Coner

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

Daniel Vliegenthart

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

by

Daniel Vliegenthart

in partial fulfilment of the requirements for the degree of

Master of Science
in Computer Science
Data Science & Technology Track
to be defended publicly on August 30, 2018

Delft University of Technology
Faculty of Electrical Engineering, Mathematics & Computer Science
Department of Software Technology
Web Information Systems Research Group

STUDENT NUMBER:
4217497

SUPERVISOR:
Dr. C. Lofi

THESIS COMMITTEE:
Prof.dr.ir. G.J.P.M. Houben
Dr. C. Lofi
Dr. S.T. Erdweg

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 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 thesis we introduce `Coner`: A collaborative approach to incrementally incorporate human feedback on the relevance of extracted entities into the training cycle of such iterative NER algorithms. 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. We do so by employing our intelligent entity selection mechanism that solely selects and visualises extracted entities with the highest potential knowledge gain from users interacting with them and providing feedback on facet relevance. Additionally, users can add new typed entities they deem relevant. Our Coner collaborative human feedback pipeline consists of three novel modules; a document analyser that extracts deep metadata from documents and selects a representative set of publications from a corpus to receive human feedback on, an interactive document viewer that allows users to give feedback on and add new typed entities simply by selecting the relevant text with their mouse and an explicit entity feedback analyser that calculates a facet relevance score through users' majority vote for each recognised entity. The resulting Coner entity facet relevance scores are then incorporated in the TSE-NER training cycle to boost the expansion and filtering heuristic steps. Remarkably, we revealed that even with limited availability of human resources we were able to boost TSE-NER's performance by up to 23.1% in terms of recall, up to 5.7% in terms of precision and the F-score with 13.1% depending on the setup of our smart entity selection mechanism and instructions given to evaluators.

Keywords: Information Extraction, Named Entity Recognition, Document Metadata, Long-Tail Entity Types, Human Feedback, Crowdsourcing

Acknowledgements

I would like to thank my supervisor at the Delft University of Technology, Christoph Lofi, for his exceptional guidance, mentoring and continuous critical feedback on my research progress. Also, I am incredibly thankful for the fruitful time I got to spend doing research on natural language processing as a member of Akiko Aizawa's lab at the National Institute of Informatics in Tokyo. I would like to express my very great appreciation to Aizawa-sensei for her excellent supervision. I can truly say that I have learnt more than I could imagine during my internship of 8 months in Tokyo, not just in terms of academics but also in irreplaceable life experience and self exploration.

I received an abundant amount of help and feedback from peers and researchers at TU Delft and NII, and for that I am very grateful. In particular I would like to express my gratitude to Sepideh Mesbah and Manuel Valle Torre for their advice and assistance in my research process.

A special thank you goes out to my family and girlfriend for always having my back and supporting me unconditionally, I couldn't have finished this thesis without you.

Daniel Vliegenthart
Katwijk, The Netherlands
August 24, 2018

Table of Contents

List of Figures

List of Tables

Acronyms

CDA Coner Document Analyser 17, 18, 45, 49, 52

CFA Coner Feedback Analyser 17, 30, 35, 49

CIDV Coner Interactive Document Viewer 17, 20, 21, 23, 24

EFRS Entity Facet Relevance Score 29, 30, 43

FCB Filtering Coner Boost 30, 40, 42, 50

SE Sentence Expansion 14

TE Term Expansion 13

TSECB Term Sentence Expansion Coner Boost 30, 42

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 [24], but also for techniques which can provide effective retrieval and search experiences. To this end, *"deep metadata"* extracted from scientific publication, i.e., metadata able to represent domain-specific aspects (*facets*) in which a document can be understood within its (research) domain, allows for novel exploration capabilities [28], like faceted keyword publication search and trend exploration and visualisation of facets in digital libraries [44].

Domain-specific typed named entities [30] are a representative example of deep metadata. 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* realising these methods [26].

The availability of deep metadata about the facets (*datasets* and *methods*) would enable rich queries like; *Which research methods are trending in the current domain?*, *Discover state of the art methods for point of interest recommendation that have been applied to geo-located social media data with high accuracy results*, or *Which methods are commonly applied to the MovieLens dataset?* Figure 1.1 illustrates an example of an intelligent query that requires the availability of facets deep metadata in publications. In order to successfully retrieve the list of applied methods to the MovieLens dataset and publications in which these methods occur, it is crucial that named entities of types *method* and *dataset* are automatically extracted and stored for each uploaded publication.

However, extracting and typing of 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) [17], 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 bootstrappin (a technique to expand training data by generating synthetic duplicates of existing items) [38], or entity expansion [4, 17] techniques, achieving promising performance. Entity expansion relies on heuristics to enhance a set of extracted entities. However, how to train high-performance *long-tail* entity extraction and typing with minimal human supervision remains an open research question.

In previous work, TSE-NER is introduced, an approach for sentence classification and named entity extraction using distant supervision [30]. This approach was further extended to recognise facets relevant to academic literature and data processing [28]. At its core, this approach is iterative, with each iteration starting with a set of known instances of defined types. Creating these starting seed sets of known instances is designed to be quick and with minimum human effort; for each type it is sufficient to have one or two domain experts think of 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 [27] 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 thesis is to extend TSE-NER with incremental, collaborative feedback from human contributors. 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 and add entities that they deem relevant for a specific facet. 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 previous work's initial design (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 as discussed in [27]), while exploiting the human feedback into the next NER training iteration. Coner allows the TSE-NER system to improve its performance over time by benefitting from additional human intelligence in the training process.

In this thesis we focus on answering the following research questions:

– **Research Question 1 (RQ1)**: *What is the nature of elicited human feedback?* We want to unveil insights about user's behaviour when interacting with Coner, to get tangible results of how efficient and scalable human feedback is. The biggest bottleneck in our approach is the scarcity of human resources; time and motivation, thus it is crucial that each instance of human feedback leads to the maximum potential knowledge gain with regards to entity relevance (we evaluate the knowledge gain by measuring the resulting TSE-NER's performance in RQ3). We measured the nature of human feedback for example with average time spend to give feedback on a single entity, inter-annotator agreement between pairs of users and how often users change their opinion about an entity's relevance.
– **Research Question 2 (RQ2)**: *In how far does human feedback confirm or conflict with TSE-NER heuristics?* It is crucial that this preliminary research question is answered before we can do any type of performance comparison. We hypothesise that if only a miniscule conflict between human feedback and TSE-NER heuristics is observed, then Coner will most likely not have a noticeable positive boost of TSE-NER performance. In our evaluation we will measure the similarities between entity retention rate, false positive and false negative rates between different filtering setups with and without incorporated human feedback.
– **Research Question 3 (RQ3)**: *How does incorporating human feedback on entities into the TSE-NER expansion and filtering steps improve the overall performance with respect to precision, recall, and F-measures?* The main goal of Coner is to boost TSE-NER's performance by utilising humans' superior ability to judge relevance of entities in their full context compared to machines, thus we hypothesise that Coner will indeed improve the performance measures and help improve faceted keyword search in digital libraries.

To answer these research questions, the following original contributions are made in this thesis:

1. We describe `Coner`, an extension for TSE-NER which incorporates collaborative user feedback for continuously supporting its term expansion and entity filtering steps. The Coner pipeline consists of three novel modules; a corpus document analyser that analyses each document and generates the necessary information to visualise entity occurrences (and filters documents on domain representativeness metrics for lab experiments), a document annotation viewer that visualises named entities and

allows users to interact with them, and a feedback analyser that calculates relevance scores for evaluated entities to be leveraged to boost TSE-NER heuristics.

2. We evaluate our approach on a collection of 11,598 data science publications from 11 conference series. We show that even limited feedback can significantly improve the quality of the entity expansion and filtering steps. Two experiments gave significant insight in how human feedback can be utilised to boost TSE-NER. The first exploratory experiment performed on 10 papers and with 10 users shows that by utilising 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, inter-annotator agreement of up to 0.61, an increase of recall of up to 23.1% and a boost of F-score of up to 13.1%. For the second experiment only entities with high expected information gain were selected by Coner to receive human feedback. Our smart entity selection mechanism only visualised entities that were classified by both facet NER classifiers. This resulted in an average per-entity annotation time of just above 15 seconds and an increase of precision of up to 5.7% by boosting the expansion and filtering steps of TSE-NER.

