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

Daniel Vliegenthart, Sepideh Mesbah, Christoph Lofi, Akiko Aizawa
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}@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 a iterative distantly-supervised fashion, thus limiting human interaction to only providing a small set of seed terms. Such approaches heavily rely 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, therefore allowing to still train new long-tail NER extractors with low costs, but with ever increasing performance while the algorithm is actively used.

Keywords: Information Extraction, Named Entity Recognition, Document Metadata, Long-Tail Entity Types, Human Feed-back

1. Introduction

With the ever 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 (Mathew et al., 2016), but also for techniques which can provide effective retrieval and search experiences. To this end, *"deep meta-data"* extracted from scientific publication – i.e. meta-data able to represent domain-specific aspects (*facets*) in which a document can be understood within its (research) domain – allows for novel exploration capabilities (Mesbah et al., 2017b).

Domain-specific typed named entities (Mesbah et al., 2017c) are a representative example of deep meta-data. Consider the domain of *data processing and data science*, which is currently popular due to its real-life implications on machine learning algorithms and data-centric business models. In this domain, the main entity types of interests to the user base of a scientific collection would for example be: *datasets* used in a given publication; the *methods* applied to the data or used in implementation; or *software packages* realizing these methods (Mesbah et al., 2017a).

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. locations and age) (Kejriwal and Szekely, 2017), 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. Few approaches addressed this problem by relying on bootstrapping (Tsai et al., 2013) or entity expansion (Brambilla et al., 2017; Kejriwal and Szekely, 2017) 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.

In our previous works, we introduced TSE-NER, an approach for sentence classification and named entity extraction using distant supervision (Mesbah et al., 2017c), and extended it for recognizing facets relevant to academic literature and data processing (Mesbah et al., 2017b). At its core, this approach is iterative, with each iteration starting with a set of known instances of defined types. These sets are then heuristically expanded to train a new NER classifier, and heuristically filtered to remove likely false positives to create the entity set for the next iteration. As experiments in (Mesbah et al., 2018) 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.

The core goal of this paper is to extend TSE-NER with incremental, collaborative feedback from human contributors, designed to support the filter phase of the algorithm. The human-in-the-loop approach allows us to still maintain the advantages of our initial design (i.e., training a NER algorithm cheaply, only relying on a small seed set, and providing an immediate result to users which acceptable extraction quality discussed in (Mesbah et al., 2018)), while benefiting from additional intelligence in the process.

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. This feedback is then exploited into the next NER training iteration, thus allowing the system to improve its performance over time with active use.

The contribution of this paper are as follows:

- We describe `Coner`, an extension for TSE-NER which incorporates collaborative user feedback for continuously supporting its entity filtering step.
- We evaluate our approach on a collection of 11,589 data science publications from ten conference series. We show that even limited feedback can significantly

improve the quality of the entity filtering step. An exploratory experiment performed on 3 papers and with 10 users shows that by utilizing human feedback, up to **55.6%** of false positives can be detected for the *dataset* entity type and **11.1%** for the *method* entity type, with an average per-document annotation time just below 8 minutes per user.

The remainder of this paper is structured as follows: section 2. provides an overview of our distantly-supervised long-tail entity extraction approach, section 3. introduces our novel extension for continuous collaborative human-feedback, while in section 4. we evaluate the effectiveness of this approach. We conclude with discussions on related works (section 5.) and an outlook of our future works (section 6.).

2. TSE-NER: Distantly Supervised Long-tail NER

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 (Mesbah et al., 2018). 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 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 a 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. Training Data Extraction

In the first step, a set of training data sentences is created by extracting all the sentences containing any of the seed terms. In the first iteration, the seed term set can contain from 5 to 50 terms, that are provided manually by expert users at a very low cost (arguably, any expert in a domain can name more than 5 examples of a named entity).
As an example of this step, consider the word "Letor" (i.e., an entity of dataset type) in the seed term list. All sentences in the containing the word "LETOR" in the corpus, such as *"We performed a systematic set of experiments using the LETOR benchmark collections OHSUMED, TD2004, and TD2003"* are extracted, and provide as examples of the positive classification class. We also extract surrounding sentences in the text to better capture the usage context of the seed entity.

