Predicting Entity Mentions in Scientific Literature

Yalung Zheng[1], Jon Ezeiza[2], Mehdi Farzanehpour[2], and Jacopo Urbani[1(✉)]

[1] Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
jacopo@cs.vu.nl
[2] SCITODATE B.V., Amsterdam, The Netherlands

Abstract. Predicting which entities are likely to be mentioned in scientific articles is a task with significant academic and commercial value. For instance, it can lead to monetary savings if the articles are behind paywalls, or be used to recommend articles that are not yet available. Despite extensive prior work on entity prediction in Web documents, the peculiarities of scientific literature make it a unique scenario for this task. In this paper, we present an approach that uses a neural network to predict whether the (unseen) body of an article contains entities defined in domain-specific knowledge bases (KBs). The network uses features from the abstracts and the KB, and it is trained using open-access articles and authors' prior works. Our experiments on biomedical literature show that our method is able to predict subsets of entities with high accuracy. As far as we know, our method is the first of its kind and is currently used in several commercial settings.

1 Introduction

Retrieving relevant scientific literature is crucial to advance the state-of-the-art in many disciplines. Unfortunately, a considerable subset of scientific articles is unavailable to the general public. For instance, recent estimates suggest that so far only 28% of publications are released as Open Access [22], and this excludes publications which are not yet available (e.g., preprints). In this case, third parties can only use public metadata to search for relevant literature, but this might be only a subset of all information contained in the article.

Typically, the search of scientific articles is driven by some entities of interest. For instance, one user might be interested in retrieving all papers that mention "cardiovascular disorders" or "phosphorene". These entities are often domain-specific (e.g., drugs, or experimental procedures) and are contained in high-quality knowledge bases (e.g., BioPortal [20]). Unfortunately, if the full article is missing then this process can only return articles which explicitly mention these entities in their abstracts or other metadata.

To overcome this limitation, one would need to be able to predict whether a paper might contain a given entity. This task, which we call entity prediction (EP), but is also known as entity suggestion [31], recommendation [5], or set

P. Hitzler et al. (Eds.): ESWC 2019, LNCS 11503, pp. 379–393, 2019.
https://doi.org/10.1007/978-3-030-21348-0_25

expansion [29], can be used to rank unseen articles and can lead to significant monetary savings if the articles are behind paywalls. Moreover, EP can also be used for other tasks like to augment existing knowledge bases, or might contribute for capturing the results of certain experiments in a more formal way (e.g., see the movement around nanopublications [10]).

In the literature, entity prediction has been previously applied to improve Web search results [3,12] or for knowledge base expansion [23]. In these cases, the prediction models use the entities contained in the queries as a seed to predict related entities mentioned in larger collections of documents (e.g., Web pages). Our context, however, is more challenging. First, scientific articles contain more technical nomenclature than regular Web pages and fewer entities are relevant. Second, it is harder for us to acquire large amounts of training data due the extreme coverage of topics, and because a significant number of articles is either behind paywalls or available in obsolete formats.

In this paper, we address this challenge by proposing a novel method for entity prediction on scientific literature. Our strategy is to construct a statistical model to predict which entities are likely to be mentioned in an article given its abstract and other metadata. We rely on knowledge bases to detect domain-specific entities of interest and use scientific articles released with Open Access to construct a training dataset of entity co-occurrences. After some initial failed attempts where we tried different types of models, which range from standard binary classifiers to neural networks with dense embeddings, linguistic, and other semantic features extracted from the KB, we finally obtained satisfactory results by restricting our focus to specific target entities. In this case, our model consists of a multi-layer neural network that is trained to predict whether the body of an article is likely to mention one entity of interest (or a class of entities). As input, the network receives a Bag-Of-Word (BOW) feature vector constructed using the entities in the abstract, and, optionally, also the entities mentioned in prior works of the authors. As output, it returns the probability that one or more target entities are mentioned.

We empirically evaluated our method considering scientific literature in the biomedical field. In this context, our results are encouraging: The average accuracy on predicting eight example entities from the NCIT ontology [24] in about 2K scientific articles from PubMed was 0.865 (0.804 AUC). As far as we know, we are not aware of other techniques for predicting entities in unseen articles, and our results indicate that this is a valuable asset to improve semantic search of scientific literature. In a more commercial setting, these predictions can also be used to connect suppliers of scientific equipment (e.g., special machines or chemical compounds) to potential customers (i.e., research labs) by looking at the customers' published papers. This last use case is precisely the one that motivated our research and is currently explored in a number of industrial scenarios.