Preliminary versions of this thesis' contributions have been previously introduced in [40]; a workshop submission paper with me as lead author and several other authors. Parts of this thesis (related work in Section 2 and an overview of TSE-NER in Section 3) is based on the contents of this workshop paper.

The remainder of this thesis is structured as follows; related work is aggregated and discussed in Section 2 and Section 3 provides an overview of TSE-NER, the baseline distantly-supervised long-tail named entity extraction approach. Section 4 introduces our novel extension for continuous collaborative human-feedback called *Coner*, while in Section 5 we describe how Coner boosts the baseline algorithm's performance. In Section 6 we evaluate the effectiveness of our Coner novel extension in two experiments and discuss the meaning of our results. Finally we conclude in Section 7 and present an outlook of future work in Section 8.

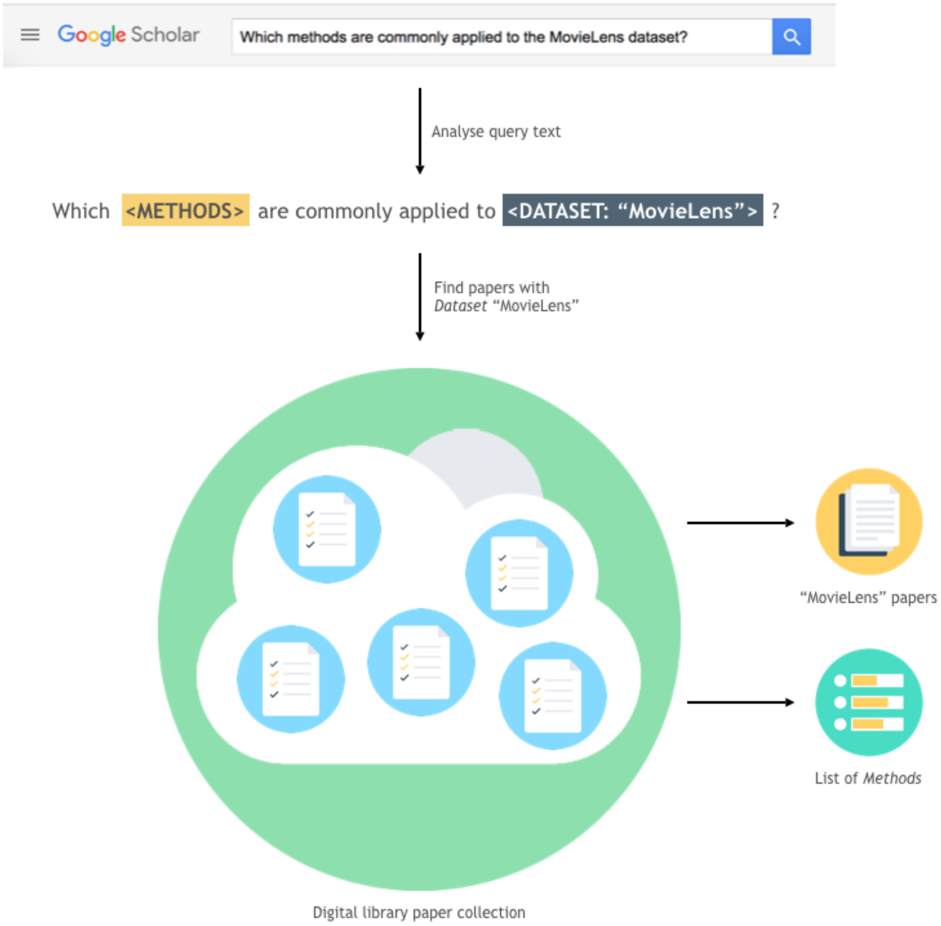

Google Scholar
Which methods are commonly applied to the MovieLens dataset?
Analyse query text
Which <METHODS> are commonly applied to <DATASET: "MovieLens"> ?
Find papers with
Dataset "MovieLens"
"MovieLens" papers
List of *Methods*
Digital library paper collection

Figure 1.1: An example of how *deep metadata* can be used to construct an intelligent query utilising the *method* and *dataset* facets

2 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 [37], rule-based [9], machine-learning [36] or hybrid (combination of rule based and machine learning) [39] 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. In recent years, many attempts have been made to reduce annotation costs such as bootstrapping [38] and entity set expansion [4, 17] 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 has been proposed in the past few years, asking users to annotate a small part of a text for various natural language processing approaches [35, 41, 13] or generating patterns used to recognize entities [23]. With active learning, the unlabelled instances are chosen intelligently by the algorithm (e.g. least confidence, smallest margin, informativeness, etc) for annotation. Furthermore, combining an active learning approach with uncertainty sampling as retraining annotation selection method has been widely researched [43, 21, 46, 45, 34].

The proposed approach in this paper is inspired by active learning techniques [35, 41, 13] 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 with high expected information gain into the training cycle. In addition, in contrast to [13] 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 [27]. 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.

The unique challenges of crowdsourcing user feedback have been widely investigated. First, proper incentivisation mechanisms for truthful evaluation and annotation are essential to ensure feedback quality [16, 8]. Also, task formulation should be made with fraudulent workers in mind [10, 12].

3 TSE-NER: Distantly Supervised Long-tail NER

In this section we will summarise TSE-NER, the iterative five-step low-cost approach for training NER/NET classifiers for long-tail entity types. TSE-NER was built by Sepideh Mesbah et al. and introduced, evaluated and discussed in in [27]. For more detailed information on this approach, refer to [27].

The approach is summarised in the following five steps, as can be seen in Figure 3.1:

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 Recogniser* (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.

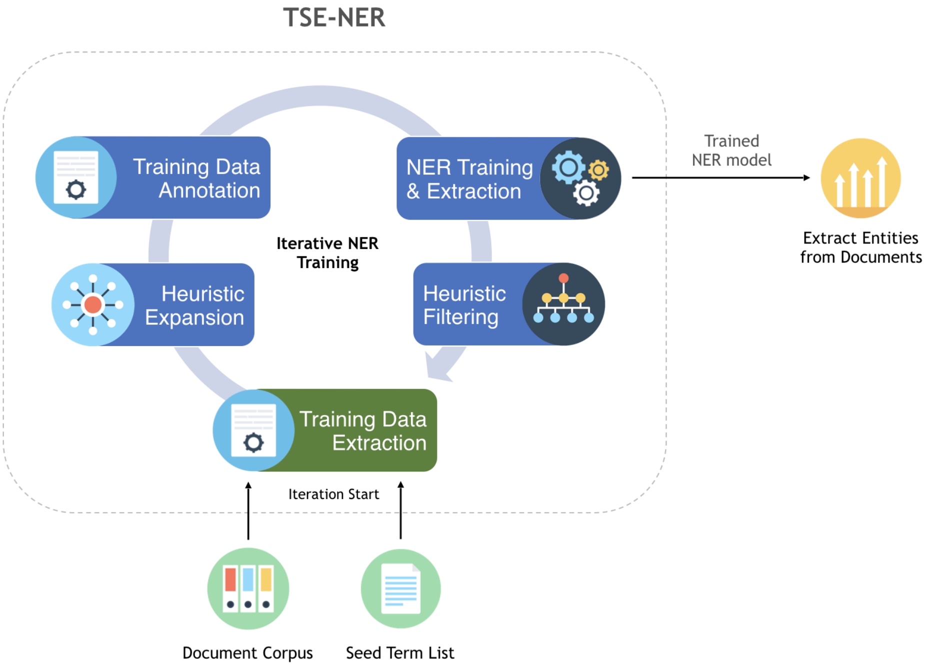


Figure 3.1: Overview of TSE-NER five-step iterative approach for training domain specific NER classifiers

The focal point of this paper is to boost the performance of the *expansion* and *filtering* steps by continuously incorporating human feedback on visualised entities expected to have high information gain.

