longest query. The same process is applied to the documents. This approach, in general, works well if the majority of the claims, for instance, is close to the longest claim. However, if they differ, then applying padding will cause unnecessarily high memory consumption and will make the models take much time to train. In our case, this is important to investigate because we aim at optimizing the models' architecture through Bayesian optimization and a wrong decision regarding padding will make it harder for an algorithm to learn. Therefore, we investigated the balance between padding data and real data.

Unfortunately, most of the claim lengths of the corpus are far from the maximum as our analysis revealed. We observed some outliers with claims that are very long with more than 200 words. Thus, we decided to try different values until we reach a good trade-off between models' performance and computational resources. In summary, we used the following lengths to represent our queries and documents: query length is 100; document length is 1,345; document length using UMLS concept representation is 231, and vocabulary size of 50,000.

4.2 Results and Discussion

In this section, we will show the results of our experiments. Firstly, we start with the results of the 'rhetoric' classification task, and then we introduce each learning to rank model used to answer our research questions. To account for reproducibility of our results, we will be providing on request all the datasets used and the source code of all the models trained in our website.

Rhetoric Classification Task. To put the task into perspective, Table 2 shows the statistics of the metadata that we refer to as 'rhetoric' class. We can observe more than 20 Million documents that do not have this critical information. Moreover, none of the documents used in our datasets contained the metadata. Thus, we used a sample of documents from SemMedDB to learn a function that at the sentence level can determine a rhetoric category. We randomly sampled 200 thousand sentences per class. We have then divided the data using 80% as training data and 20% for testing.

For this task, we compared two machine learning approaches: naïve Bayes and Support Vector Machine both using Bag of Words to represent each sentence. In Table 3, we show the results for this task where we can see that the SVM outperforms NB by a small but significant margin of 6%.

Moreover, in Table 4, we show the results per class for the SVM. We can observe a fair balance between the F1 scores among the classes. The classes 'Results' 'Methods' and 'Conclusions' were more straightforward for the model to distinguish. Furthermore, the 'Objective' class was the most challenging performing with very high precision but at the cost of the lowest recall. Afterward, we used our machine learning model to annotate each sentence of each document present in our dataset.

Table 1. Summary of the Dataset

Data	#queries	#relevant documents	#irrelevant documents
Training	18,512	19,450	83,678
Validation	2,317	2,614	11,350
Testing	2,296	2,579	44,030

Table 2. Documents in SemMedDB with rhetoric classes

Data	#documents
Background	1,717,959
Conclusions	3,640,469
Methods	3,285,462
Objective	2,388,024
Results	3,469,876
With no rhetoric	27,851,118

Table 3. Results of the Rhetoric Classification Task

Model	Precision	Recall	F1
NB-BOW	0.76	0.76	0.76
SVM-BOW	0.82	0.82	0.82

Table 4. Results of the SVM-BOW

Class	Precision	Recall	F1
Background	0.75	0.75	0.75
Conclusions	0.86	0.86	0.86
Methods	0.81	0.88	0.85
Objective	0.87	0.73	0.79
Results	0.80	0.87	0.83

Table 5. Results models using abstracts

Model	MAP	NDCG
CNNRank-Content	0.2800	0.4415
CNNRank-UMLS	0.2374	0.4043
CNNRank-Content-Prestige	**0.3605**	**0.5068**
CNNRank-UMLS-Prestige	0.3154	0.4693
CNNRank-Cont-UMLS-Prestige	0.3306	0.4821

Table 6: Results models using rhetoric

Model	MAP	NDCG
CNN-Background-Pres	0.3227	0.4749
CNN-Conclusions-Pres	0.2807	0.4388
CNN-Methods-Pres	0.2943	0.4506
CNN-Objective-Pres	0.3295	0.4785
CNN-Results-Pres	0.3156	0.4676

Table 7. Summary of Datasets per Rhetoric

Data	# queries	#relevant documents	#irrelevant documents
Background			
Training	16,092	18,039	80,460
Validation	2,062	2,330	10,310
Testing	2,043	2,314	35,830
Conclusions			
Training	12,894	14,145	64,470
Validation	1,525	1,685	7,625
Testing	1,573	1,717	22,099
Methods			
Training	15,255	17,009	76,275
Validation	1,841	2,071	9,205
Testing	1,892	2,109	31,081
Objective			
Training	12,782	14,049	63,910
Validation	1,571	1,744	7,855
Testing	1,559	1,720	21,770
Results			
Training	15,402	17,148	77,010
Validation	1,970	2,221	9,850
Testing	1,939	2,153	32,377

We noticed that in some cases, our model did not detect some of the rhetoric classes in some documents. Thus, we decided to build new training, validation, and testing sets.

We report in Table 7 the statistics of the number of queries, relevant and irrelevant documents per rhetoric category.

Claim-Evidence Rank Task. Herein we show the results of the different variations of the models used in our ranking task.