2 Related Work

Semantic Search. Our work falls into the broad research topic of semantic search which largely focuses on searching related entities in knowledge bases

using structured and unstructured inputs. In this context, it is important to discover related entities, and this is a process that usually starts with a small entity subset of the target, namely the seed entities. In [9], the authors propose a Bayesian model to determine if an entity belongs to a cluster or concept, and use it to expand the set with more entities belonging to the same cluster as the seeds. In [23], it is proposed to crawl the Web to get coordinated words which are conjuncted by "and", "or" and commas, then define similarity on top of them. Moreover, the authors of [29] propose to learn wrappers of the seeds from the semi-structured documents, e.g., HTML, then use learned wrappers to find new entities in a bootstrap manner. Finally, [12] proposes GQBE, a system that takes entity tuples as examples to find similar combinations from knowledge graph. Our setting differs from these works since we assume that a large part of the related entities are not available and we focus on the retrieval of domain-specific entities which appear with lower frequencies.

In the context of query answering, the works at [14,30] propose to use language models to estimate the probability of an entity given query term and category. Furthermore, [28] proposes to use lexical similarity to constrain the entity results with categories while [3] introduces a probabilistic framework to model the query focusing on category information. More recently, [5] proposes to take the neighbor nodes of the initial entities in a knowledge graph and rank them with a learn-to-rank framework using co-occurrence, popularity and graph attributes as features. This work takes only the entity from user query and outperforms [7] which requires long descriptive text that contains concepts. Also, the authors of [31] have proposed a technique to conceptualize the input entities and build two probabilistic models between entities and concepts, thus they give not only the related entities but also the concept that explains the relationship. While these works are related in terms of objective, they are applied to domains which are significantly different. To the best of our knowledge, we are not aware of any previous works that apply EP to scientific literature using abstracts as seeds.

Co-occurrence Analysis. We use co-occurrence as a measure of relatedness. Co-occurrence is widely studied, especially in the biomedical field, in order to discover new connections between entities of interest. The most related field is *literature-based discovery* where the co-occurrence in academic publications is used as the evidence of links between concepts [25]. Moreover, many researchers have used co-occurrence for domain-specific tasks: For instance, the authors of [13] use co-occurrence as a source of information to retrieve the biological relationships between genes while [8] use co-occurrence information to form indirect links and discover the hidden connections between drugs, genes, and diseases. The work at [15] also uses co-occurrence in scientific articles to predict implicit relations between biomedical entities. While these works also make use of explicit mentions to draw conclusions, they focus on specific problems and do not consider the co-occurrence relations between abstracts (which are highly dense summaries) and the full document. Another emerging form of co-occurrence is encoded in a latent space in the form of dense numerical vectors. The seminal

work *word2vec* [17] is perhaps the most popular example of this kind applied to English words. In our work, we did use a "*word2vec*"-like approach to encode the co-occurrence of entities but we did not obtain good results.

Bag of Words (BOW). Finally, our approach uses a bag-of-words vector to represent the entities. Typically BOW models treat all the words in the same piece of text equally, but there is a significant research to enhance the performance by adding a weighting scheme [26]. In our work, we choose standard intra-document term frequency as a weighting scheme. The application of more sophisticated weighting scheme should be seen as future work.

3 Entity Prediction in Scientific Literature

Our goal is to predict whether the unseen body of the article contains some entities of interest given in input some author information and abstract. To this end, our proposal is to train a model that learns correlations between entity mentions in the abstract and in the full body and use these to make the predictions. More formally, let E and A be two predefined sets of entities and authors. Given two sequences $\langle e_1, \ldots, e_n \rangle$ and $\langle a_1, \ldots a_m \rangle$, which represent respectively the list of entities that appear in the abstract and list of authors, we want to build a model to predict with high confidence whether some entities $t_1, \ldots, t_n \in E$ appear in the body (which we assume is not accessible).

We make a few assumptions: First, we assume that we have available a significant number of full articles which we can use for training our model. This assumption is met in practice by considering articles published using the open-access model. Second, we assume that entities are available in knowledge bases which allow us to exploit semantic relations to improve the prediction. In practice, useful knowledge bases can be large domain-specific ontologies such as Unified Medical Language System (UMLS) [16], National Cancer Institute Thesaurus (NCIT) [24], Headings and Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) [6], or other encyclopedic ones like DBpedia [2]. For the purpose of this work, we view a knowledge base as a graph G where E is a set of vertices (i.e., the entities in our case) while the edges encode semantic relations between them. For instance, $\langle Odontogenesis, IsA, Organogenesis \rangle$ is an example of such relation taken from the NCIT ontology.

We distinguish two operations: *training*, that is when our objective is to construct a suitable model, and *prediction*, that is when we use the model to make the predictions. In both cases, the first operation consists of applying a state-of-the-art entity recognition (NER) tool and disambiguate the entity mentions to entities in the knowledge base. In this work, we used NobleTools [27] for the recognition and the disambiguation to the KB. For each extracted entity we extract from the knowledge base its semantic type and neighbors. Moreover, we store also the position of the entity in the original text. Then, we "embed" each entity mention into a sequence of numerical features so that it can be used by the statistical model. During the training phase, the embeddings of entities in both

metadata and body are used to train a statistical model. During the prediction, the model is used to predict new entity mentions.