3.1 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). The upper limit of the seed term set size exists to strike a balance between NER performance and human effort; should be small enough so one or two experts can compose the list of seed terms in less than a few hours.

As an example of this step, consider the word “LETOR” (i.e., an entity of *dataset* type) in the seed term list. All sentences 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. Surrounding sentences in the text are also extracted to better capture the usage context of the seed entity.

3.2 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 they will be labelled 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 problems the *term expansion* and *sentence expansion* strategies were introduced and described in more detail in the sections below.

3.2.1 Term Expansion

Term Expansion (TE) is designed to reduce the number of false negatives in the training sentences and provide more positive examples. This approach uses *semantic relatedness*; terms which are semantically similar (it is common for domain-specific named entities to be in close proximity, e.g. to enumerate alternative solutions, or list technical artifacts) 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. First the *word2vec* model [31] was trained on the whole corpus by learning all uni- and bi-gram word vectors of all terms in the corpus. Then, the NLTK[1] entity detection was used 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*).

[1] https://www.nltk.org

3.2.2 Sentence Expansion

The *Sentence Expansion* (SE) strategy is designed to address 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. *Doc2vec* document embeddings [19] were utilised to learn vector representations of the sentences in the corpus. For each sentence in the training data, *doc2vec* discovers the most similar sentence which does not contain any known expanded seed terms instance of the targeted type (e.g. *dataset*)

3.3 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 labelled positively. The annotated dataset can be used as an input to train any state-of-the-art supervised NER algorithm.

3.4 NER Training

For training a new *NER*, the Stanford NER tagger[2] was used to train a Conditional Random Field (CRF) model. CRF learns the hidden structure of an input sequence by defining a set of feature functions (e.g. word features, 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.

3.5 Filtering

In this final step the trained NER model was used to annotate the whole corpus and consider all the positively annotated terms as candidate terms for the next round of iteration. As noisy training data was used to train the NER, the list of entities extracted by the NER contained 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[3]; 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 Section 3.2.1. Interested readers can refer to [27] for a more detailed explanation.

As those heuristic expansion and filtering methods are rather basic in their nature and based around semantic relatedness assumptions, we discuss in Section 5 how the expansion and filtering phases can be be supported by human feedback.

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

[3] http://wordnet.princeton.edu/

The TSE-NER source code is openly available[4]. As described earlier, TSE-NER is not an original contribution of this thesis, but introduced as the baseline named entity recognition algorithm [27]. This thesis extends TSE-NER with human feedback through the Coner pipeline to boost it's performance and get visual insights of extracted entities' quality. The Coner pipeline is described in more detail in the next section.

[4] https://github.com/mvallet91/SmartPub-TSENER

4 Coner: A Collaborative Human Feedback pipeline

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 misrecognised (this can happen easily as the used training data is noisy due to the strong reliance on heuristics). While the evaluation in [27] illustrated 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. A second heuristic step of the TSE-NER training cycle is term and sentence expansion, to increase the size of the training data and thus provide more context of positively and negatively labelled named entities for the classifier. The size of the set of training sentences is limited by the amount of named entities to fetch sentences with and the quality of training data is bounded by the quality of provided seed terms.

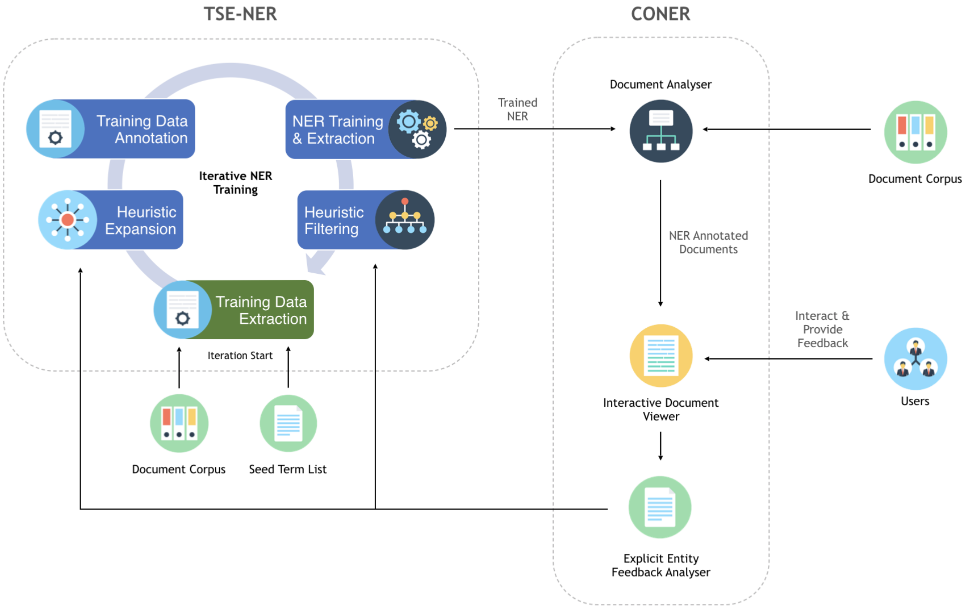


Figure 4.1: Overview of Coner Collaborative NER Pipeline: Human feedback incorporated in the TSE-NER expansion and filter phases, supporting or superseding heuristic decision making

Both shortcomings are addressed in this paper by introducing an additional layer on top of the basic TSE-NER training cycle described in Section 3. 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 setup, the following pipeline workflow (see Figure 4.1) is designed to boost the TSE-NER performance and visualise entities in the full context of their documents:

1. A NER classifier is trained for each entity type of interest (e.g. *datasets, methods and algorithms* for data science) using the TSE-NER workflow, as described in Section 3 and seen in the left part of Figure 4.1, for a manually set number of iterations or until training converges towards stable extraction performance.
2. The resulting trained NER classifier is applied to all documents in the repository to extract all entities, annotating their full texts.
3. Instead of using these extracted entities directly as input for the filtering and expansion phases of the normal TSE-NER workflow, the annotated documents are the input of the Coner pipeline, to start the collaborative human feedback process. See the right side of Figure 4.1 for an overview of the three novel Coner modules:
   (a) **Coner Document Analyser (CDA)**: This module serves two purposes; analyse documents to extract ”deep metadata” and extract entities with our smart entity selection mechanism, which only selects named entities that were recognised by multiple facet NERs and thus indicate the presence of a semantic overlap between NERs (more details in sections 4.1 and 6). When deployed in a smaller scale environment, like private lab setting, it also selects representative papers from the document corpus based on selection criteria like availability of PDF, number of times publication has been cited, distinct number of selected entities and journal it was published in. However, this is not necessary when deployed in a full scale production environment; every reader of the digital library acts as an evaluator to some degree, because Coner would be the default document viewer for the massive amount of papers in the library's collection.
   (b) **Coner Interactive Document Viewer (CIDV)**: Online interactive viewer that visualises automatically annotated entities in full context of their publication and allows users to interact with them by giving feedback on existing annotations or adding new named entities that they deem of a type of interest (e.g. *dataset* or *method*).
   (c) **Coner Feedback Analyser (CFA)**: Calculates entity type labels for each entity that received human feedback. An entity is labelled as a certain facet when the majority of evaluators (minimum number of evaluators of three or more) rated it as 'relevant' for that type.
4. The resulting entities labelled as a type by CFA are incorporated in the expansion and filter steps of the next training iteration of TSE-NER to boost performance, as described in more detail in Section 5.

More details on each of the three Coner modules can be found in the upcoming sections 4.1, 4.2 and 4.3. All source code for the Coner pipeline modules is openly available[5][6][7].

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

[6] https://github.com/vliegenthart/coner_interactive_viewer

[7] https://github.com/vliegenthart/coner_feedback_analyser

4.1 Coner Document Analyser

The core functionality of the Coner Document Analyser is to analyse documents to extract "deep metadata", select entities according to our smart entity selection mechanism and generate a named entity occurrences overview in the required format for the Coner Interactive Document Viewer to visualise the selected name entities (more details in Section 4.2).

