Coner: A Collaborative Approach for Long-Tail Named Entity Recognition in Scienti c Publications

Vliegenhart, Daniel; Mesbah, Sepideh; Lofi, Christoph; Aizawa, Akiko; Bozzon, Alessandro

Publication date
2019

Document Version
Accepted author manuscript

Published in
International Conferences on Theory and Practice of Digital Libraries (TPDL)

Citation (APA)
Vliegenhart, D., Mesbah, S., Lofi, C., Aizawa, A., & Bozzon, A. (Accepted/In press). Coner: A Collaborative Approach for Long-Tail Named Entity Recognition in Scienti c Publications. In International Conferences on Theory and Practice of Digital Libraries (TPDL) Oslo, Norway: Springer.

Coner: A Collaborative Approach for Long-Tail Named Entity Recognition in Scientific Publications

Daniel Vliegenthart, Sepideh Mesbah, Christoph Lofi, Akiko Aizawa, Alessandro Bozzon

Delft University of Technology, National Institute of Informatics Tokyo
Van Mourik Broekmanweg 6, 2628 XE Delft Netherlands, 2 Chome-1-2 Hitotsubashi, Chiyoda, Tokyo 100-0003 Japan
{d.vliegenthart, s.mesbah, c.lofi,a.bozzon}@tudelft.nl
aizawa@nii.ac.jp

Abstract. Named Entity Recognition (NER) for rare long-tail entities as e.g., often found in domain-specific scientific publications is a challenging task, as typically the extensive training data and test data for fine-tuning NER algorithms is lacking. Recent approaches presented promising solutions relying on training NER algorithms in an iterative weakly-supervised fashion, thus limiting human interaction to only providing a small set of seed terms. Such approaches heavily rely on heuristics in order to cope with the limited training data size. As these heuristics are prone to failure, the overall achievable performance is limited. In this paper, we therefore introduce a collaborative approach which incrementally incorporates human feedback on the relevance of extracted entities into the training cycle of such iterative NER algorithms. This approach, called Coner, allows to still train new domain specific rare long-tail NER extractors with low costs, but with ever increasing performance while the algorithm is actively used in an application.

1 Introduction

With the increasing amount of scientific publications, there is a growing need for methods that facilitate the exploration and analysis of a given research field in a digital library collection [9], but also for techniques which can provide effective retrieval and search experiences. To this end, *"deep meta-data"* extracted from scientific publications allows for novel exploration capabilities [10].

Domain-specific typed named entities [11] such as *datasets* used in a given publication; the *methods* applied to the data or used in implementation are representative examples of deep meta-data. However, extracting and typing named entities for this scenario is hard, as most entities relevant to a specific scientific domain are very rare, i.e. they are part of the *entity long-tail*. Most current state-of the art Named Entity Recognition (NER) algorithms focus on high-recall named entities (e.g., person and location) [7], as they rely on extensive manually curated training and test data. Due to the rare nature of long-tail entity types,

training data is scarce or non-available. Some approaches addressed this problem by relying on bootstrapping [18] or entity expansion [3,6] techniques, achieving promising performance. However, how to train high-performance *long-tail* entity extraction and typing with minimal human supervision remains an open research question.

Recently, TSE-NER [12] was presented, an iterative approach for entity extraction in scientific publications. The approach starts with a small seed set of known entity instances; for each type it is sufficient to have one or two domain experts denote between 5 to 50 known entities. These sets are then heuristically expanded and annotated to generate training data to train a new traditional NER classifier, and heuristically filtered to remove likely false positives to create the entity set for the next iteration. As results of experiments in [12] have shown, this approach is hampered by the simplicity and unreliability of the heuristics used for expanding, but especially by those used for filtering the current iteration's entity set. Nonetheless, the approach promises a lot of potential if these heuristics can be improved by bringing intelligently human judgment in the loop.