In the following, we first describe two early attempts at implementing the model using two well-known techniques: A standard binary classifier and a Recurrent Neural Network (RNN) [11] used in combination with word embeddings. Neither of these methods returned adequate performance. In Sect. 4, we describe how we overcame the limitations of these two methods with a more performant approach.

3.1 Failed Attempts

As a first step in our research, we decided to investigate how a well-known technique such as a binary classifier would perform in our context. To this end, we followed the standard practice of representing entities with feature vectors and trained a classifier (we used a Support Vector Machine (SVM) [4]) to predict to what extent a given entity in the abstract correlates with the appearance of another entity in the article's body.

We proceeded as follows. First, we created a feature vector for each entity appearing in the abstract or body of the paper. Then, we concatenated the feature vectors of one entity in the abstract, one entity in the body, and some additional shared features together. The resulting vector was used as positive example while pairs or non-existing random pairs of entities were used as negative examples.

The entity feature vectors are composed of 13 features:

1. Two structural features: the distance from the start of the text and spread of an entity, namely distance between the first and the last mention of one entity. These features are introduced because typically important entities are mentioned first in the text;
2. Seven standard statistical features: TF, IDF, TF*IDF on both abstract and body entities, and respective co-occurrence frequencies;
3. Four features extracted from the considered ontologies: Jaccard, Dice, Milne-Witten [18], and Adamic-Adar [1] distances between the entities in the repository. These features aim at capturing how close the two entities are in the semantic domain which is represented by the ontology.

We calculated the Pearson coefficient of each feature against the true label and did a feature ablation study by removing the feature with the worst coefficient one by another to find the best feature subset. Unfortunately, none of these operations returned satisfactory performance. Using a training set of 3K articles and a test set of 3K Pubmed articles, our best results were 0.309 as precision and 0.394 as F1 score.

A limitation of the previous approach is that it does not take the sequence of entities into account. To include this aspect in our prediction, we considered the usage of Recurrent Neural Networks (RNN) and build a language model using the appearance sequence of entities in the abstracts and bodies as input.

To create suitable embeddings for the entities, we tried both a state-of-the-art word2vec-variant called *Entity2Vec* [19] (using sequences of entities in the paper as the "sentences") and another technique called *DeepWalk* [21] which performs random walks on the knowledge graph. During the training of the language model, we only considered the body of open access articles as training data since because the articles' bodies contain many more entities and about 85% of the entities which are mentioned in the abstract are also mentioned in the body. During the testing phase, we fed the RNN with the sequence of entities in an abstract (in the order they appear) and then computed the cosine distance between the output of the network and the embeddings of all the entities in our repository. The ones with the smallest values were selected as the output of the prediction.

Unfortunately, also this method did not return satisfactory results with the best precision, recall and F1 score averaging under 0.1. First, we observed that taking the whole body as one single sequence of entities dilutes the semantic relations between the tokens and adds noises to the model. A better approach would be to segment the text into smaller sequences depending on domain knowledge. Second, the quality of the entity embedding is not perfect and errors in this space affect the downstream application. To evaluate this problem, we took the embedding of one entity in the knowledge graph and calculated the cosine similarity against all other entities in E, rank them according to this measure, and extract the position of the synonyms. We repeated this process for 100 known synonyms pairs but the average position was below the top 10% with either method. This indicates that the quality of the entity embeddings is not high. Our third method, described in the next section, overcomes this problem by adopting a sparse representation of the entities instead.

4 Using a Neural Network with BOW

We now describe our third attempt which uses a neural network with bag-of-words (BOW) embeddings to perform the prediction. First, we map the list of entities in E into a BOW vector $\mathbf{e}$ of length $|E|$. We use different weighting scheme for the entities in the abstract and body. For the firsts, we use term frequency as the feature value. For the seconds, we use a binary value depending on the entity's appearance.

Then, we train a neural network that takes in input the vector with the frequency of entities in the abstract, which we call $\mathbf{e}_{abs}$, and in output another vector with the entities found in the body, which we call $\mathbf{e}_{bdy}$. More formally, let *abs* and *bdy* be the multisets of entities that appear in the paper's abstract and body respectively and let bdy_n the set of entities that appear only in *bdy*. Then,

$$\mathbf{e}_{abs} = \langle TF(e_1, abs), TF(e_2, abs), ..., TF(e_{|E|}, abs) \rangle \quad (1)$$

$$\mathbf{e}_{bdy} = \langle \chi(e_1, bdy_n), \chi(e_2, bdy_n), \ldots, \chi(e_{|E|}, bdy_n) \rangle \quad (2)$$

where $TF(e, t)$ denotes the number of mentions of entity e in t while $\chi(e, t)$ is a function that returns 1 if e appears in t or 0 otherwise.