Additionally, when deployed in a setting where the number of users of Coner is limited, like private lab experiments with relatively little human resources, CDA also tackles the imbalance between number of documents in the corpus versus the limited users' time and efforts available. In this scenario human resources are the bottleneck of our approach, so it is crucial that we attempt to maximise the knowledge gain of each user's feedback input on the relevance of entities. We have designed multiple systems with the goal to optimise efficiency of user feedback input. One of them is the design of the user experience of the viewer itself, as described in Section 4.2. Furthermore CDA aims to optimise the preprocessing of documents and deep metadata extraction:

- We designed the CDA module to analyse and filter publications from our corpus into a representative set of papers to visualise in the document viewer. We define representativeness of a paper in a domain as; the degree to which a paper conforms to the average structure, content and subject of a reviewed and published paper in that specific research domain. This paper selection process is based on a set of manually picked paper features that attempt and predict high knowledge gain in terms of entity relevance; published at a conference of interest, number of times publication has been cited, number of distinct extracted and selected entities and availability of PDF. The final candidate set of selected publications also undergoes a manual inspection of domain experts to filter out any outlying papers in terms of the defined metrics for a representative paper. More details about each step of this process can be found in the enumeration below.
- Employ a smart named entity selection mechanism that solely selects entities with high potential knowledge gain about the entity's relevance. A traditional approach to implement this is to merely select entities with a high degree of uncertainty resulting from automatic TSE-NER heuristic filtering. We considered this option, but decided to implement our alternative approach after evaluating our results from a preliminary experiment as described in sections 6 and 6.2. The Coner smart entity selection mechanism is designed to exclusively select 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.

The data flow of CDA's steps with limited availability of user resources, from now on called $cda_{limited}$, is visualised in Figure 4.2 and discussed in more detail below:

1. First, we manually create a set of conferences of interest $conf_{set}$ of size $conf_{nr}$. Conferences are picked for their widespread topical coverage in the domain of data science, big data processing and natural language processing. Our document corpus is a MongoDB database with a collection of publications owned by the Web Information Systems Research Group[8] of the faculty of EEMCS at the Delft University of Tech-

[8] http://www.wis.ewi.tudelft.nl/

nology and collected and maintained by [26, 29, 30], from now on called $corpus_A$. An ElasticSearch[9] index is created for all publications in $corpus_A$ to enable quick publication search and iteration.

2. Crawl publications from $corpus_A$ for each of $conf_{nr}$ conferences of interest. For each publication the necessary metadata and information are fetched:
   (a) `Title, conference name and full text`: Fetched from existing publications collection in $corpus_A$.
   (b) `PDF`: Check if link to PDF is present for the publication object in $corpus_A$, and if not, search the Arxiv API[10] for the PDF document.
   (c) `Number of times publication has been cited (`nr_{cited}`)`: We use this metadata entry to determine an approximation of how representative a paper is for the research domain it belongs to in terms of the type of research introduced, quality of work done and paper's impact on the domain. We used the Google Scholar API[11] to check how many times each publication has been cited.
3. Discard publications we were unable to fetch `PDF` and nr_{cited} for.
4. Extract publication's entities for each facet (*dataset* and *method*) using our NER classifiers trained analogous to the TSE-NER methodology (Section 3), with the expansion and ensemble filtering heuristics enabled. Different entity selection mechanism options are available to determine which extracted entities should be visualised in the Coner viewer and receive human feedback. We choose to solely select entities that have been doubly classified (assigned to both facets) by the NERs. More details about entity selection mechanisms can be found in the evaluation Section 6.
5. Filter out publications with less than the manually picked threshold $min\_entities_{nr}$ total distinct selected entities, to ensure viewer paper candidates contain at least minimal usable amount of entities.
6. Now we have the required metadata about each publication (title, conference name, full text, PDF, number of times cited and named entity lists), we can sort papers (sorting scoped for each conference in $conf_{set}$) by number of times paper has been cited (*primary, descending*) and on number of distinct entities (*secondary, descending*). Papers are ordered for each conference separately, because we want to ensure there is no bias towards any conference that might be more popular in terms of average number of citations per publication, as we want to keep a wide range of topics in our selected papers.
7. Pick each conference's top papers according to manually set top_papers_{nr} (10 times the maximum number of papers to receive feedback in the viewer, which is set manually dependent on the amount of evaluators).
8. Apply analysis on each of top_papers_{nr} publications for each conference using the PDFNLT[12] tools for natural language text aware PDF structure analysis. PDFNLT was built by the Aizawa Natural Language Processing Lab at the National Institute of Informatics[13] and enables the extraction of PDF structure and content information with the workflow described in [1, 15]. PDFNLT allows us to analyse publications' PDFs and generate the full text, a XHTML file, a list of full text sentences (with sentence metadata like sentence identifier, and word identifiers for each word in a sentence, that matches with word element attributes in the XHTML), a list of math formulas and references.

[9] https://www.elastic.co/
[10] https://github.com/lukasschwab/arxiv.py
[11] https://github.com/ckreibich/scholar.py
[12] https://github.com/KMCS-NII/PDFNLT-1.0
[13] http://www-al.nii.ac.jp

9. Match NER extracted and smart selected entities for each facet with text occurrences in the list of sentences generated by PDFNLT for each publication to generate an array of entity occurrences with all metadata required for each entity's visualisation in the Coner Interactive Document Viewer (CIDV); entity word bounding boxes, entity text, page number, timestamp of generation and paper identifier.
10. Sort papers again on number of entity occurrences found by PDFNLT, to account for PDFNLT entity matching error rate of about 10% (so 10% of extracted entities cannot be matched in the sentences extracted by PDFNLT) and take top $candidate\_papers_{nr}$ papers from each of $conf_{nr}$ conferences. Results in a candidate viewer paper set of size:

$$viewer\_papers_{nr} = candidate\_papers_{nr} \times conf_{nr} \tag{1}$$

11. Inspect $viewer\_papers_{nr}$ papers manually and pick representative papers for the domain (according to previously described selection criteria) with a balanced set of entities for both facets.
12. Generate highlights file, i.e., list of highlights with format shown in Figure 4.4, for the CIDV to load and visualise entity occurrences for final selection of viewer papers.

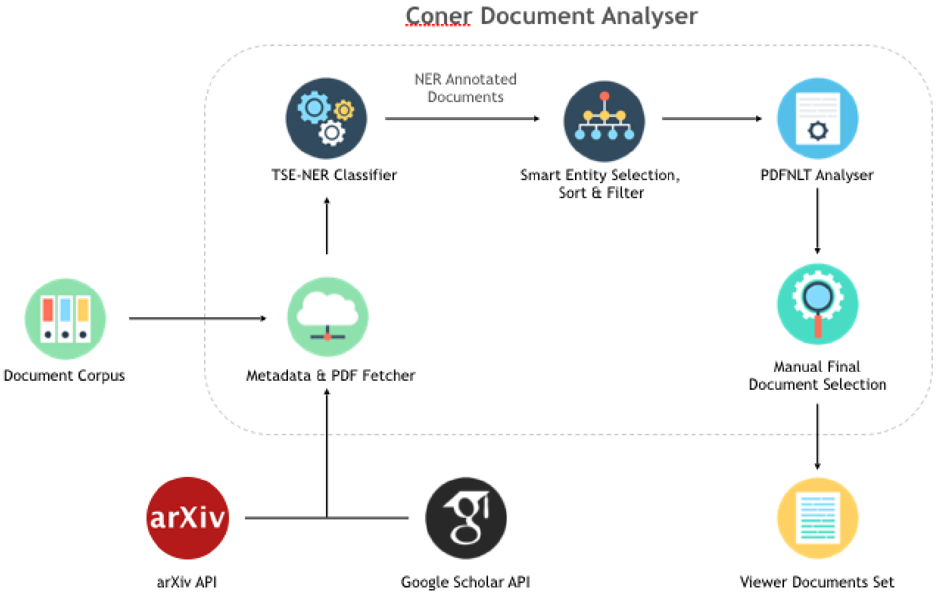


Figure 4.2: Corpus documents pass through Coner Document Analyser (CDA) components (limited users setup) to fetch metadata & PDFs, extract entities, smart select entities, sort & filter, analyse PDFs & manually inspect the final documents selection to be visualised in the viewer

When Coner would be deployed in a full production system, like a full scale digital library, only five slightly modified versions of the described $cda_{limited}$ steps are required, from now on called $cda_{production}$. The starting document corpus is the entire paper collection or a subset of the paper collection describing a specific domain of the selected digital library, from now on called $corpus_{lib}$. The steps of $cda_{production}$ are described below and visualised in Figure 4.3:

1. Fetch metadata not present in digital library documents corpus. A digital library's document collection will most likely include most required metadata for Coner, as authors at least submit their paper's title, conference name and PDF. Worst case CDA will have to handle fetching the number of a times a paper has been cited analogous to *step 2c* of $cda_{limited}$.
2. Extract and select publication's entities for each facet (*dataset* and *method*) analogous to *step 4* of $cda_{limited}$.
3. Analyse each paper with PDFNLT analogous to *step 8* of $cda_{limited}$, with top_papers_{nr} set to the size of set $corpus_{lib}$.
4. Match NER extracted and selected entities with paper's extracted full text analogous to *step 9* of $cda_{limited}$.
5. Generate highlights file for CIDV analogous to step 12 of $cda_{limited}$.