Original Contribution. In this paper we extend TSE-NER with incremental, collaborative feedback from human contributors to support the heuristic filters. The core goal of this paper is to further the understanding of *how far does human feedback confirm or conflict with TSE-NER heuristics* and *how does incorporating human feedback into the TSE-NER filtering step improve the overall performance with respect to precision, recall, and F-measures*. For this we introduce `Coner`, an approach that allows the users of our system to continuously provide easy-to-elicit low-effort feedback on the semantic fit and relevance of extracted entities. Also, new entities may be added that they deem relevant for a specific facet/type. This feedback is then exploited to support the heuristic expansion and filter phases of the TSE-NER algorithm. The human-in-the-loop approach allows us to still maintain the advantages of the initial design of TSE-NER (i.e., training a NER algorithm cheaply, only relying on a small seed set, and providing an immediate result to users with acceptable extraction quality), while exploiting the human feedback into the next NER training iteration. Coner allows the TSE-NER system to improve its performance over time by benefiting from additional human intelligence in the training process. Coner is available as an open-source project.[1]

We performed two experiments to evaluate our approach on a collection of 11,589 data science publications from ten conference series: 1) an exploratory experiment performed on 10 papers and with 10 users showing that by utilizing human feedback, up to **94.3%** of false positives can be detected for the *dataset* entity type and **57.9%** for the *method* entity type; 2) similar to experiment (1) but receiving only human feedback on entities with high expected information gain in order to maximize the impact of user feedback. This resulted in an average per-entity annotation time of just above 15 seconds and an increase of precision of up to 4% by boosting the filtering step of TSE-NER.

[1] `https://github.com/vliegenthart/coner_interactive_viewer`

2 TSE-NER: An Iterative Approach for Long-Tail Entity Extraction

In this section we will summarize TSE-NER, an iterative five-step low-cost approach for training NER/NET classifiers for long-tail entity types. For more detailed information on this approach, refer to [12]. The approach is summarized in the following five steps:

1. For *Training Data Extraction*, a set of *seed terms* is determined, which are known named entities of the desired type. The *seed terms* are then used to identify a set of sentences containing the term.
2. *Expansion strategies* are used to automatically expand the set of seed terms of a given type, and the training data sentences.
3. The *Training Data Annotation* step is used to automatically annotate the expanded *training data* using the expanded seed terms.
4. A new *Named Entity Recognizer* (NER) will be trained using the annotated training data for the desired type of entity.
5. The *Filtering step* refines the list of extracted named entities by heuristically removing those entities which are most likely false positives. The set of remaining entities is treated as a seed set for the next iteration. This step is the focal point of this paper.

2.1 Heuristic Filtering

In this final step of TSE-NER, which is also the focus of this work, the trained NER model is used to annotate the whole corpus and consider all the positively annotated terms as candidate terms for the next round of iteration. As the training data for training the NER is noisy, the list of entities extracted by the NER contains many items which are not specifically related to the entity type of interest. Therefore, the goal of this last step is to filter out all terms which are most likely not relevant using the following basic heuristics:

Wordnet + Stopwords (WS). To preserve only the domain-specific terms and exclude the general English terms we filter out the stopwords (e.g. something) and concepts coming from "common" English language (e.g., "dataset", "software") that could be found in Wordnet[2].

Similar Terms (ST). The idea is to keep only the entities which are semantically similar to the seed terms of a given entity type. While there can be many implementations for capturing semantic relatedness, word embeddings [13] have shown to perform this task particularly well. In this step we cluster entities based on their embedding feature using K-means clustering, and keep only the entities that appear in the cluster that contains a seed term.

Pointwise Mutual Information (PMI). This filtering heuristic is inspired by Hearst Patterns [14]. We measure the number of times two given keywords appear

[2] `http://wordnet.princeton.edu/`

together in a *sentence* in our corpus. As an example the word `SVM` appears mostly with the word `Method` in sentences, which is an indicator of being a `method` entity type. In this step, we filter out the entities having a PMI measure lower than a threshhold.

Knowledge Base Lookup (KBL). Excluding entities that have a reference in the DBpedia knowledge base under the assumption that, if they are mentioned in DBpedia, then they are not long-tail domain-specific entities.

Ensemble Majority Vote (EMV). Preserving the entities that are passed through two out of three selected filtering strategies.

Interested readers can refer to [12] for detailed explanation. As those heuristics are rather basic in their nature, we discuss in the next section if filtering can be supported by human feedback.

3 Collaborative Crowd Feedback

As outlined in the previous section, a core design feature of TSE-NER is the heuristic filter step in each iteration, which is designed to filter out named entities which are most likely misrecognized (this can easily happen as the used training data is noisy due to the strong reliance on heuristics). While it was shown in [12] that this filter step indeed increases the precision of the overall approach, it does also impact the recall negatively. For example, this could happen by filtering out *true positives*, i.e. entities which have been correctly identified by the newly trained NER extractor but are filtered out by the heuristic. This could for example happen if a domain-specific named entity is part of common English language. More importantly, the heuristic filter often does not reach its full potential by not filtering all *false positives*, i.e. entities which are incorrectly classified as being of the type of interest, and should have been filtered out by the heuristics but were missed. Also, for the expansion phase, the heuristics often miss relevant entities which should be added.