2.2. Expansion

As seen in the sentence example provided in the previous section, also OHSUMED, TD2004 and TD2003 are identified as belonging to the dataset entity type, but since they are not in our seed terms we will label them negatively – thus leading to more false negatives. At the same time, the extraction of sentences in the training data that are related to seed terms will cause a shortage of negative examples for training purposes. In order to avoid these problem we introduced the *term expansion* and *sentence expansion* strategies

2.2.1. Term Expansion

Term Expansion is designed to reduce the number of false negatives in the training sentences and provide more positive examples. In this work we use *semantic relatedness*: terms which are semantically similar or related to terms in the seed list should be included in the expansion. For example, given the dataset seed term LETOR , the expansion should add semantically related terms like OHSUMED or TD2004 which are also benchmarks used in the field of information retrieval. We first trained the *word2vec* model (Mikolov et al., 2013) on our whole corpus by learning all uni- and bi-gram word vectors of all terms in the corpus. Then, we use NLTK entity detection to obtain a list of all entities contained in the sentences of the training data and cluster them with respect to their embedding vectors using K-means clustering. Silhouette analysis is used to find the optimal number k of clusters. Finally, clusters that contain at least one of the seed terms are considered to contain entities of the same type (e.g *Dataset*)

2.2.2. Sentence Expansion

Sentence Expansion (SE) strategy is designed to addresses the problem of the over-representation of positive examples and to increase the size and variety of the training set. The goal of this step is to include sentences that are similar in semantics and vocabulary to the original training sentences, and are unlikely to contain instances of the desired type, to serve as informative negative examples for boosting the NER training accuracy. We first use the *doc2vec* document embeddings (Le and Mikolov, 2014), to learn vector representations of the sentences in the corpus. For each sentence in the training data, we use *doc2vec* to discover the most similar sentence which does not contain any known instance of the targeted type (i.e., expanded terms).

2.3. Training Data Annotation

After obtaining an expanded set of *seed terms* and *training sentences*, if any of the words in the *seed terms* matches a word in the *training sentences*, the word will be labeled positively. The annotated dataset can be used as an input to train any state-of-the-art supervised NER algorithm

2.4. NER Training

For training a new NER, we used the Stanford NER tagger[1] to train a Conditional Random Field (CRF) model. CRF is to learns the hidden structure of an input sequence by defining a set of feature functions (e.g. word features,

[1] https://github.com/dat/stanford-ner

current position of the word labels of the nearby word), assigning them weights and transforming them to a probability to detect the output label of a given entity.

2.5. Filtering

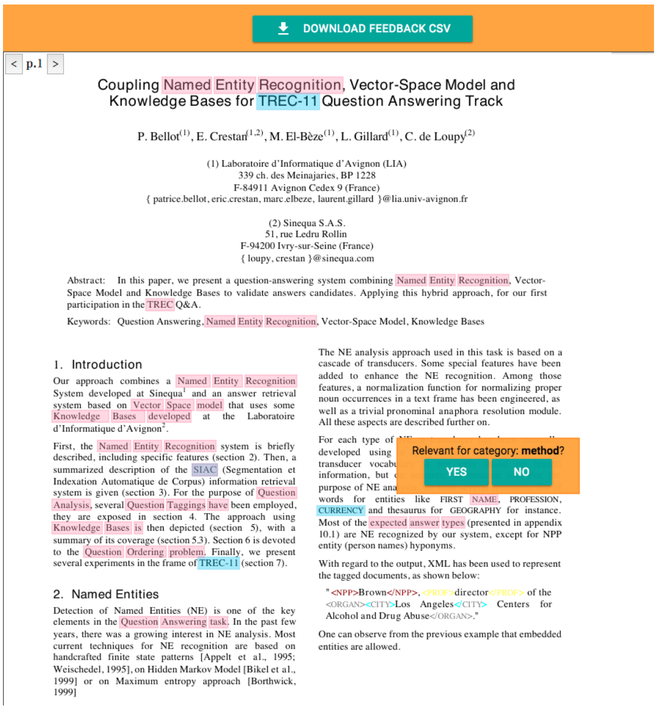
In this final step, which is also the focus of this work, we use the trained NER model to annotate the whole corpus and consider all the positively annotated terms as candidate terms for the next round of iteration. As we are using noisy training data to train our NER, the list of entities extracted 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 four basic heuristics, each relying on a different underlying assumptions: 1) filtering stopwords (e.g. something); 2) concepts coming from "common" English language (e.g., "dataset", "software") that could be found in Wordnet[2]; 3) exclude the entities that have a reference in the DBpedia knowledge base (under the assumption that, if they are mentioned in DBpedia, then they are not from the sought for type); and 4) exclude the entities that do not appear in the same cluster that contains a seed term - i.e. explained in 2.2.1.. Interested readers can refer to (Mesbah et al., 2018) for detailed explanation.