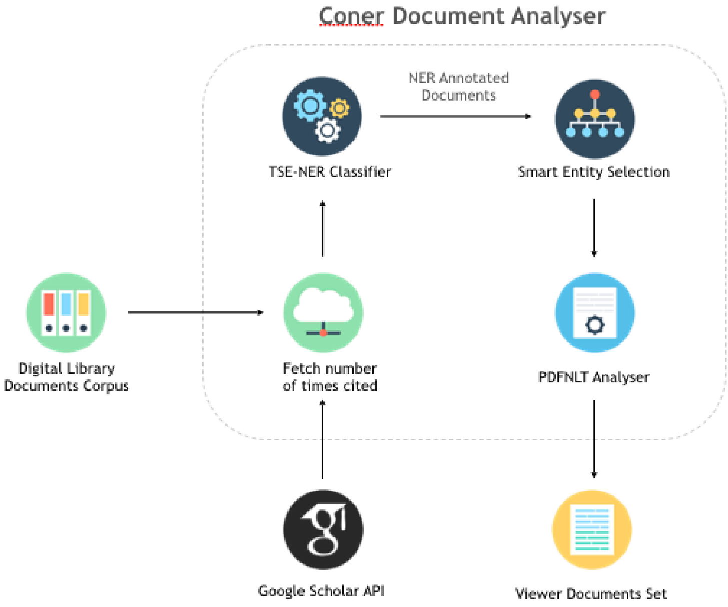
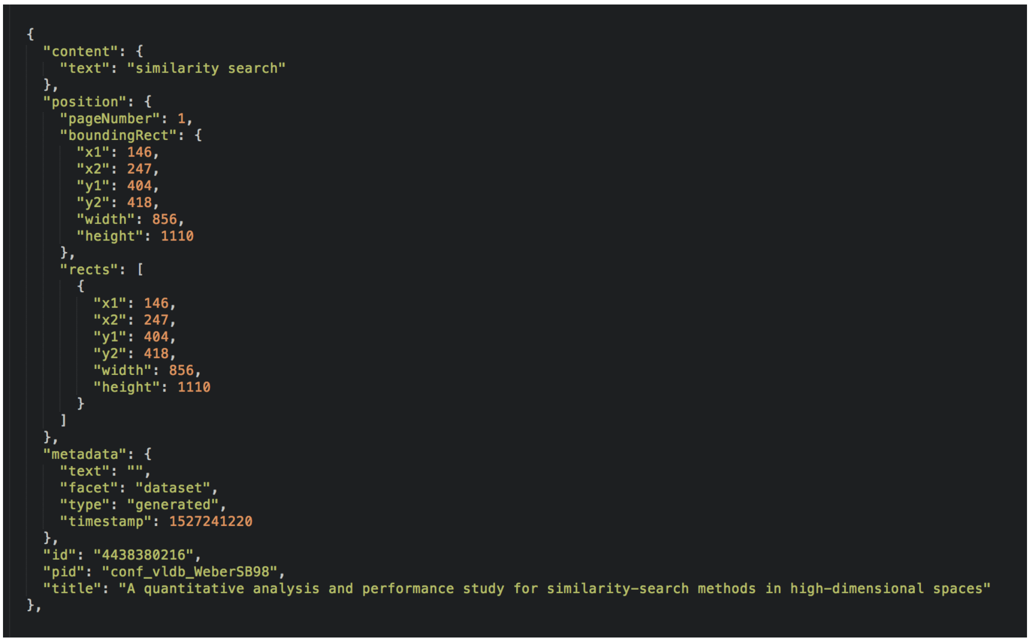


Figure 4.3: Corpus documents pass through Coner Document Analyser (CDA) components (full production setup) to fetch number of times cited, extract entities, smart select entities, analyse PDFs and generate highlights for viewer



```
{
  "content": {
    "text": "similarity search"
  },
  "position": {
    "pageNumber": 1,
    "boundingRect": {
      "x1": 146,
      "x2": 247,
      "y1": 404,
      "y2": 418,
      "width": 856,
      "height": 1110
    },
    "rects": [
      {
        "x1": 146,
        "x2": 247,
        "y1": 404,
        "y2": 418,
        "width": 856,
        "height": 1110
      }
    ]
  },
  "metadata": {
    "text": "",
    "facet": "dataset",
    "type": "generated",
    "timestamp": 1527241220
  },
  "id": "4438380216",
  "pid": "conf_vldb_WeberSB98",
  "title": "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces"
},
```

Figure 4.4: Entity highlight format for CIDV

4.2 Coner Interactive Document Viewer

As a second core module of Coner, we introduce an *online interactive document viewer*. 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 recognised entities with respect to their relevance of their identified type. Additionally, users can select words or a sequence of words directly in the viewer to annotate them as named entities they deem relevant as one of the entity types (*dataset* or *method*). A screenshot of the current version of the document viewer *V1.0* is displayed in Figure 4.8. This section describes the agile design process, features and technology stack of the document viewer.

4.2.1 Design Process & Features

The Coner Interactive Document Viewer (CIDV) was inspired by the document viewer included in the PDFNLT[14] toolkit. This PDFNLT document viewer (Figure 4.5) has a layout of two vertical columns; the left column displays the raw XHTML file and the right column shows the original publication in image format. Words and paragraphs are highlighted in the right column when the mouse is hovered over the corresponding word in the left XHTML file column.

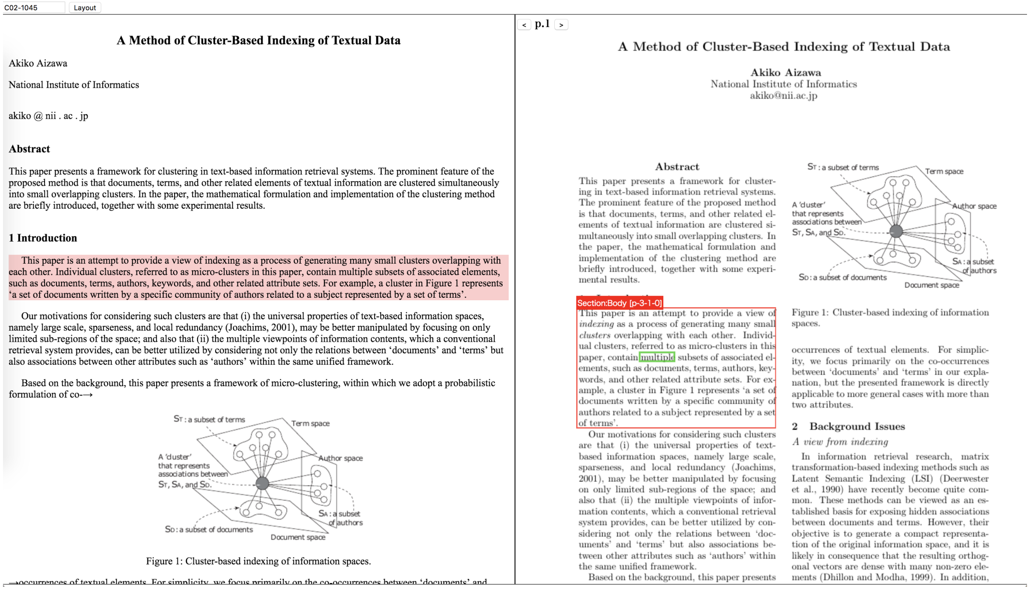
[14] https://github.com/KMCS-NII/PDFNLT-1.0

Coner Interactive Document Viewer POC

We built our first Proof Of Concept (POC) of the Coner viewer by extending the features of the original PDFNLT viewer in two steps:

- Add feature that shows highlights of entities, loaded from a JSON file consisting of an array of entities' document bounding boxes, on top of the document image (Figure 4.6).
- Enhance entity highlighting with dynamic highlight colors for each entity facet and add user feedback popups when clicking on a highlighted entity (Figure 4.7).