These shortcomings are addressed in this paper by introducing an additional layer on top of the basic TSE-NER training cycle described in Section 2. Instead of treating the algorithm only in isolation, we also consider the surrounding production system and its users (in most cases, this would be a digital library repository with search, browsing, and reading/downloading capabilities). When the production system is set-up, a NER algorithm is trained for each entity type of interest (e.g., datasets, methods, and algorithms for data science) using the TSE-NER workflow until training converges towards stable extraction performance. Then, the resulting trained NER algorithm is applied to all documents in the repository, annotating their full-texts. Users then can interact with the recognized entities, providing feedback.

For this, we introduce novel Coner modules:

1. **Coner Document Analyser (CDA)**: This module serves two purposes; analyse documents to extract "deep metadata" and intelligently select entities for annotation. In an user experiment like the one presented in this paper,

the CDA selects the documents and entities where user feedback would be most effective (see Section 4). In a real-life deployed version of Coner, users would continuously provide feedback on documents they are currently reading as part of their normal consumption workflow, so no document selection is necessary.
2. **Coner Interactive Document Viewer (CIDV)**: Online interactive viewer that renders PDF documents and visualises automatically annotated entities. The CIDV allows users to interact with entities by giving feedback on existing annotations, or adding new typed named entities.
3. **Coner Feedback Analyser (CFA)**: Analyzes the entity type labels for each entity that received human feedback, and also decides which labels should be considered valid and which ones are irrelevant. This feedback is then incorporated into the iterative TSE-NER training.

3.1 Coner Document Analyser CDA

This module selects representative papers from the document corpus based on being published at a higher-level conference, having average length, high citation counts, and an average number of distinct recognized typed entities in their full texts. In addition we employ a heuristic smart named entity selection (HSES) mechanism that solely selects entities with high potential knowledge gain about the entitys relevance. A traditional approach to implement this is to use merely active learning techniques to select the most useful examples (e.g. based on informativeness) [15] for labeling and add the labeled example to training set to retrain the NER model. However the reliability of active learning (AL) techniques suffers when dealing with noisy training data generated in a semi-automatic fashion as AL techniques are heavily influenced by the quality of the initial labeled examples. For this reason we designed the heuristic smart entity selection mechanism specific for long-tail entities in our document collection (i.e. where there is an overlap of semantic spaces between the different facets). HSES exclusively selects heuristic filtered entities that were doubly classified; recognised as a relevant entity and kept by the TSE-NER filter for multiple facet NERs. Doubly classified entities clearly indicate an overlap of semantic spaces between NERs for different facets, because in reality, it is extremely unlikely that a single named entity describes a *dataset* and a *method* name.

3.2 Coner Interactive Document Viewer CIDV

We introduce an *interactive document viewer*, rendering PDF documents and highlighting recognized named entities. The viewer is based on the NII PDFNLT [1,2], which already included a basic viewer and a sentence annotation tool. One of our design goals for the interactive viewer component was to impose as little cognitive load on the users as possible, thus only very simple feedback mechanisms have been considered. During our proof of concept testing phase, we recruited 10 lab student of graduate or post-graduate level to stress test and give feedback on the viewer. Based on the feedback of these users, we opted for

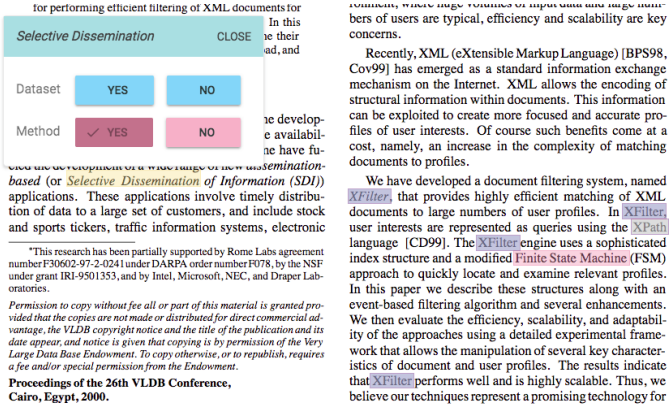

Fig. 1: Coner Interactive Document Viewer with highlighted entities

a system design allowing for simple YES/NO relevance feedback for recognized entities. Furthermore, users can add new typed entities by selecting n-grams in the document and assigning an entity type (Figure 1). For other users, these manually added entities are also highlighted, and additional user feedback can be provided for them.