As those heuristics are rather basic in their nature, we discuss in the next section of filtering can be supported by human feedback.

3. Collaborative Feedback with Coner

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 happen easily as the used training data is noisy due to the strong reliance on heuristics). While we have shown in (Mesbah et al., 2018) that this filter step indeed increases the precision of the overall approach, it does also impact the recall negatively (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 (for example, this could 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 *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.

Both 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 our 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.



Coupling Named Entity Recognition, Vector-Space Model and
Knowledge Bases for TREC-11 Question Answering Track

P. Bellot(1), E. Crestan(1,2), M. El-Bèze(1), L. Gillard(1), C. de Loupy(2)

(1) Laboratoire d'Informatique d'Avignon (LIA)
339 ch. des Meinajaries, BP 1228
F-84911 Avignon Cedex 9 (France)
{ patrice.bellot, eric.crestan, marc.elbeze, laurent.gillard }@lia.univ-avignon.fr

(2) Sinequa S.A.S.
51, rue Ledru Rollin
F-94200 Ivry-sur-Seine (France)
{ loupy, crestan }@sinequa.com

Abstract: In this paper, we present a question-answering system combining Named Entity Recognition, Vector-Space Model and Knowledge Bases to validate answers candidates. Applying this hybrid approach, for our first participation in the TREC Q&A.

Keywords: Question Answering, Named Entity Recognition, Vector-Space Model, Knowledge Bases

1. Introduction

Our approach combines a Named Entity Recognition System developed at Sinequa[1] and an answer retrieval system based on Vector Space model that uses some Knowledge Bases developed at the Laboratoire d'Informatique d'Avignon[2].

First, the Named Entity Recognition system is briefly described, including specific features (section 2). Then, a summarized description of the SIAC (Segmentation et Indexation Automatique de Corpus) information retrieval system is given (section 3). For the purpose of Question Analysis, several Question Taggings have been employed, they are exposed in section 4. The approach using Knowledge Bases is then depicted (section 5), with a summary of its coverage (section 5.3). Section 6 is devoted to the Question Ordering problem. Finally, we present several experiments in the frame of TREC-11 (section 7).

2. Named Entities

Detection of Named Entities (NE) is one of the key elements in the Question Answering task. In the past few years, there was a growing interest in NE analysis. Most current techniques for NE recognition are based on handcrafted finite state patterns [Appelt et al., 1995; Weischedel, 1995], on Hidden Markov Model [Bikel et al., 1999] or on Maximum entropy approach [Borthwick, 1999]

The NE analysis approach used in this task is based on a cascade of transducers. Some special features have been added to enhance the NE recognition. Among those features, a normalization function for normalizing proper noun occurrences in a text frame has been engineered, as well as a trivial pronominal anaphora resolution module. All these aspects are described further on.

For each type of NE, a transducer has been manually developed using [illegible] transducer vocabulary is not only based on lexical information, but on semantic information too. For the purpose of NE analysis, we built several resources: list of words for entities like FIRST NAME, PROFESSION, CURRENCY and thesaurus for GEOGRAPHY for instance. Most of the expected answer types (presented in appendix 10.1) are NE recognized by our system, except for NPP entity (person names) hyponyms.

With regard to the output, XML has been used to represent the tagged documents, as shown below:

"<NPP>Brown</NPP>, <PROF>director</PROF> of the <ORGAN><CITY>Los Angeles</CITY> Centers for Alcohol and Drug Abuse</ORGAN>."

One can observe from the previous example that embedded entities are allowed.