We report two metrics to evaluate the models and answer our research questions. Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP). To calculate both metrics, we used the official $trec\_eval$ tool. To validate the differences between the models we used the Wilcoxon signed-rank test considering a two-sided hypothesis with significance level of 0.95%.

We show in Table 5 the results of each model that we implemented. In the table, 'CNNRank-Content' stands for a model that uses tokenized versions of the claim and the document pairs to learn a relevance score. 'CNNRank-UMLS' stands for a model that uses domain knowledge –UMLS concepts as the representation for each document. 'CNNRank-Content-Prestige' is a model similar to 'CNNRank-Content' but incorporates the 'prestige' attributes. 'CNNRank-UMLS-Prestige' is a model that incorporates our proposed 'prestige' attributes and uses domain knowledge to represent documents. Finally, 'CNNRank-Cont-UMLS-Prestige' is a model that uses the prestige attributes, the tokenized versions of the documents, and the domain knowledge representation of them.

To find the hyperparameters of each model's architecture, we used Bayesian optimization [2]. In what follows, we discuss the results regarding our research questions.

RQ1: How different is claim-evidence ranking from content-based ranking?

To address our first research question, we compare the performance differences between the models using only content-based matching and the models that incorporate 'prestige'. Thus, we refer here to the models 'CNNRank-Content' and 'CNNRank-UMLS' as the models using only content-based matching and compare them with its corresponding counterparts incorporating 'prestige'.

We can observe that in both cases, content-based matching models are significantly outperformed with statistical significance according to Wilcoxon Test (both cases with a p-value < 2.2e-16 using MAP and NDCG). This finding indicates that the neural models can profit from the notion of 'prestige' to achieve a better notion of 'relevance' beyond what is implied by the semantics of the matching between the claim and evidence pairs. This finding supports our hypothesis that claim-based ranking requires the inclusion of features that can represent the notion of 'prestige' to decide between papers that have similar semantics but with a different degree of 'prestige'.

RQ2: How relevant is domain knowledge, given that we are targeting the biomedical field?

The complexity of the terminology in the biomedical field motivated us to account for models where documents are sequences of medical concepts using the UMLS. Thus, to answer our second research question, we compare here the performance of the models using tokenized versions of the documents with the models using medical concept versions of the documents.

To our surprise, the tokenized versions of the documents outperformed the medical concept-based representation with statistical significance according to Wilcoxon Test (p-value 3.369e-13 for MAP and p-value 2.418e-13 for NDCG). Furthermore, we can

observe the same behavior in the models that incorporated the 'prestige' attributes (p-value of 4.721e-14 for MAP, p-value of 3.3e-14 for NDCG).

Given the results obtained, word embeddings at the tokenized label outperformed the UMLS embeddings. This is probably due to the semantics behind word embeddings that account to some degree for different representations of related and similar terms. In contrast, UMLS concept embeddings are an abstraction of the documents that for our task cannot guide the models as efficient as their word embeddings counterparts.

In summary, given the results of the experiments we have conducted, we have to conclude that for our task, domain knowledge is not needed. This finding indicates that we can disregard domain knowledge.

RQ3: Can we gain performance improvement if the models are aware of rhetorical parts of documents? By rhetorical parts in this work, we mean metadata such as 'objective', 'methods' 'conclusions', 'background', or 'methods'?

In Table 6, we show the results of our models looking at the specific rhetoric of the documents' abstracts. In particular, we used our tailored trained SVM to classify each sentence of an abstract as being of one of the five classes that we have learned from a vast collection from SemMedDB: 'background', 'conclusions', 'methods', 'objective' and 'results'.

The idea here is to discover if specific metadata of documents could benefit our models to uncover claim-evidence relevance. We can see that our best running model performed somewhat stable among all the different datasets. We can observe that no matter in which 'rhetorical' part of a document our model focuses on, the differences across the datasets are negligible. This finding indicates that the model can generate a latent representation of a biomedical paper regardless of the presence or absence of specific rhetoric parts.

We also investigated whether the models are making the same mistakes or if they could somehow be combined to build a stronger model using, for example, majority vote. Thus, we proceeded as follows: we looked at the results of each model and computed the Jaccard coefficient to measure the similarity between the models regarding the correct queries (queries where the models could find the relevant document) and incorrect queries.

After our analysis, we decided to combine the 'CNNRank-Content-Prestige' with the 'CNNRank-UMLS' because they had the lowest Jaccard coefficient regarding correct queries (0.13). Then we computed a weighted sum of the predictions of the models as follows: $\alpha * \text{pred}_1 + \beta * \text{pred}_2$. Where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the accuracy of 'CNNRank-UMLS' and 'CNNRank-Content-Prestige' respectively and $\boldsymbol{pred_1}$ and $\boldsymbol{pred_2}$ are the predictions of the corresponding models. Unfortunately, we got inferior results compare to 'CNNRank-Content-Prestige' alone. Thus, this simple combination cannot lead to better performance.