We add a number of hidden layers to bridge through the high-level latent semantic correlations and add non-linearity to the model. Considering that the dimension of the input is high (i.e., $|E|$), we set the size of hidden layers much smaller than $|E|$ to densify the representation. Finally, the model is trained by minimizing the cross-entropy as usual.

After the training is finished, the network is ready to make the prediction. Let $\mathbf{e}_{\widehat{bdy}} = \langle \widehat{e}_1, \ldots, \widehat{e_{|E|}} \rangle$ be the output of the network for a given abstract. The likelihood $P(e_i)$ that entity e_i appears in the body of the article is computed as:

$$P(e_i) = \frac{\widehat{e_i}}{\|\mathbf{e}_{\widehat{bdy}}\|_2} \tag{3}$$

Since the network outputs a likelihood score for all entities, Eq. 3 can be used to make a prediction for either *all* entities in E or for a subset of them. If we restrict our focus to one or a few specific entities, then we can substantially reduce the size of the BOW vectors to only the entities which are related to our focus. To this end, let us assume that we are interested only on predicting whether the paper mentions one entity of interest e^*. In this case, we can identify all entities which are close to e^* in the knowledge base and reduce the size of the BOW vector to only those entities. We use the length of the paths between entities in the knowledge graph as distance value. More formally, let the set $N(e) = \{e\} \cup \{e_j \in V(G) | \langle e_j, e \rangle \in E(G) \vee \langle e, e_j \rangle \in E(G)\}$ be the neighbour set of the entity e in the graph G. Then we define $N^0(e) = N(e)$ and $N^{i+1}(e) = N^i(e) \cup \bigcup_{e_j \in N^i(e)} N(e_j)$ for all $i \geq 0$. In some of our experiments, we considered entities in $N^i(e^*)$ where $1 \leq i \leq 2$. This reduces the size of the embeddings from $|E|$ to $|N^i(e*)|$ and this consequently improves significantly training time. In Sect. 5, we report the performance of the model for predicting either all entities, multiple or a single class of entities, or a single one.

4.1 Including Author's Co-authorship

So far, only the entities in the abstract were considered for the prediction. However, researchers tend to specialize on specific topics, and co-authorships indicate shared interests. Therefore, the list of authors is also a valuable asset for our goal.

We propose one extension to our previous method to exploit this information. The main idea consists of using the authors as proxies to collect more relevant entities. More specifically, our approach is to first collect up to n previous publications from each of the m authors of an article, and then construct the BOW vectors for the abstract and body of a given article as follows.

$$\mathbf{e}'_{abs} = (1 - \alpha)\mathbf{e}_{abs} + \alpha \frac{1}{mn} \sum_{j=1}^{m} \sum_{i=1}^{n} \mathbf{e}_{abs_{ij}}, \tag{4}$$

$$\mathbf{e}'_{\widehat{bdy}} = (1 - \beta)\mathbf{e}_{\widehat{bdy}} + \beta \frac{1}{mn} \sum_{j=1}^{m} \sum_{i=1}^{n} \mathbf{e}_{bdy_{ij}}, \tag{5}$$

where $\mathbf{e}_{abs}$ and $\mathbf{e}_{\widehat{bdy}}$ are the BOW vectors of the abstract and the body of the given article constructed with Eqs. 1 and 2, $\mathbf{e}_{abs_{11}}, \ldots, \mathbf{e}_{abs_{mn}}$ and $\mathbf{e}_{bdy_{11}}, \ldots, \mathbf{e}_{bdy_{mn}}$ are the vectors of the abstracts and bodies of the previous n papers of m authors, and α and β are two hyperparameters used to control the weights given to the modeled histories.

Initially, we gave an equal weight to all authors. However, since they can contribute non-equivalently to the article, we decided to first determine the importance of each author by comparing the frequencies of the entities in the authors' abstracts with the content of the paper and then consider only the author with the highest overlap. In this way, we can exclude authors which have also published in many other domains and therefore might introduce noise.