3.3 Coner Feedback Analyser CFA

The purpose of the feedback analyzer is to aggregate collected user feedback on entities, and decide which new entities to finally add and which entities to label as incorrectly typed. In the current version of the feedback analyzer, this is realized with a simple majority vote on the user feedback.

However, like with any crowd-sourcing task, the feedback analyzer can be further extended to cope with common crowd-sourcing problems like spam, malicious indent, or incompetent users. For example, while for our prototype system maliciousness was not an issue, we could already see that some users were significantly more reliable than others. This also reflects in their time investment: more reliable users took much longer to provide feedback on a document, while some users provided feedback in a time frame which should not be sufficient for even reading the paragraphs surrounding an entity. Here, more complex user and task models should help to increase the reliability of aggregated user annotations. As a minimalist step towards this, we only consider users who provided feedback on at least 10 entities per publication, and only considered majority votes with at least 3 votes.

As described in Section 2.1, TSE-NER filters the current set of terms every iteration. Coner boosts this process by adding or removing entities from the iterations. Filtering heuristics can be used individually or in an ensemble. Ensemble filtering was shown to have the best, but still limited performance [12]. Coner overwrites the filtering heuristics by ensuring that entities which were labeled by users as irrelevant for a type are always removed during filtering, and entities labeled as relevant are always retained.

4 Evaluation

To evaluate the effectiveness of incorporating crowd feedback into the NER training process, we focus on the following two research questions:

RQ1 What are the properties of obtained user feedback? Especially, in how far does human feedback confirm or conflict with TSE-NER heuristics?

RQ2 How does incorporating human feedback into the TSE-NER filtering step improve the overall performance with respect to precision, recall, and F-measures?

To answer these research questions we conducted two user experiments. We focus on the two entity types *dataset* and *method* in data science publications. We had corpus of 11,589 papers from 10 conferences on data science available (this is the same corpus as used by [12]). We conducted the user interaction with the Coner system in a lab setting, recruiting graduate-level / post-graduate-level volunteers who are knowledgeable in the data science domain.

The first experiment, as described in Section 4.1, focuses on answering RQ1 by asking users to give feedback or add to unfiltered extracted entities (i.e., on the output of TSE-NER using expansion but no filtering heuristics). By comparing crowd-based filtering to the different filter heuristics, we can obtain insights into their relative performance.

As we only had a limited number of volunteers available for this evaluation, we selected papers from our corpus using the Coner Document Analyser CDA (without the heuristic smart entity selection mechanism HSES) for which the expected impact of additional annotations is representative for the whole collection.

Furthermore, for the second experiment in section 4.2, instead of relying on our users to decide themselves on which entity to provide feedback, we actively steer this process towards entities for which human feedback would have a significant expected impact and use the HSES mechanism. In particular, we focus on entities which were classified as both *dataset* and *method*. This happened quite often in our collection (i.e. 22% of all the detected entities in the whole corpus), and in nearly all cases, at least one classification is incorrect. We divert the decision which of the two types (if any) is correct for the entity to our system's users.

4.1 Experiment 1: Human Feedback on Unfiltered Entities

In this section, we look into the properties of user feedback itself, and also evaluate how it conflicts or supports TSE-NER heuristics.

Documents and Evaluators: Ten papers were selected from multiple conferences of interest using the Document Analyzer. We selected from the following conferences: The Web Conference (3 papers), ACL (3 papers), ICWSM (2 papers) and VLDB (2 papers). The selected documents contain overall 255 distinct recognised *dataset* entities before filtering, and 85 distinct recognized *method* entities before filtering. The average number of times each selected paper has been

	Dataset (FP%)	Method (FP%)
User added	25.9%	11.7%
NER extracted	94.3%	57.9%
Total	**80.4%**	**27.4%**

Table 1: Comparison of false positive rates, resulting from majority vote on relevance of unfiltered extracted entities for both user added and NER extracted entities

cited is 581. The 10 human evaluators are randomly and uniformly assigned to the documents such that each document is processed by at least 3 evaluators. Note that users could add new entities, increasing the number of distinct entities. The evaluators showed quite varying task completion times for giving feedback on all entities contained in a document, with an average of 7:57 minutes to provide feedback for a single document, while the fastest evaluator only needed 3:14 minutes and the slowest 19:38 minutes.