One of our design goals for the interactive viewer POC module 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 recognised 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 for the POC limited us to only elicit feedback on entities which have been detected by the NER (i.e. feedback for filtering), but did not allows us to find entities which have been missed completely (i.e., feedback for expansion). We address this issue, along with additional features and user interaction & interface improvements in our next version of the CIDV described in the next section.



A Method of Cluster-Based Indexing of Textual Data

Akiko Aizawa

National Institute of Informatics

akiko @ nii . ac . jp

Abstract

This paper presents a framework for clustering in text-based information retrieval systems. The prominent feature of the proposed method is that documents, terms, and other related elements of textual information are clustered simultaneously into small overlapping clusters. In the paper, the mathematical formulation and implementation of the clustering method are briefly introduced, together with some experimental results.

1 Introduction

This paper is an attempt to provide a view of indexing as a process of generating many small clusters overlapping with each other. Individual clusters, referred to as micro-clusters in this paper, contain multiple subsets of associated elements, such as documents, terms, authors, keywords, and other related attribute sets. For example, a cluster in Figure 1 represents 'a set of documents written by a specific community of authors related to a subject represented by a set of terms'.

Our motivations for considering such clusters are that (i) the universal properties of text-based information spaces, namely large scale, sparseness, and local redundancy (Joachims, 2001), may be better manipulated by focusing on only limited sub-regions of the space; and also that (ii) the multiple viewpoints of information contents, which a conventional retrieval system provides, can be better utilized by considering not only the relations between 'documents' and 'terms' but also associations between other attributes such as 'authors' within the same unified framework.

Based on the background, this paper presents a framework of micro-clustering, within which we adopt a probabilistic formulation of co-→

S_T : a subset of terms
Term space
A 'cluster' that represents associations between S_T, S_A, and S_D.
Author space
S_A : a subset of authors
S_D : a subset of documents
Document space

Figure 1: Cluster-based indexing of information spaces.

→occurrences of textual elements. For simplicity, we focus primarily on the co-occurrences between 'documents' and

A Method of Cluster-Based Indexing of Textual Data

Akiko Aizawa
National Institute of Informatics
akiko@nii.ac.jp

Abstract

This paper presents a framework for clustering in text-based information retrieval systems. The prominent feature of the proposed method is that documents, terms, and other related elements of textual information are clustered simultaneously into small overlapping clusters. In the paper, the mathematical formulation and implementation of the clustering method are briefly introduced, together with some experimental results.

This paper is an attempt to provide a view of *indexing* as a process of generating many small *clusters* overlapping with each other. Individual clusters, referred to as micro-clusters in this paper, contain multiple subsets of associated elements, such as documents, terms, authors, keywords, and other related attribute sets. For example, a cluster in Figure 1 represents 'a set of documents written by a specific community of authors related to a subject represented by a set of terms'.

Our motivations for considering such clusters are that (i) the universal properties of text-based information spaces, namely large scale, sparseness, and local redundancy (Joachims, 2001), may be better manipulated by focusing on only limited sub-regions of the space; and also that (ii) the multiple viewpoints of information contents, which a conventional retrieval system provides, can be better utilized by considering not only the relations between 'documents' and 'terms' but also associations between other attributes such as 'authors' within the same unified framework.

Based on the background, this paper presents

S_T : a subset of terms
Term space
A 'cluster' that represents associations between S_T, S_A, and S_D.
Author space
S_A : a subset of authors
S_D : a subset of documents
Document space

Figure 1: Cluster-based indexing of information spaces.

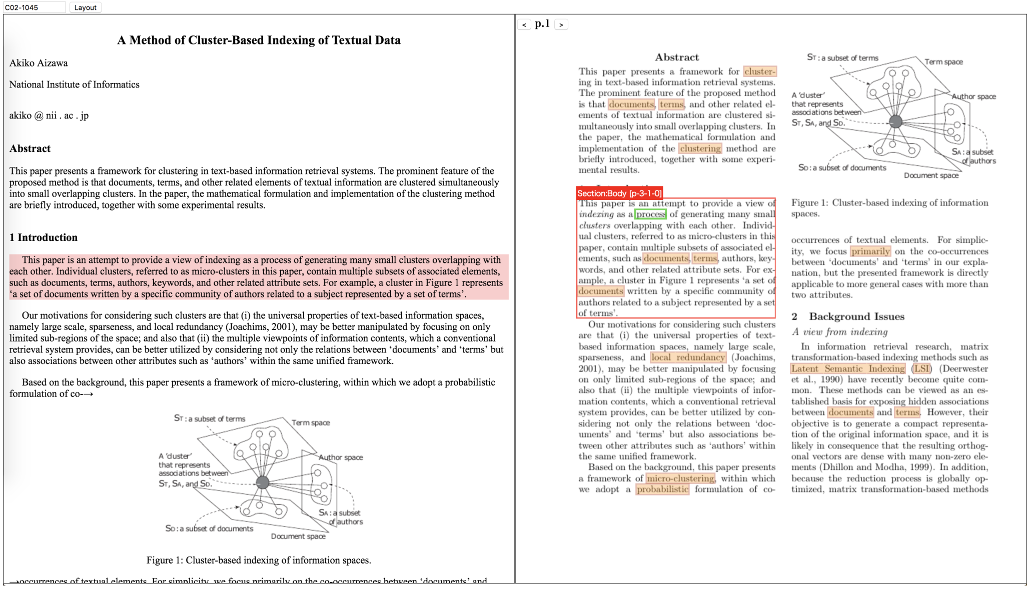
occurrences of textual elements. For simplicity, we focus primarily on the co-occurrences between 'documents' and 'terms' in our explanation, but the presented framework is directly applicable to more general cases with more than two attributes.

2 Background Issues

A view from indexing

In information retrieval research, matrix transformation-based indexing methods such as Latent Semantic Indexing (LSI) (Deerwester et al., 1990) have recently become quite common. These methods can be viewed as an established basis for exposing hidden associations between documents and terms. However, their objective is to generate a compact representation of the original information space, and it is likely in consequence that the resulting orthogonal vectors are dense with many non-zero elements (Dhillon and Modha, 1999). In addition,

Figure 4.5: PDFNLT Document Viewer



A Method of Cluster-Based Indexing of Textual Data

Akiko Aizawa

National Institute of Informatics

akiko @ nii . ac . jp

Abstract

This paper presents a framework for clustering in text-based information retrieval systems. The prominent feature of the proposed method is that documents, terms, and other related elements of textual information are clustered simultaneously into small overlapping clusters. In the paper, the mathematical formulation and implementation of the clustering method are briefly introduced, together with some experimental results.

1 Introduction

This paper is an attempt to provide a view of indexing as a process of generating many small clusters overlapping with each other. Individual clusters, referred to as micro-clusters in this paper, contain multiple subsets of associated elements, such as documents, terms, authors, keywords, and other related attribute sets. For example, a cluster in Figure 1 represents 'a set of documents written by a specific community of authors related to a subject represented by a set of terms'.

Our motivations for considering such clusters are that (i) the universal properties of text-based information spaces, namely large scale, sparseness, and local redundancy (Joachims, 2001), may be better manipulated by focusing on only limited sub-regions of the space; and also that (ii) the multiple viewpoints of information contents, which a conventional retrieval system provides, can be better utilized by considering not only the relations between 'documents' and 'terms' but also associations between other attributes such as 'authors' within the same unified framework.