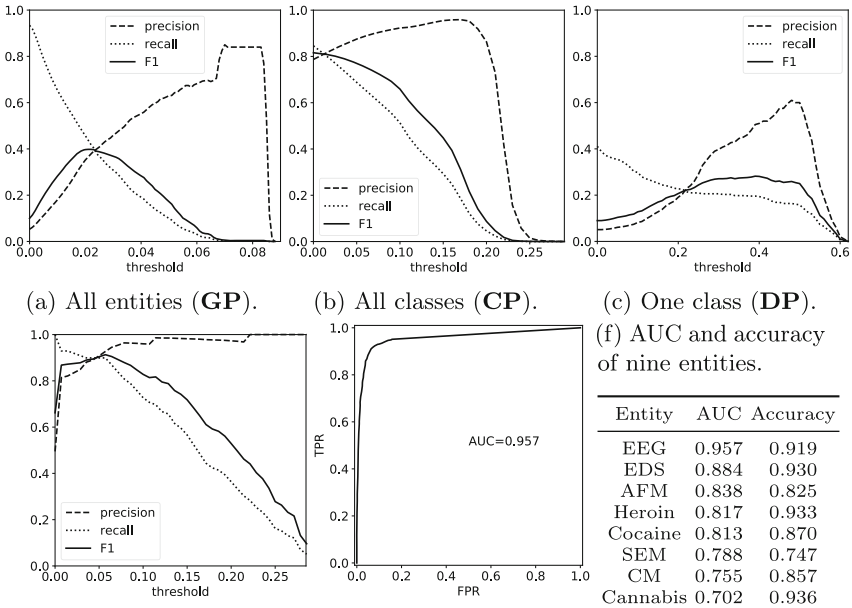
5 Evaluation

We report an empirical evaluation of the approach described in Sect. 4 on biomedical scientific literature. We chose this field since it contains high quality knowledge bases and many scientific papers. The goal of our experiments was to evaluate the accuracy in predicting either a single or all entities (Sect. 5.1), the effect of hyperparameters like the network structure or training size (Sect. 5.2), and what is the impact of adding also author information in the prediction (Sect. 5.3). All code, models, and data is available at https://github.com/NiMaZi/BioPre.

Input. As input, we considered the scientific publications which are archived in PubMed, the largest repository of biomedical articles. This collection contains about 27M articles, of which about 17M (65.5%) of them contain only the abstract, while 8M (32.3%) contain both abstract and full body. About 93.3% of the papers in the second subset contain also author information. The content of these papers is available and stored in raw text on Elasticsearch[1], which we use to query and retrieve the content of the papers.

Preprocessing. We extracted the entities in the articles with NobleTools [27], which is a popular entity annotator for biomedical text. This tool can be configured to use ontologies as the entity thesaurus. In our experiments, we used the NCIT ontology since it is a well-known ontology that covers concepts that range from disease to clinical care and is compatible with NobleTools. This ontology can be seen as a knowledge graph with 133K entities and 1.6M relations. NobleTools uses a number of heuristics to select potential entity candidates for each mention. Then, it selects one candidate among them by preferring first candidates with most synonyms, rejecting candidates that resemble abbreviations but lack a case-sensitive match, and preferring at last candidates that are unstemmed. Using NCIT, we extracted on average 59 entity mentions per abstract and 496 entity mentions in each body. Finally, we used an adapted version of *Beard*[2] for author disambiguation.

[1] https://elastic.co.
[2] https://github.com/inspirehep/beard.



(a) All entities (**GP**). (b) All classes (**CP**). (c) One class (**DP**).

(d) One entity (**EP**). (e) One entity (**EP**) (AUC).

(f) AUC and accuracy of nine entities.

Entity	AUC	Accuracy
EEG	0.957	0.919
EDS	0.884	0.930
AFM	0.838	0.825
Heroin	0.817	0.933
Cocaine	0.813	0.870
SEM	0.788	0.747
CM	0.755	0.857
Cannabis	0.702	0.936
duck	0.682	0.770
Avg.	0.804	0.865

Fig. 1. Precision, recall, and F1 for four types of predictions. (e) reports also the area under the curve (AUC) for the **EP** prediction.

Testbed. We used Keras[3] to implement the various models and Tensorflow[4] as backend. All the models are constructed with fully connected layers, and have batch-wise L1 normalization and 0.5 dropout rate associated with each layer. Unless otherwise specified, the models were trained with binary cross-entropy as loss function and the weight matrix was updated with Nesterov-accelerated Adaptive Moment Estimation (Nadam). We used mini-batch strategy for updating the model, where each batch contains 1024 articles. All the models were trained using a machine with a dual 8-core 2.4 GHz (Intel Haswell E5-2630-v3) CPU, 64 GB RAM, and two NVIDIA TITAN X graphic cards with Pascal architecture and one NVIDIA GTX 980 graphic card. Training a batch of articles took about 6.5 min and we did not observe improvements after 5–10 epochs of training.

5.1 Entity Prediction Using Abstract Entities

We trained a number of models to perform four types of predictions: First, we perform a general prediction (**GP**), which means that try to predict all entity

[3] https://github.com/keras-team/keras.
[4] https://www.tensorflow.org/.

Table 1. Accuracy (**EP** prediction) changing several parameters.

(a) Different activation functions.

Activation Function	Accuracy
Sigmoid	0.894
Hyperbolic Tangent (tanh)	0.800
Rectified Linear Unit (ReLU)	**0.903**

(b) Different number of entities.

Entities	Accuracy	Input Size
All entities	**0.903**	133,609 (100%)
N^2	0.878	**33,934 (25.4%)**
Leaf Nodes	0.663	110,184 (82.5%)

(c) Different training sets.

Training Set	Accuracy
2,048 (2 batches)	0.693
10,240 (10 batches)	0.840
19,456 (19 batches)	0.873
28,672 (28 batches)	0.890
56,320 (55 batches)	0.903

(d) Different weighting scheme.