Entities and Agreement: The evaluators were not forced to rate all occurrences of recognized entities (the assignment was: "provide feedback on the recognized entities as you see fit."). The average percentage of recognized entities (highlighted in the Coner Viewer) each evaluator gave feedback on is 65.9%. There were no discernible differences between *dataset* and *method* entities. After the experiment we interviewed the evaluators on their reasons for skipping feedback: First, ambiguous meanings of the same entities annotated in different sections and contexts caused doubt about type relevance (e.g. the named entity `Microsoft` can reference a dataset created by Microsoft or the actual company itself). Second, some bigram or trigram *method* entities were recognized with additional useless trailing words (e.g. *question taggings have*), therefore also not receiving feedback from some evaluators.

Table 1 compares the percentage of *dataset* and *method* entities that where considered correct by the TSE-NER classifier (i.e. without the filtering step) or manually added by an evaluator, but judged as incorrect by the majority of evaluators. The false positive rates in Table 1 indeed show the effectiveness of collaborative feedback on TSE-NER. Interestingly, not all of the named entities added by users were rated as relevant for their intended type; for user added entities, we observe a false positives rate of 25.9% for *dataset* and 11.7% for *method*. This means that it is crucial to also receive user feedback from evaluators on entities other users added to ensure the quality of human feedback. Evaluators differ in skill, expertise, and also effort they put into feedback, which clearly influences their decision making.

We calculated the average Cohen's Kappa between the 10 evaluators for each entity type. On average, Cohen's Kappa for *dataset* entities is 0.51, while for *method* entities it is 0.63.

Comparison Filtering Techniques: Coner vs TSE-NER
Table 2 compares the performance of Coner human feedback filtering and different filtering heuristic setups for TSE-NER in terms of retention rate; the percentage of unfiltered extracted entities kept by each filter. The different filtering techniques were performed on the complete set of entities that received

	PMI	WS	ST	KBL	EMV	FCB
Dataset	9.0%	86.9%	34.4%	90.7%	35.0%	19.5%
Method	9.4%	73.7%	69.0%	81.2%	41.6%	52.2%

Table 2: Comparison of entity retention rate between Coner and TSE-NER filter techniques (315 entities for *dataset* and 198 entities for *method*). Filtering acronyms: Pointwise Mutual Information (PMI), Wordnet + Stopwords (WS), Similar Terms (ST), Knowledge Base Look-up (KBL), Ensemble Majority Vote (EMV), Filtering Coner Boost (FCB): EMV + Coner Human Filtering

	PMI	WS	ST	KBL	EMV	FCB
Dataset	38.7%	73.9%	79.7%	79.4%	76.7%	8.8%
Method	25.0%	28.2%	40.3%	37.7%	37.7%	3.9%

Table 3: Percentage of false positives in the remaining filtered entity sets of TSE-NER filtered heuristics compared to Coner human filtered entities. Filtering acronyms same as Table 2

feedback from at least three evaluators in the 10 selected papers; 315 *dataset* and 198 *method* entities. As illustrated in Table 2, the Coner Boost (FCB) filtering technique described in this paper is more strict than Ensemble Majority Vote originally used by TSE-NER for the *dataset* type, but less strict for the *method* type. This can be explained by the larger percentage of user added named entities for the *method* type compared to the *dataset* type, with user added named entities having a much lower average false positive rate compared to NER extracted entities (Table 1).

To get a better insight into the filtering performances, we compared the false positives rate for each filtering technique with regards to the set of entities determined to be relevant by human evaluators (Table 3); if an entity is kept by a filter for a type, but was voted as irrelevant for a type by the majority of evaluators, then it is considered a false positive instance. For most of the TSE-NER filtering setups the average false positives rate for both facets is above 50% (only PMI has a lower false positive rate, because it is much more selective in its retention of entities). This means there are a significant number of entities that were recognised as irrelevant for a type by human judgement, but TSE-NER heuristic filtering was unable to do so.