Figure 1: Coner Interactive Document Viewer with feedback buttons for entities highlighted with different colors for facets; light blue for *dataset*, pink for *method* and purple for entities classified for both *dataset* and *method* facets

As a new core component, we introduce an *interactive document viewer*, which users use to view the documents. The viewer is based on the NII PDFNLT (Abekawa and Aizawa, 2016) (Aizawa, 2018), which already included a PDF viewer and a sentence annotation tool.

Our document viewer highlights all detected named entities of the types relevant to the current domain, but also allows users to provide explicit feedback on recognized entities with respect to if they are indeed a correctly identified entities of that type or not (see Figure 1 for a screenshot of our document viewer).

One of our design goals for the interactive viewer component was to impose as little cognitive load on the system's users as possible, thus only very simple feedback mechanisms have been considered. In particular, we settled on providing a simple yes/no feedback option for each recognized named entity of a desired type. Users may use it to let the system know if the entity has been detected correctly or not. This design choice limits us to only elicit feedback on entities which have been detected by the NER (i.e. feedback for filtering), but does not allows us to find entities which have been missed completely (feedback for expansion). We will address this issue in our future works (see section 6.).

During the usage of the system, we collect all explicit user feedback on the correctness of the detected entities. This information is then used to periodically (e.g., every few hours or nightly) continue the training of the NER algorithm by executing more training iterations until convergence is reached again (see figure 2). However, for this new training cycle, manual feedback supersedes all heuris-

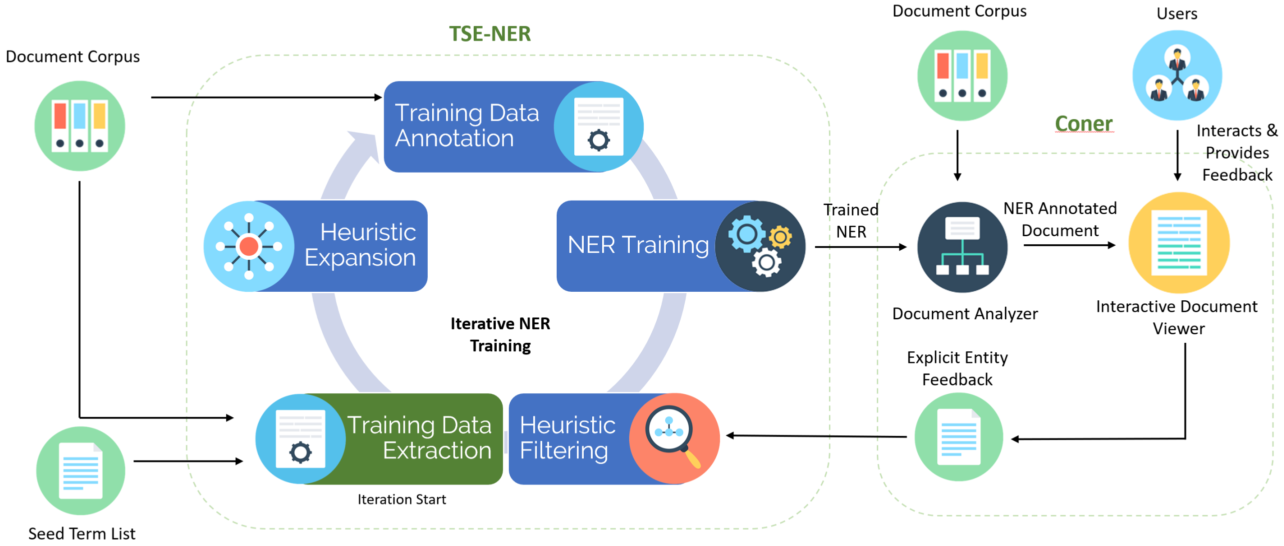


Figure 2: Overview of Coner Collaborative NER Pipeline: Human Feedback influences the TSE-NER filter phase, supporting or superseding heuristic decision making

tic decisions, thus entities are kept during filtering if they have been considered as being correct by majority of users, and those considered incorrect are discarded. Heuristics still apply for all entities without user feedback. We will provide more detailed information on how human feedback conflicts with heuristic decisions in the evaluation sections. The TSE-NER source code is openly available[4], as is the code of the Coner extension[5].