Based on the background, this paper presents a framework of micro-clustering, within which we adopt a probabilistic formulation of co-→

ST : a subset of terms
Term space
A 'cluster' that represents associations between ST, SA, and SD.
Author space
SA : a subset of authors
SD : a subset of documents
Document space

Figure 1: Cluster-based indexing of information spaces.

→occurrences of textual elements. For simplicity, we focus primarily on the co-occurrences between 'documents' and

Abstract

This paper presents a framework for clustering in text-based information retrieval systems. The prominent feature of the proposed method is that documents, terms, and other related elements of textual information are clustered simultaneously into small overlapping clusters. In the paper, the mathematical formulation and implementation of the clustering method are briefly introduced, together with some experimental results.

Section:Body [p-3-1-0]

This paper is an attempt to provide a view of *indexing* as a process of generating many small *clusters* overlapping with each other. Individual clusters, referred to as micro-clusters in this paper, contain multiple subsets of associated elements, such as documents, terms, authors, keywords, and other related attribute sets. For example, a cluster in Figure 1 represents 'a set of documents written by a specific community of authors related to a subject represented by a set of terms'.

Our motivations for considering such clusters are that (i) the universal properties of text-based information spaces, namely large scale, sparseness, and local redundancy (Joachims, 2001), may be better manipulated by focusing on only limited sub-regions of the space; and also that (ii) the multiple viewpoints of information contents, which a conventional retrieval system provides, can be better utilized by considering not only the relations between 'documents' and 'terms' but also associations between other attributes such as 'authors' within the same unified framework.

Based on the background, this paper presents a framework of micro-clustering, within which we adopt a probabilistic formulation of co-

Figure 1: Cluster-based indexing of information spaces.

occurrences of textual elements. For simplicity, we focus primarily on the co-occurrences between 'documents' and 'terms' in our explanation, but the presented framework is directly applicable to more general cases with more than two attributes.

2 Background Issues

A view from indexing

In information retrieval research, matrix transformation-based indexing methods such as Latent Semantic Indexing (LSI) (Deerwester et al., 1990) have recently become quite common. These methods can be viewed as an established basis for exposing hidden associations between documents and terms. However, their objective is to generate a compact representation of the original information space, and it is likely in consequence that the resulting orthogonal vectors are dense with many non-zero elements (Dhillon and Modha, 1999). In addition, because the reduction process is globally optimized, matrix transformation-based methods

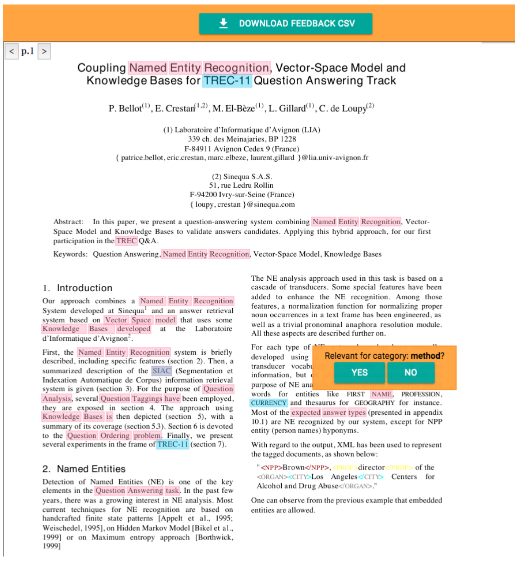
Figure 4.6: PDFNLT Document Viewer with named entity highlights

Coner Interactive Document Viewer V1.0

During our POC user testing phase, where we recruited 10 lab student of graduate or post-graduate level to stress test and give feedback on the POC viewer, we received valuable feedback on the usability of the viewer. We integrated this feedback alongside our own design goals and feature ideas into the following list of functional and nonfunctional requirements, each successfully implemented, for our first full version of the CIDV:

- The POC viewer only allows users to provide feedback on the entity's facet it was recognised as. We extended this by adding YES/NO relevance feedback buttons for both facets per entity.
- Ability for user to add new typed entity simply by selecting text, with their mouse, they encounter in the document (Figure 4.9a).
- All users can give feedback on other users' added entities as extra quality control in a similar fashion like they would give feedback on automatically highlighted entities (Figure 4.9b). User added entities have their distinct highlight colour, yellow, however other users can not see as which type the entity was added by the original contributor, to remove bias towards the original type resulting from users' trust in the contributor's expertise instead of their own.
- When rating an entity as YES for a facet or adding a new typed entity, the other facet is automatically rated as NO, as it is extremely rare for an entity to be of multiple types. If the rare occasion does occur where an entity is of multiple types, the evaluator is free change the automatic NO rating to YES.
- Sidebar with clear overview of entities in current paper and tags for feedback of user on each of entity's facets (Figure 4.8).
- Built viewer from scratch using a lightweight state of the art web technology stack (more details in Section 4.2.2) to emphasise the scalability, extensibility and maintainability of our open source[15] web viewer.

[15] https://github.com/vliegenthart/coner_interactive_viewer



DOWNLOAD FEEDBACK CSV

< p.1 >

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.

Relevant for category: **method**?

YES NO

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 4.7: POC Coner Interactive Document Viewer with relevance feedback buttons for entities classified as *dataset* (blue highlights), *method* (pink highlights) or both (purple highlight)

– Full user authentication and authorisation system.
– Persistent high speed storage and access for all user activity and entity feedback instances.

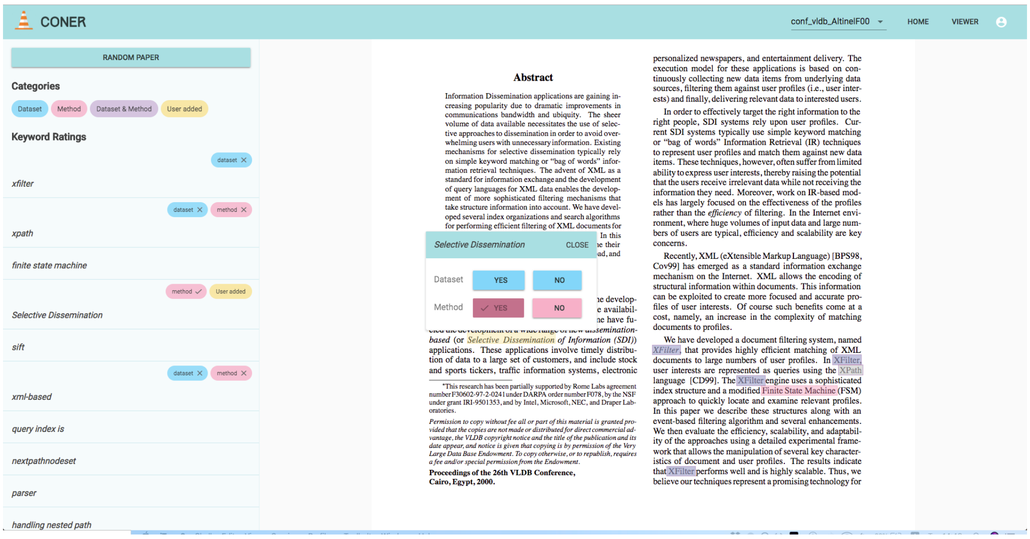
As an additional feature, the full history of user feedback inputs for each entity is stored. This gives us the ability to analyse how often users change their mind about an entity's type relevance.

4.2.2 Technology Stack

The CIDV *V1.0* is built with a modern open source component-based JavaScript full-stack. We used Node.js[16] as our server-side JavaScript framework, NPM[17] to manage Node.js dependencies and packages and React.js[18] to built our client-side JavaScript web components. Furthermore we used the Webpack[19] static module bundler for modern JavaScript applications. We choice this setup for it's extensive community of open source developers, massive ecosystem of NPM packages for Node.js and React.js and overall customisation, scalability and lightweight properties. The Coner viewer's React.js web components are reusable, customisable, asynchronously loaded and have internal data management to enforce the modular isolated design. Any contributors proficient in this extremely common JavaScript full-stack setup will be able to easily customise the viewer's functionality.