We also considered the false negatives which were excluded by the filtering techniques but were labelled as relevant by majority of evaluators (Table 4). The PMI filtering as explained in [12] achieved the highest precision among the TSE-NER filtering techniques in their evaluation. Table 4 clearly indicates a major shortcoming of the PMI filtering heuristic; it filters out on average 82.2% of Coner viewer entities that were rated as true positives by Coner human feedback. Even for the EMV filtering heuristic, which is regarded as most effective in terms of F-score by [12], the average false negatives rate is 60.8%. Also, in Table 2 we see that KBL has the highest average retention rate of named entities, which also translates in a high false positive rate and lower false negatives rate.

Finally, Table 3 and Table 4 demonstrate that the FCB filtering approach results in the lowest false positives and false negatives rates compared to Coner human filtering; this is good for the quality of filtered entities, because more

	PMI	WS	ST	KBL	EMV	FCB
Dataset	76.2%	3.8%	70.0%	20.0%	65.0%	0.0%
Method	88.2%	4.6%	30.9%	15.1%	56.6%	1.3%

Table 4: Percentage of false negatives in the remaining filtered entity sets of TSE-NER filtered heuristics with regards to Coner filtered entities.

relevant named entities overlap with the Coner human filtering (regarded as true positives), but it also means it difficult to scale this approach with a significantly larger number of named entities.

Qualitative Entity Inspection: When there is a user consensus, Coner removes or adds entities to the TSE-NER expansion and filter phases, effectively overwriting the heuristics. We manually inspected some of these entities to obtain an intuition on what entities the TSE-NER heuristics usally fail at. Table 5b shows some randoms sample entities which have been consensually labeled as wrong with respect to the recognized type, while table 5a shows entities which are labeled as correct. Table 6 shows some samples which failed to obtain user concensus and obtained a mix of positive and negative labels.

Dataset	`digg, flickr, wikipedia, datasets`
Method	`hybrid multimodal method, similarity search, reinforcement learning, logistic regression, acyclic subgraph`

(a) Correct

Dataset	`digg interfaces, logistic regression, acyclic subgraph`
Method	`digg, flickr, wikipedia, dynamic programming, system description signed clustering`

(b) Incorrect

Table 5: Examples of *Dataset* and *Method* entities annotated as correct or incorrect.

For example users seem to be uncertain and fail to reach consensus when entities are related to a type but are too generic, e.g. `signed networks, news article, news feed, data base` for *dataset* and `algorithm, decision rule and used search algorithm` for *method* entity type. This could be explained by a difference in domain expertise or interpretation of what belongs to a certain type between evaluators. This shows that even for humans, reliably typing entities is hard as there is quite some room for subjective interpretation.

Also, during our inspection, we encountered frequently entities which are classified both as *method* and *dataset* by TSE-NER like `digg, flickr, wikipedia, logistic regression, acyclic subgraph`. Most of these double classifications are wrong, and we will further investigate this double classification phenomenon in Section 4.2.

Dataset	`signed networks, slash, data base news article, news feed`
Method	`10-foldcross validation, algorithm, decision rule, used search algorithm, vldb, web services`

Table 6: Sample of *Dataset* and *Method* annotated without clear user consensus

4.2 Experiment 2: NER Performance

We picked 28 papers from 4 conferences in our document corpus, similarly to our document selection described in section 4.1; 13 papers from VLDB, 9 papers from The Web Conference, 4 from SIGIR and 2 from ICWSM.

We recruited 15 graduate-level/post-graduate-level volunteers and instructed them to focus their efforts on judging entities recognized in these papers. However, for this experiment we want to make sure that user feedback is as effective as possible to use our human annotators time efficiently. As a heuristic we focus on entities which have been double-classified as both *dataset* and *method*, and thus one of the types is nearly guaranteed to be wrong. As mentioned before, double classifications between *dataset* and *method* are quite common. This can be explained by the relative similarity of these two types: both types appear in similar contexts and/or sentence structures, and are much closer to each other than typical entities types considered in NER like *location* and *person*. Thus, distinguishing between *dataset* and *method* can be considered a very hard task for an automatic classifier. Cases like these is when user feedback are the most valuable.

In order to measure the effect of human feedback into the TSE-NER filtering, we repeat the experiments described in [12] and use the same test set, measuring the F-score, precision, and recall with and without the Coner feedback. We used the output of the experiment and the TSE-NER to train the NER model. For training we used 71,292 and 103,568 (i.e, *dataset* and *method* entity type) sentences for TSE-NER and 25,819 and 53,200 (i.e, *dataset* and *method* entity type) sentences for Coner and employed the SE strategy. For testing, 3149 sentences were used for dataset and 1097 sentences for method entity type.