4. Evaluation

For evaluating our approach, we relied on the same corpus of 11,589 papers from 10 conferences on data science as already used in (Mesbah et al., 2018). We simulated interaction with the Coner system in a lab setting, recruiting 10 graduate-level / post-graduate-level volunteers knowledgeable in the data science domain.

After training TSE-NER on the training data analogously to (Mesbah et al., 2018), we annotated all corpus documents with recognizable *method* and *dataset* entities. We selected three documents with the highest combined number of recognized entities of both types for this evaluation, namely (Balog et al., 2010; Bellot et al., 2002; Losey et al., 2015). The 10 human evaluators are randomly and uniformly assigned to documents such that each document is processed by at least 3 evaluators ((Balog et al., 2010) has 4 evaluators). We asked the evaluators to check all recognized named entities of the *method* type or *dataset* type for correctness, or ignore them if they were unsure.

In this evaluation, we focus mainly on the following two research questions:

RQ1 What is the nature of the elicited human feedback? 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?

4.1. Coner Human Feedback

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

Documents and Evaluators: The three documents selected for this evaluation contain overall 57 distinct recognized *dataset* entities, and 74 distinct recognized *method* entities. We obtained at least one user feedback on the recognition correctness on 56 *dataset* entities (98.2%), and 67 *method* entities (91.8%). In average, each document contained 43.67 distinct entities of either type. 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, thus relevance feedback was only given when an evaluator was certain of his input. The average percentage of extracted entities (highlighted in the Coner Viewer) each evaluator gave feedback on in one paper is 83.7% for *dataset* entities and 83.5% for *method* entities. The fact that some highlighted entities were not rated by every evaluator is due to multiple factors. First, ambiguous meanings of the same entities annotated in different sections and contexts caused doubt about the rating (e.g. the entity *city* can indicate a database table attribute or a reference to an actual city), therefore feedback input was not always given on these entities. 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.

[4] https://github.com/mesbahs/TSE-NER

[5] https://github.com/vliegenthart/coner_collaborative_ner

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), but incorrect by the majority of evaluators. The false positve rates in Table 1 indeed show the effectiveness of collaborative feedback on TSE-NER. We further filtered the extracted entities from TSE-NER (i.e. here we used the PMI filtering explained in (Mesbah et al., 2018)) which achieved the highest precision among the filtering techniques), and compared the remaining entities with the labels provided by the majority of the evaluators to identify the percentage of the false positives in the remaining filtered set of TSE-NER. We also considered the false negatives which were excluded by the filtering step but were labelled as relevant by the evaluators. As shown in Table 2, by leveraging human feed-back in Coner, we can identify and exclude false positives while we can also keep the entities which were wrongly excluded from the TSE-NER filtering step (e.g. billion triple challenge corpus), thus leading to higher recall.

	Dataset (FP%)	Method (FP%)
(Balog et al., 2010)	75.0%	55.6%
(Bellot et al., 2002)	85.2%	57.1%
(Losey et al., 2015)	46.2%	65.5%
Total	68.8%	59.4%

Table 1: Comparison of false positive rates, resulting from users' majority vote on relevance of unfiltered extracted entities, for each paper for two types of entities: *Dataset* and *Method*

	Dataset	Method
False Positive	55.6%	11.1%
False Negative	10.5%	29.1%

Table 2: Percentage of the false positives and false negatives in the remaining filtered set of *TSE-NER* identified by the *Coner* for two types of entities: *Dataset* and *Method*

We measured inter-annotator agreement using Cohen's Kappa for all pairs of evaluators that labeled the same set of entities. In average, Cohen's Kappa for *dataset* entities is 0.36, while for *method* entities it is 0.32.

Qualitative Entity Inspection: As can be seen in Tables 3 and 4, *clueweb* and *dbpedia* were classified for both facets by TSE-NER, but with human feedback it becomes undeniably clear these entities belong to the *dataset* facet. Also, some entities recognized by the classifier as *method* or *dataset* names were actually names of for example cities, research teams, companies or universities. In a specific context, an occurrence of such a name could point to a method developed or dataset used by that team, but this is usually not that case. This statement is reinforced by the occurrence of entities like *florida*, *paypal* and *manatee (name of research team)* in table 3.