Weighting Scheme	Accuracy
binary	0.877
tf	**0.903**
$log(tf)$	0.887
$tf \cdot idf$	0.881

mentions in the body. Second, we predict all possible *classes* of entities in the body (**CP**). Third, we predict whether the article mentions one class of entities. Fourth, we predict whether the article contains the mention of one specific entity. For the third and fourth cases, we chose the class "Disease and Syndromes" (**DP**) which contains 5227 entities while for the fourth case we chose the entity "electroencephalography (EEG)" (**EP**). This arbitrary choice was selected due to a real-world business case.

We created a neural network with one hidden layer of 512 units and the Rectified Linear Unit (ReLU) as a global activation function. Then, we selected a random subset of 147K articles as training data and 3K articles as test data for the first three types of predictions. For the fourth type of prediction, we selected 56K and 2K random articles as training and test data respectively. In this case, the test dataset contains about 1K positive examples and 1K negative ones.

We performed various experiments changing the output threshold value and calculated the precision and recall (for **EP** we also computed the area under the curve of ROC). The results are shown in Fig. 1. As we can observe from the graphs, the F1 score for the **GP** predictions is significantly lower than for the prediction of a single entity (**EP**). The F1 for **CP** is high as well, but this is misleading because here we are predicting all classes and the articles almost always contain the same classes of entities. For them, the model learns to always return true (and indeed the best results are obtained by setting the threshold closed to zero). In contrast, the F1 for predicting EEGs (**EP**) is high, but in this case the threshold is not zero which means that the network has learned to discriminate. From these results, we conclude that our model has indeed learned to predict the occurrence of one entity of interest with high accuracy.

It is important to mention that Figs. 1d and e report the results for one specific entity, namely EEGs. In order to verify whether similar results can also be obtained with other entities, we selected eight different entities and repeat the same experiment. Instead of picking random entities, we made an effort to

select a representative sample that contains entities which both are specific and generic, and which belong to different classes. More specifically, we picked "Energy Dispersive Spectroscopy (EDS)" from the category of spectroscopy, "Atomic Force Microscope (AFM)", "Scanning Electron Microscope (SEM)" and "Confocal Microscope (CM)" from the category of microscope, "Heroin", "Cocaine" and "Cannabis" from the category of drugs, and "duck" from the category of birds. Figure 1f reports the Area Under the ROC Curve (AUC) and accuracy for each entity (note that the table reports also the same AUC of EEG shown in Fig. 1(e). As we can see from the table, the models are able to return fairly high scores also for other entities, which means that it can handle other types of entities as well.

5.2 Entity Prediction with Different Hyperparameters

We have also performed a series of experiments changing some hyperparameters or configurations of the network to see to what extent the performance of the single entity prediction (EEG) is affected by these changes. More in particular, we tested different activation functions (Table 1a), different subsets of entities, i.e., all entities, only the neighbours in $N^2(x)$ where x is the target entity, and only the "leaf" entities in KB (Table 1b), different training set sizes (Table 1c), and different weighting schemes (Table 1d). All these models, except the study on different training set size, are trained on 56K articles and share the same network settings as the previous experiments.

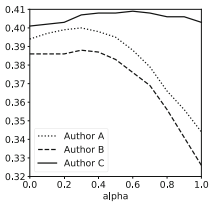
We can draw some conclusions from these results. First, we observe from Table 1a that ReLU delivers the best results. Second, the study reported in Table 1b shows that while the best results are obtained by considering all entities, if we consider only the neighbors of the entity (N^2), then we still get a fairly high accuracy, but with the additional advantage that we reduced the size of the input vectors to 25% of the original size. This loss in terms of accuracy might be acceptable if the domain contains a very large number of entities. Table 1c shows that while the accuracy gradually saturates with more than ten batches of training articles, we still need to use the entire training set to get the best results. Finally, we learn from Table 1d that the term frequency (tf) is the best weighting scheme for the BOW vector.

5.3 Entity Prediction Including Co-Authorship

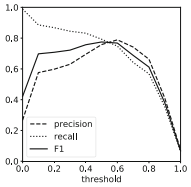
We now provide some preliminary results on including authors' information in the prediction as described in Sect. 4.1. First, we selected three representative authors whose history vectors have different variances (anonymized details are reported in Fig. 2a). For each author, we randomly picked 200 articles to create the history of abstracts. Then, we used these vectors to predict all entities in the body of other 100 random articles. We measure the F1 score by changing the complement parameter α from 0.0 to 1.0 (if α is zero then the approach is not considering any prior work while if it is 1 it only considers prior works). The results, shown in Fig. 2b, show that including the information of authors with lower variance does improve the F1 but this is true as long as the author

(a) Statistics of three sampled authors.