Table 7 compares the performance of TSE-NER with and without Coner feedback focused on double-classified entities in terms of precision, recall and F-score. As shown in Table 7 there is an increase in precision for both *dataset* and *method* type classifiers when incorporating user feedback with Coner, while recall and F-score remains stable. Naturally, providing feedback on recognized entities as part of the filter step cannot increase recall, but only affect precision by removing *false positives*. Overall, the test dataset covered 555 unique entities, and we obtained user feedback on 29 unique entities of the test set. Nonetheless, this shows that by focusing user feedback on parts which are in doubt, like the double-classified entities, even a smaller number of user feedback can make a difference, i.e. by obtaining feedback on only 0.05% of the entities in the test set we could increase the precision by 4%. This significant increase in precision is mainly due to the fact that user feedback improves the quality of the input data

	Dataset (P/R/F)	Method (P/R/F)
TSE-NER	0.66/0.60/0.63	0.56/**0.21**/0.30
Coner	**0.70**/0.60/**0.65**	**0.59**/0.20/0.30

Table 7: Comparison of performance of *TSE-NER* and *Coner* in terms of Precison/Recall/F-score for two type of doubly filtered entities: *Dataset* and *Method*

for each training iteration of TSE-NER, thus the effect of each feedback is greatly magnified. In a scenario where Coner is constantly running in the background, we expect notable increases both for precision and recall (due to allowing users to suggest new entities).

5 Related Work

A considerable amount of literature published in recent years addressed the *deep analysis* of text such as topic modelling, domain-specific entity extraction, etc. Common approaches for *deep analysis* of publications rely on techniques such as dictionary-based [17], rule-based [4], machine-learning [16] or hybrid (combination of rule-based and machine-learning) [19] techniques. Despite its high accuracy, a major drawback of dictionary-based approaches is that they require an exhaustive dictionary of domain terms. These dictionaries are often too expensive to create for less relevant domain-specific entity types. The same holds for rule-based techniques, which rely on formal languages to express rules and require comprehensive domain knowledge and time to create. The lack of large collections of labelled training data and the high cost of data annotation for a given domain is one of the main issues of machine-learning approaches. Many attempts have been made to reduce annotation costs such as bootstrapping [18] and entity set expansion [3,6] which rely only on a set of seed terms provided by the domain expert. Unfortunately, this reliance on weak supervision just providing seed terms limited also the maximal achievable performance with respect to precision, recall, and F-scores.

Active learning (AL) is another technique that has been proposed in the past few years, asking users to annotate a small part of a text for various natural language processing approaches [15,20,5] or generating patterns used to recognize entities [8]. With active learning, the unlabelled instances are chosen intelligently by the algorithm (e.g. least confidence, smallest margin, informativeness, etc) for annotation. However the AL techniques are heavily influenced by the quality of the previous labeled examples and its reliability suffers when dealing with noisy training data generated in a semi-automatic fashion. Our approach on the other hand relies on training NER algorithms for long-tail entities in a weakly-supervised fashion which incrementally incorporates human feedback on the relevance of extracted entities with high expected information gain into the training cycle. In addition, in contrast to [5] where the authors just present bibliographic sentences to Amazon Mechanical Turk annotators for labelling, our work focuses on the annotation of long-tail entities which relies on the occurrence context for easier annotation. We incorporate collaborative user feedback

on type relevance of classified entities and annotation of new entities to continuously support the sentence expansion and entity filtering steps of the iterative TSE-NER algorithm [12]. Newly annotated relevant domain specific entities are added to the seed set in the expansion step, to fetch additional relevant training sentences and terms to increase the number of true positive occurrences in the training data. Furthermore, we allow to filter out irrelevant entities in the filtering step, to reduce the number of false positives detected by the noisy NER.

6 Conclusion and Future Work

In this work we focused on augmenting the filter step of TSE-NER by incorporating user feedback into the NER training process. Our lab experiments showed that 94.3% for *dataset* and 57.9% for *method* of entities detected by partial TSE-NER without heuristic filtering were indeed false positives. We observed that by using different filtering heuristics we can reduce the number of false positives up to 38.7% for *dataset* and 25% for *method* (i.e. using the PMI filtering heuristic) which also results in higher false negatives rate as shown in Table 4. In order to reduce the number of false positives as well as false negatives we proposed incorporating user feedback into filtering which resulted in the lowest false positives (i.e. 8.8% for *dataset* and 3.9% for *method*) and false negatives (i.e. 0.0% for *dataset* and 1.3% for *method*). Furthermore we showed that by obtaining feedback on only 0.05% of the entities in the test set (and others outside the set), we could increase the precision by 4% while keeping recall and f-score stable.