Dataset	`multimodal, bm25, purdue,`
	`pittsis, uamsterdam, depth`
	`country, planet, percentage`

Method	`urls, clueweb, dbpedia,`
	`florida, paypal, manatee,`
	`name`

Table 3: Dataset and Method annotated entities examples with *lowest relevance* scores from users

Dataset	`clueweb, wikipedia, dbpedia,`
	`siac, sandbox, blackhat`

Method	`hybrid multimodal method,`
	`similarity search,`
	`vector space model`
	`named entity recognition,`
	`semantic web crawl`

Table 4: Dataset and Method annotated entities examples with *highest relevance* scores from users

	Dataset (P/R/F)	Method (P/R/F)
TSE-NER	.64/**.41**/**.50**	.58/.21/.31
Coner	**.65**/.40/.49	**.62**/**.23**/**.33**

Table 5: Comparison of performance of *TSE-NER* and *Coner* in terms of Precison/Recall/F-Score for two type of entities: *Dataset* and *Method*

4.2. NER Performance

In this section, we repeat the experiments described in (Mesbah et al., 2018), measuring the F-Score, precision, and recall of TSE-NER with and without the Coner feedback on the three selected papers. We use the same test data set of manually annotated text snippets already employed in our previous works. The results are summarized in Table 5.

Table 5 compares the performance of TSE-NER with and without Coner feedback in terms of precision, recall and F-Score. For Coner we kept all the entities that were labelled by at least one evaluator as *relevant*. While the number of false positives decreased in Coner for both *dataset* (from TSE-NER: 653 to Coner: 629) and *method* entity types (from TSE-NER: 154 to Coner: 136), this could not be significantly reflected in the measured scores presented in Table 5. We attribute this to the shortcoming that the set of entities covered in the documents we chose for this evaluation (and thus received feedback for), and the entities in the test set for measuring performance are disjoint, and thus the elicited human feedback does not impact the measured scores notably.

In Table 6 we analyze the performance of Coner for the entities that were labelled by at least 2, 3 and 4 evaluators. As shown in Table 6 for the *dataset* entity type if we use the entities labelled by at least two evaluators the precision increases compared to using just the labels of one evaluator. For the *method* entity on the other hand the performance decreased. This can be due to lack of enough examples for the *method* entity type thus leading to decrease in true positives.

Entity Type	#Evaluators	Precision/Recall/F-Score
Dataset	1	.65/.40/.49
	2	.67/.42/.51
	3	.67/.42/.51
	4	.68/.42/.52
Method	1	.62/.23/.33
	2	.60/.16/.25
	3	.60/.22/.32
	4	.60/.19/.29

Table 6: Precision, Recall and F-Score using the agreement between different number of evaluators for two types of entities *Dataset* and *Method*

4.3. Limitations of the Evaluation

The evaluation we presented in the paper is supposed to provide an intuition about the effectiveness of collaborative feedback on NER extraction, and also shed some light on the limitations of the heuristics used in TSE-NER.

For a more extensive understanding, we would require a larger user study, preferable outside of a lab setting with real-life system users as described in section 3.. In particular, this study should cover more than 3 documents, and those documents should be selected by their representatives within the corpus (while in our current evaluation, we simply chose documents with the most recognized named entities). Despite these limitations, we believe that we could show that collaborative user feedback is indeed an effective tool for increasing NER performance.

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 (Song et al., 2015), rule-based (Eftimov et al., 2017), machine-learning (Siddiqui et al., 2016) or hybrid (combination of rule based and machine learning) (Tuarob et al., 2016) 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 labeled training data and the high cost of data annotation for a given domain is one of the main issues of machine learning approaches. In recent years, many attempts have been made to reduce annotation costs such as bootstrapping (Tsai et al., 2013) and entity set expansion (Brambilla et al., 2017; Kejriwal and Szekely, 2017) which rely only on a set of seed terms provided by the domain expert. Unfortunately, this reliance on very weak supervision (i.e. just providing the seed terms) limited also the maximal achievable performance with respect to precision, recall, and F-scores.