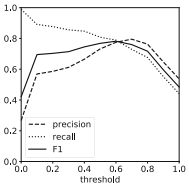
Author	Articles	Variance
Author A	6,742	0.63
Author B	1,384	0.48
Author C	1,312	0.32



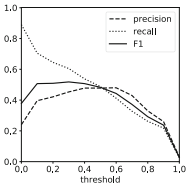
(b) F1 with author embeddings.



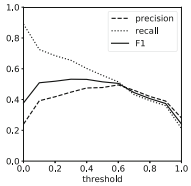
(c) DD without authors.



(d) DD with authors.



(e) Mi without authors.



(f) Mi with authors.

Fig. 2. a–b: Statistics and performance including author embeddings. c–f: Performance on entities "Drug Dependencies (DD)" and "Microscope (Mi)" with and without author embeddings.

publishes in the same domain (i.e., with low variance) because otherwise noise is introduced (as shown for authors A and B in Fig. 2b). This motivates our choice of selecting only the author with the highest overlap with the content of the paper.

We then evaluate the change of performance if we restrict our focus to the prediction of small groups of entities that are in the same category. We picked the class "microscopes" (that contains entities like "Scanning Electron Microscope", "Transmission Electron Microscope", "Scanning Tunneling Microscope", "Confocal Microscope", etc.) and "drug dependences" (with entities like "Alcohol Dependence", "Cannabis Dependence", "Cocaine Dependence", etc.) as our target groups (we selected these two classes of entities since they are the ones whether the authors have published). The results, shown in Figs. 2c–f, show a moderate increase of the F1 when we include prior abstracts of the selected author. In the first case, the increase of the F1 was about 0.8% while in the second case it was about 2.5%[5]. These results confirm that indeed the authors constitute a valuable asset to improve the performance of the prediction.

5.4 Limitations

Table 2 reports, as anecdotal evidence, the top 10 relevant entities identified for EEG. The relevance scores in this table were computed by simply creating a fake abstract where only EEG was mentioned and ranking the entities with the highest output values. While these results do not reflect completely the output

[5] These experiments are repeated multiple time ($\geq$5).

Table 2. Top 10 relevant entities of EEG.

Entity	Relevance score	Entity	Relevance score
Pharmacodynamic Study	13.98	NIPBL wt Allele	12.68
Obstructive Sleep Apnea	13.91	Audit	12.39
Proband	13.23	Central Nervous System Involvement	12.38
Lactic Acidosis	13.12	Cornelia De Lange Syndrome	12.35
Sweden	13.00	Reye Syndrome	12.19

of the network (since typically more entities are added in the input), they give us an indication on which are the most relevant entities according to our model. As we can see from the table, the model recognizes some relevant entities (like diseases which require the use of EEG), but also returns some generic entities (like Sweden).

We investigated the causes of errors using the optimal configuration to gain some insights into the limitations of our method. If we consider once again the prediction of EEG, then we obtained 244 errors in our test set, of which 118 were false positives and 126 false negatives. After analyzing the false positives, we divided the errors into three major problems:

1. The prediction might become too much biased towards entities with high intra-document frequency.
2. A similar bias is given for common entities which appear frequently in a wide variety of topics.
3. The method is not able to distinguish secondary content in the abstract that will not be discussed in detail in the corresponding body.

The first two problems make up 96% of all the false positives while the third problem makes up for 4% of the cases. We believe that the first two problems can be addressed by introducing more sophisticated weighting schemes to balance different intra-frequencies and by giving smaller weights to abstract concepts. Addressing the third cause of error requires a deeper understanding of the organization of a scientific article. Intuitively, the important content is mentioned in the front of the abstract, thus the position of each entity in the abstract could be used as a naive measurement of importance. A deeper investigation on these issues should be seen as future work.

6 Conclusion

We proposed a machine-learning-based technique to predict entities mentioned in scientific articles using the articles' metadata. This task is useful to improve the retrieval of relevant publications when the full content is not available either because of a paywall or due to other reasons (e.g., preprints). Our technique can be used to search for classes of entities or be targeted to specific entities (e.g., some special equipment, as it was for one of our business cases). Moreover, it can be also useful for performing knowledge base completion, or more generally to discover related entities based on co-occurrences.

Our best-performing approach uses a neural network and a sparse representation such as BOW vectors using the entities in the abstract. We also show that the performance can be further improved if we also consider prior works by the authors. The results on the biomedical field are encouraging, especially if we restrict the focus to subsets of entities. To the best of our knowledge, ours is the first technique of its kind and several directions for future research come to mind. First, we plan to address the limitations outlined in Sect. 5.4. Second, it is interesting to research more sophisticated mechanisms to include author information to further improve the performance. It is also interesting to improve the predictions in order to distinguish principal and secondary entities, or to determine also the position of the entity in the paper (e.g., either in the evaluation or in the related work section). Finally, we plan to extend the application of this work also to other fields such as physics or chemistry.