For future work, we can leverage Coner's full potential by integrating it into an existing production system, like a large scale digital library. In this case we can receive continuous feedback from the system's users on a number of papers magnitudes bigger than our private lab experiment conducted so far and improve the performance of the NER models over time. Likely, user feedback techniques usable for term expansion will require a heavier toll, and thus need further investigation. To a certain extend, this could be offset using appropriate *incentivation* techniques: by motivating user to be willing to contribute feedback (for example by means of gamification), even more elaborate feedback mechanisms could be employed without degrading user satisfaction.

References

1. Abekawa, T., Aizawa, A.: Sidenoter: Scholarly paper browsing system based on pdf restructuring and text annotation. In: COLING (Demos). pp. 136–140 (2016)
2. Aizawa, A.: Pdfnlt. `https://github.com/KMCS-NII/PDFNLT` (2018)
3. Brambilla, M., Ceri, S., Della Valle, E., Volonterio, R., Acero Salazar, F.X.: Extracting emerging knowledge from social media. In: Proceedings of the 26th International Conference on World Wide Web. pp. 795–804. International World Wide Web Conferences Steering Committee (2017)

4. Eftimov, T., Seljak, B.K., Korošec, P.: A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. PloS one **12**(6), e0179488 (2017)
5. Goldberg, S.L., Wang, D.Z., Kraska, T.: Castle: crowd-assisted system for text labeling and extraction. In: First AAAI Conference on Human Computation and Crowdsourcing (2013)
6. Kejriwal, M., Szekely, P.: Information extraction in illicit web domains. In: Proceedings of the 26th International Conference on World Wide Web. pp. 997–1006. International World Wide Web Conferences Steering Committee (2017)
7. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of NAACL-HLT. pp. 260–270 (2016)
8. Marrero, M., Urbano, J.: A semi-automatic and low-cost method to learn patterns for named entity recognition. Natural Language Engineering pp. 1–37 (2017)
9. Mathew, G., Agarwal, A., Menzies, T.: Trends in topics at SE conferences (1993-2013). arXiv preprint arXiv:1608.08100 (2016)
10. Mesbah, S., Fragkeskos, K., Lofi, C., Bozzon, A., Houben, G.J.: Facet embeddings for explorative analytics in digital libraries. In: Int. Conf. on Theory and Practice of Digital Libraries (TPDL). Thessaloniki, Greece (sep 2017)
11. Mesbah, S., Fragkeskos, K., Lofi, C., Bozzon, A., Houben, G.J.: Semantic annotation of data processing pipelines in scientific publications. In: European Semantic Web Conference. pp. 321–336. Springer (2017)
12. Mesbah, S., Lofi, C., Torre, M., Bozzon, A., Houben, G.J.: Tse-ner: An iterative approach for long-tail entity extraction in scientific publications. In: 17th International Semantic Web Conference. pp. 127–143. Springer (2018)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
14. Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., Ponzetto, S.P.: A large database of hypernymy relations extracted from the web. In: LREC (2016)
15. Shen, D., Zhang, J., Su, J., Zhou, G., Tan, C.L.: Multi-criteria-based active learning for named entity recognition. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. p. 589. Association for Computational Linguistics (2004)
16. Siddiqui, T., Ren, X., Parameswaran, A., Han, J.: Facetgist: Collective extraction of document facets in large technical corpora. In: Int. Conf. on Information and Knowledge Management. pp. 871–880. ACM (2016)
17. Song, M., Yu, H., Han, W.S.: Developing a hybrid dictionary-based bio-entity recognition technique. BMC medical informatics and decision making **15**(1), S9 (2015)
18. Tsai, C.T., Kundu, G., Roth, D.: Concept-based analysis of scientific literature. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. pp. 1733–1738. ACM (2013)
19. Tuarob, S., Bhatia, S., Mitra, P., Giles, C.L.: Algorithmseer: A system for extracting and searching for algorithms in scholarly big data. IEEE Transactions on Big Data **2**(1), 3–17 (2016)
20. Wang, A., Hoang, C.D.V., Kan, M.Y.: Perspectives on crowdsourcing annotations for natural language processing. Language resources and evaluation **47**(1), 9–31 (2013)