Active learning is another technique that have been proposed in the past few years, asking users to annotate a small part of a text for various natural language processing approaches (Shen et al., 2004; Wang et al., 2013; Goldberg et al., 2013) or generating patterns used to recognize entities (Marrero and Urbano, 2017). With active learning, the unlabeled instances are chosen intelligently by the algorithm (e.g. least confidence, smallest margin, informativeness, etc) for annotation. The proposed approach in this paper is inspired by active learning techniques (Shen et al., 2004; Wang et al., 2013; Goldberg et al., 2013) but relies on training NER algorithms for long-tail entities in a distantly-supervised fashion which incrementally incorporates human feedback on the relevance of extracted entities into the training cycle. In addition, in contrast to (Goldberg et al., 2013) were the authors just present bibliographic sentence to Amazon Mechanical Turk annotators for labelling, our work focuses on the annotation of long-tail entities which relies on the context for easier annotation. We incorporate collaborative user feedback for continuously supporting the entity filtering step of the iterative TSE-NER algorithm. We allow to filter out irrelevant entities, to reduce the number of false positives detected by the noisy NER.

6. Conclusion and Future Work

In this paper, we introduced Coner, a collaborative approach for long-tail named entity recognition in scientific publications. Coner extends TSE-NER, our previously established technique for iterative training of NER algorithms using distant supervision (Mesbah et al., 2018). In order to keep the training costs low, TSE-NER relied on heuristics to steer the training process (i.e., by expanding and filtering entity sets), requiring only on a small seed set of known named entities of the desired type as manual input. Unfortunately, this reliance on automatic heuristic expansion and filtering limited also the maximal achievable performance with respect to precision, recall, and F-Score.

We approached this problem with a unique solution: instead of requiring extensive manual input to initially train a NER algorithm upfront as most common state-of-the-art algorithms demand, we considered the synergy between NER training and the productive system it is employed in, including the respective user base. In particular, Coner allows us to mostly automatically train a NER algorithm at very low cost, and then exploit the daily user interaction for continuously improving the algorithm's performance, requiring only simple and intuitive feedback actions from the users. This is realized with an *interactive viewer component*, which allows users to elicit feedback on the correctness of recognized entities unobtrusively while reading the document.

In this first work incorporating the userbase into the NER training process, we focused on augmenting the filter step of TSE-NER, allowing manual user feedback to overrule decisions which would have been taken by the heuristics. In our lab experiments with a repository of 11,589 data science publications and 10 users, we could show that 67.5% of all entities detected by TSE-NER in the publications selected for evaluation were indeed false positives (and 49.5% for *dataset*). While these results emphasize the importance of incorporating human feedback into the NER training cycle, we could unfortunately not show the impact this feedback has on F-Score, precision, and recall when being integrated into the filter step, as the established benchmark dataset for this task and the entities in the publications chosen in this paper are mostly disjoint. This issue will be rectified in the near future, providing more tangible insights into the impact on classification performance.

Beyond those immediate challenges, several other interesting open challenges remain to be explored: We hypothesize that using user feedback for also *expanding the term set* (instead of only filtering) in each TSE-NER iteration should considerably increase the recall of the overall approach. However, one of the advantages of Coner is that its user feedback requires only little cognitive effort (i.e. simple yes / no feedback on recognized entities), and can be easily elicited while users use the system without much interruption. 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. However, as with all systems relying on crowdsourcing or explicit user feedback, *fraud* and *vandalism* become a central concern. If Coner is to be used with real-life users outside of a lab setting, such issues need to be addressed by for example user reputation management (Daniel et al., 2018) or different voting consensus techniques (El Maarry et al., 2015). Finally, in this work, we relied on our system's users to decide themselves on which entity to provide feedback on (users usually opted for providing feedback on nearly all entities). We speculate that NER training efficiency could be improved with the same amount of user feedback if we actively steer this selection process, e.g., by only asking for user feedback for entities with an *high expected information gain*, thus making the most out of each user interaction.

7. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 16H01756.

8. Bibliographical References

Abekawa, T. and Aizawa, A. (2016). Sidenoter: Scholarly paper browsing system based on pdf restructuring and text annotation. In *COLING (Demos)*, pages 136–140.