5 Summary and Outlook

In this paper, we motivated the novel problem of learning to rank claim-evidence pairs to assist scientific argumentation.

Our results showed that to successfully rank claim-evidence pairs, a model should account for other semantic properties beyond content-matching. In particular, the inclusion of features that can guide the learning process considering the 'prestige' of a paper. In our case, 'prestige' included external properties of the papers that we hypothesized could approximate better the ground truth. We have also provided empirical evidence that in our proposed solution, including domain knowledge in the form of UMLS concepts surprisingly resulted in inferior performance.

Moreover, we showed that our best-tailored model exhibits a stable performance even when it focuses on a specific rhetoric part of a document such as 'background' or 'conclusions', instead of using the whole abstract. As a future line of work, we will explore listwise approaches to improve our results.

References

1. Aharoni, E. et al.: A Benchmark Dataset for Automatic Detection of Claims and Evidence. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics. pp. 1489--1500 Dublin City University and Association for Computational Linguistics, Dublin, Ireland (2014).
2. Bergstra, J. et al.: Algorithms for Hyper-parameter Optimization. In: Proceedings of the 24th International Conference on Neural Information Processing Systems. pp. 2546–2554 Curran Associates Inc., USA (2011).
3. Bergstrom, C.T.: Eigenfactor: Measuring the value and prestige of scholarly journals. Coll. Res. Libr. News. 68, 5, 314--316 (2007).
4. Bordes, A. et al.: Open Question Answering with Weakly Supervised Embedding Models. In: Calders, T. et al. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 165–180 Springer Berlin Heidelberg, Berlin, Heidelberg (2014).
5. Dehghani, M. et al.: Neural Ranking Models with Weak Supervision. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 65–74 ACM, New York, NY, USA (2017).
6. Falagas, M.E. et al.: Comparison of SCImago journal rank indicator with journal impact factor. FASEB J. 22, 8, 2623–2628 (2008).
7. Fetahu, B. et al.: Finding News Citations for Wikipedia. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 337--346 ACM, Indianapolis, Indiana USA (2016).
8. Garfield, E.: The history and meaning of the journal impact factor. J. Am. Med. Assoc. 295, 1, 90--93 (2006).
9. Guo, J. et al.: A Deep Relevance Matching Model for Ad-hoc Retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 55--64 ACM, Indianapolis, Indiana USA (2016).
10. Kalchbrenner, N. et al.: A Convolutional Neural Network for Modelling Sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. pp. 655--665 Association for Computational Linguistics, Baltimore, Maryland (2014).
11. Kilicoglu, H. et al.: SemMedDB: A PubMed-scale repository of biomedical semantic predications. J. Bioinforma. 28, 23, 3158–3160 (2012).

12. Kim, Y.: Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751 Association for Computational Linguistics, Doha, Qatar (2014).
13. Levy, R. et al.: Context Dependent Claim Detection. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics. pp. 1489–1500 Dublin City University and Association for Computational Linguistics, Dublin, Ireland (2014).
14. Liu, T.-Y.: Learning to Rank for Information Retrieval. Found. Trends Inf. Retr. 3, 3, 225--331 (2009).
15. Lu, Z., Li, H.: A Deep Architecture for Matching Short Texts. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1. pp. 1367–1375 Curran Associates Inc., USA (2013).
16. Mikolov, T. et al.: Distributed Representations of Words and Phrases and Their Compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems. pp. 3111--3119 Curran Associates Inc., Lake Tahoe, Nevada (2013).
17. Mitra, B., Craswell, N.: An Introduction to Neural Information Retrieval. Found. Trends Inf. Retr. 13, 1, 1–126 (2018).
18. Rindflesch, T.C., Fiszman, M.: The interaction of domain knowledge and linguistic structure in natural language processing: Interpreting hypernymic propositions in biomedical text. J. Biomed. Inform. 36, December 2003, 462--477 (2003).
19. Roitman, H. et al.: On the Retrieval of Wikipedia Articles Containing Claims on Controversial Topics. In: Proceedings of the 25th International Conference Companion on World Wide Web. pp. 991--996 International World Wide Web Conferences Steering Committee, Montreal, Canada (2016).
20. Severyn, A., Moschitti, A.: Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 373–382 ACM, Santiago, Chile (2015).
21. Shen, Y. et al.: Learning Semantic Representations Using Convolutional Neural Networks for Web Search. In: Proceedings of the 23rd International Conference on World Wide Web. pp. 373–374 ACM, New York, NY, USA (2014).
22. De Vine, L. et al.: Medical Semantic Similarity with a Neural Language Model. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 1819–1822 ACM, New York, NY, USA (2014).
23. Zamani, H. et al.: From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 497–506 ACM, Torino, Italy (2018).