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Soc. Netw. **25**(3), 211–230 (2003)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC - 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Balog, K., Bron, M., De Rijke, M.: Query modeling for entity search based on terms, categories, and examples. ACM Trans. Inf. Syst. (TOIS) **29**(4), 22 (2011)
4. Bishop, C.: Pattern Recognition and Machine Learning. Springer, New York (2006)
5. Blanco, R., Cambazoglu, B.B., Mika, P., Torzec, N.: Entity recommendations in web search. In: Alani, H., et al. (eds.) ISWC 2013. LNCS, vol. 8219, pp. 33–48. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41338-4_3
6. Côté, R.A., College of American Pathologists, et al.: Systematized nomenclature of medicine. College of American Pathologists (1977)
7. Damljanovic, D., Stankovic, M., Laublet, P.: Linked data-based concept recommendation: comparison of different methods in open innovation scenario. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 24–38. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30284-8_9
8. Frijters, R., Van Vugt, M., Smeets, R., Van Schaik, R., De Vlieg, J., Alkema, W.: Literature mining for the discovery of hidden connections between drugs, genes and diseases. PLoS Comput. Biol. **6**(9), e1000943 (2010)
9. Ghahramani, Z., Heller, K.A.: Bayesian sets. In: Proceedings of NIPS, pp. 435–442 (2005)
10. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nanopublication. Inf. Serv. Use **30**(1–2), 51–56 (2010)
11. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Nat. Acad. Sci. **79**(8), 2554–2558 (1982)
12. Jayaram, N., Gupta, M., Khan, A., Li, C., Yan, X., Elmasri, R.: GQBE: querying knowledge graphs by example entity tuples. In: Proceedings of ICDE, pp. 1250–1253 (2014)

13. Jelier, R., Jenster, G., Dorssers, L.C., van der Eijk, C.C., van Mulligen, E.M., Mons, B., Kors, J.A.: Co-occurrence based meta-analysis of scientific texts: retrieving biological relationships between genes. Bioinformatics **21**(9), 2049–2058 (2005)
14. Jiang, J., Lu, W., Rong, X., Gao, Y.: Adapting language modeling methods for expert search to rank Wikipedia entities. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2008. LNCS, vol. 5631, pp. 264–272. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03761-0_27
15. Kastrin, A., Rindflesch, T.C., Hristovski, D.: Link prediction on a network of co-occurring MeSH terms: towards literature-based discovery. Methods Inf. Med. **55**(04), 340–346 (2016)
16. Lindberg, D.A., Humphreys, B.L., McCray, A.T.: The unified medical language system. Methods Inf. Med. **32**(04), 281–291 (1993)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of NIPS, pp. 3111–3119 (2013)
18. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of CIKM, pp. 509–518 (2008)
19. Ni, Y., Xu, Q.K., Cao, F., Mass, Y., Sheinwald, D., Zhu, H.J., Cao, S.S.: Semantic documents relatedness using concept graph representation. In: Proceedings of WSDM, pp. 635–644 (2016)
20. Noy, N.E., et al.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Res. **37**, W170–W173 (2009)
21. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: Proceedings of KDD, pp. 701–710 (2014)
22. Piwowar, H., et al.: The state of OA: a large-scale analysis of the prevalence and impact of open access articles. PeerJ **6**, e4375 (2018)
23. Sarmento, L., Jijkuon, V., de Rijke, M., Oliveira, E.: More like these: growing entity classes from seeds. In: Proceedings of CIKM, pp. 959–962 (2007)
24. Sioutos, N., de Coronado, S., Haber, M.W., Hartel, F.W., Shaiu, W.L., Wright, L.W.: NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information. J. Biomed. Inform. **40**(1), 30–43 (2007)
25. Swanson, D.R.: Fish oil, Raynaud's syndrome, and undiscovered public knowledge. Perspect. Biol. Med. **30**(1), 7–18 (1986)
26. Tirilly, P., Claveau, V., Gros, P.: A review of weighting schemes for bag of visual words image retrieval. Technical report (2009)
27. Tseytlin, E., Mitchell, K., Legowski, E., Corrigan, J., Chavan, G., Jacobson, R.S.: NOBLE-Flexible concept recognition for large-scale biomedical natural language processing. BMC Bioinformatics **17**(1), 32 (2016)
28. Vercoustre, A.-M., Pehcevski, J., Thom, J.A.: Using Wikipedia categories and links in entity ranking. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds.) INEX 2007. LNCS, vol. 4862, pp. 321–335. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85902-4_28
29. Wang, R.C., Cohen, W.W.: Iterative set expansion of named entities using the web. In: Proceedings of ICDM, pp. 1091–1096 (2008)
30. Weerkamp, W., Balog, K., Meij, E.: A generative language modeling approach for ranking entities. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2008. LNCS, vol. 5631, pp. 292–299. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03761-0_30
31. Zhang, Y., Xiao, Y., Hwang, S.w., Wang, H., Wang, X.S., Wang, W.: Entity suggestion with conceptual explanation. In: Proceedings of IJCAI, pp. 4244–4250 (2017)