Aizawa, A. (2018). Pdfnlt. `https://github.com/KMCS-NII/PDFNLT`.

Balog, K., Serdyukov, P., and Vries, A. P. d. (2010). Overview of the trec 2010 entity track. Technical report, Norwegian University of Science and Technology Trondheim.

Bellot, P., Crestan, E., El-Bèze, M., Gillard, L., and de Loupy, C. (2002). Coupling named entity recognition, vector-space model and knowledge bases for trec 11 question answering track. In *TREC*.

Brambilla, M., Ceri, S., Della Valle, E., Volonterio, R., and Acero Salazar, F. X. (2017). Extracting emerging knowledge from social media. In *Proceedings of the 26th International Conference on World Wide Web*, pages 795–804. International World Wide Web Conferences Steering Committee.

Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B., and Allahbakhsh, M. (2018). Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)*, 51(1):7.

Eftimov, T., Seljak, B. K., and Korošec, P. (2017). A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PloS one*, 12(6):e0179488.

El Maarry, K., Güntzer, U., and Balke, W.-T. (2015). A majority of wrongs doesn't make it right-on crowdsourcing quality for skewed domain tasks. In *International Conference on Web Information Systems Engineering*, pages 293–308. Springer.

Goldberg, S. L., Wang, D. Z., and Kraska, T. (2013). Castle: crowd-assisted system for text labeling and extraction. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Kejriwal, M. and Szekely, P. (2017). Information extraction in illicit web domains. In *Proceedings of the 26th International Conference on World Wide Web*, pages 997–1006. International World Wide Web Conferences Steering Committee.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

Losey, R., Sullivan, J., Reichenberger, T., Kuehn, L., and Grant, J. (2015). e-discovery team at trec 2015 total recall track. In *TREC*.

Marrero, M. and Urbano, J. (2017). A semi-automatic and low-cost method to learn patterns for named entity recognition. *Natural Language Engineering*, pages 1–37.

Mathew, G., Agarwal, A., and Menzies, T. (2016). Trends in topics at SE conferences (1993-2013). *arXiv preprint arXiv:1608.08100*.

Mesbah, S., Bozzon, A., Lofi, C., and Houben, G.-J. (2017a). Describing data processing pipelines in scientific publications for Big Data injection. In *Workshop on Scholary Web Mining (SWM)*, Cambridge, UK, feb.

Mesbah, S., Fragkeskos, K., Lofi, C., Bozzon, A., and Houben, G.-J. (2017b). Facet Embeddings for Explorative Analytics in Digital Libraries. In *Int. Conf. on Theory and Practice of Digital Libraries (TPDL)*, Thessaloniki, Greece, sep.

Mesbah, S., Fragkeskos, K., Lofi, C., Bozzon, A., and Houben, G.-J. (2017c). Semantic annotation of data processing pipelines in scientific publications. In *European Semantic Web Conference*, pages 321–336. Springer.

Mesbah, S., Bozzon, A., Lofi, C., and Houben, G.-J. (2018). Long-tail entity extraction with low-cost supervision. *`https://2018.eswc-conferences.org/paper_8/`*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances* *in neural information processing systems*, pages 3111–3119.

Shen, D., Zhang, J., Su, J., Zhou, G., and Tan, C.-L. (2004). Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 589. Association for Computational Linguistics.

Siddiqui, T., Ren, X., Parameswaran, A., and Han, J. (2016). Facetgist: Collective extraction of document facets in large technical corpora. In *Int. Conf. on Information and Knowledge Management*, pages 871–880. ACM.

Song, M., Yu, H., and Han, W.-S. (2015). Developing a hybrid dictionary-based bio-entity recognition technique. *BMC medical informatics and decision making*, 15(1):S9.

Tsai, C.-T., Kundu, G., and Roth, D. (2013). Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1733–1738. ACM.

Tuarob, S., Bhatia, S., Mitra, P., and Giles, C. L. (2016). Algorithmseer: A system for extracting and searching for algorithms in scholarly big data. *IEEE Transactions on Big Data*, 2(1):3–17.

Wang, A., Hoang, C. D. V., and Kan, M.-Y. (2013). Perspectives on crowdsourcing annotations for natural language processing. *Language resources and evaluation*, 47(1):9–31.