We settled on a Firebase[20] storage back-end. Firebase offers many benefits over other no-SQL solutions for our relatively small scale data production and management; full user

[16] https://nodejs.org/en/
[17] https://www.npmjs.com/
[18] https://reactjs.org/docs/getting-started.html
[19] https://webpack.js.org/concepts/
[20] https://firebase.google.com/docs/



Abstract

Information Dissemination applications are gaining increasing popularity due to dramatic improvements in communications bandwidth and ubiquity. The sheer volume of data available necessitates the use of selective approaches to dissemination in order to avoid overwhelming users with unnecessary information. Existing mechanisms for selective dissemination typically rely on simple keyword matching or "bag of words" information retrieval techniques. The advent of XML as a standard for information exchange and the development of query languages for XML data enables the development of more sophisticated filtering mechanisms that take structure information into account. We have developed several index organizations and search algorithms for performing efficient filtering of XML documents for
In this
ne their
ad, and

ne develop-
e availabil-
ne have fu-
eled the development of a wide range of new *dissemination-based* (or *Selective Dissemination of Information (SDI)*) applications. These applications involve timely distribution of data to a large set of customers, and include stock and sports tickers, traffic information systems, electronic personalized newspapers, and entertainment delivery. The execution model for these applications is based on continuously collecting new data items from underlying data sources, filtering them against user profiles (i.e., user interests) and finally, delivering relevant data to interested users.

In order to effectively target the right information to the right people, SDI systems rely upon user profiles. Current SDI systems typically use simple keyword matching or "bag of words" Information Retrieval (IR) techniques to represent user profiles and match them against new data items. These techniques, however, often suffer from limited ability to express user interests, thereby raising the potential that the users receive irrelevant data while not receiving the information they need. Moreover, work on IR-based models has largely focused on the effectiveness of the profiles rather than the *efficiency* of filtering. In the Internet environment, where huge volumes of input data and large numbers of users are typical, efficiency and scalability are key concerns.

Recently, XML (eXtensible Markup Language) [BPS98, Cov99] has emerged as a standard information exchange mechanism on the Internet. XML allows the encoding of structural information within documents. This information can be exploited to create more focused and accurate profiles of user interests. Of course such benefits come at a cost, namely, an increase in the complexity of matching documents to profiles.

We have developed a document filtering system, named *XFilter*, that provides highly efficient matching of XML documents to large numbers of user profiles. In XFilter, user interests are represented as queries using the XPath language [CD99]. The XFilter engine uses a sophisticated index structure and a modified Finite State Machine (FSM) approach to quickly locate and examine relevant profiles. In this paper we describe these structures along with an event-based filtering algorithm and several enhancements. We then evaluate the efficiency, scalability, and adaptability of the approaches using a detailed experimental framework that allows the manipulation of several key characteristics of document and user profiles. The results indicate that XFilter performs well and is highly scalable. Thus, we believe our techniques represent a promising technology for

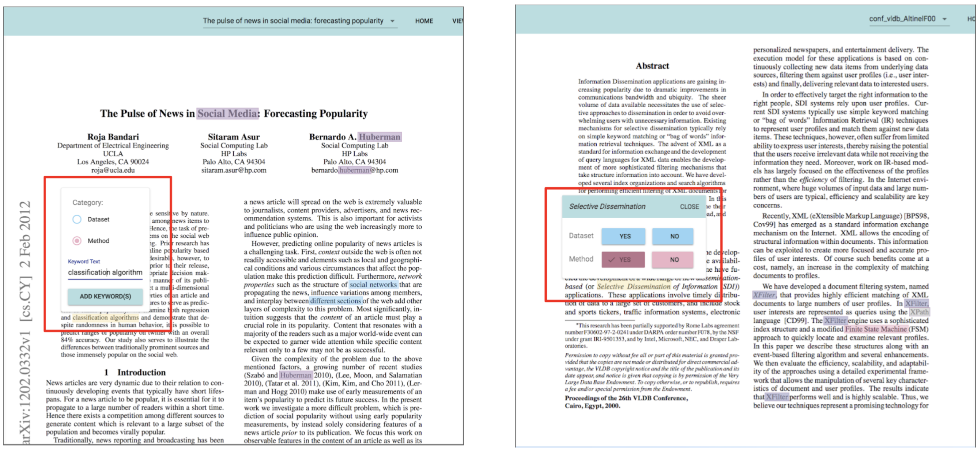
*This research has been partially supported by Rome Labs agreement number F30602-97-2-0241 under DARPA order number F078, by the NSF under grant IRI-9501353, and by Intel, Microsoft, NEC, and Draper Laboratories.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, requires a fee and/or special permission from the Endowment.

Proceedings of the 26th VLDB Conference,
Cairo, Egypt, 2000.

Figure 4.8: *V1.0* Coner Interactive Document Viewer with feedback buttons for entities highlighted with different colors for facets; light blue for *dataset*, pink for *method*, purple for entities classified for both *dataset* and *method* facets and yellow for newly user added entities

authentication integration with Node.js and React.js, real-time database management online console, offline data persistence and automatic database backups. We wrote our own Firebase API endpoint wrapper methods in JavaScript, which are called and triggered from various user interactions with React.js client-side components. This allows for a flexible back-end setup where Firebase is effortlessly exchangeable for any other no-SQL document or key-value based storage (e.g. MongoDB, Couchbase, Redis, etc.) to allow for more fine grained control and scalability for the future.



(a) User adds *method* entity

(b) User added entity feedback popup

Figure 4.9: Coner Viewer features: User can add new entity and give feedback

4.2.3 Coner Viewer Competitors

We performed an analysis on the main strengths and weakness of other online document annotation tools. The main competitors for the Coner Viewer are:

- Prodigy[21]: Online annotation tool that uses it's self proclaimed "continuous active learning" technique to only ask users to annotate examples that the machine learning classifier cannot identify itself. The major weakness with this solution, is that it only visualises annotated entities in a single sentence on the screen, not the full context of the paper, thus not using human's ability to detect context to it's full ability.
- Tagtog[22]: Semi-supervised text annotation tool to train machine learning models with an extensive suite of features, but incredibly pricey if you want to receive feedback from more than one evaluator. Also, it is not open sourced, so contributors are unable to extend and customise features.
- Anafora[23]: Doesn't actually show full PDF with annotated entities, just the raw text, and the user interface is not intuitive.
- Annotate[24]: Expensive for more than three contributors, not open sourced and doesn't focus on annotation scientific documents.

Our comparison of competitors highlighted that none of them possessed the combination of three properties we see as essential for an adequate web annotation tool for scientific documents:

- Visualise annotated entities in full context of the scientific publication's PDF.
- Allow users to interact with annotated entities in full context of the scientific publication's PDF.
- Code is open sourced, free and customisable, to take full benefit of our research community's creativity and software development abilities.

[21] https://prodi.gy/
[22] https://www.tagtog.net/
[23] https://github.com/weitechen/anafora
[24] https://www.annotate.co

The Coner Interactive Document Viewer is specifically designed and built to adhere to these three requirements and that is why we believe that it is the ultimate web annotation tool for scientific documents.

4.3 Coner Feedback Analyser

During the usage of the Coner system, we collect all explicit user feedback on the type correctness of the detected and user added entities, from now on called $entities_{fb}$. The complete set $entities_{fb}$ are preprocessed to remove erroneous symbols and trailing whitespaces and converted to lowercase. The Entity Facet Relevance Score (EFRS) is defined to be the ratio of evaluators that rated an entity as relevant for a facet ($votes_{relevant}$) divided by the total number of evaluators that rated the entity for the same facet ($votes_{total}$). An entity is labelled as *relevant for facet* by majority vote, thus when more than half of the entity's evaluators rated the entity as relevant for a facet:

$$EFRS_{entity} = votes_{relevant} \div votes_{total} > 0.5 \tag{2}$$

The following restrictions are enforced for the entity facet relevance scoring process through majority vote:

- Only calculate relevance score if three or more evaluators gave facet relevance feedback for that entity.
- Only consider an evaluator for a paper if he/she gave feedback on more than 10 entities in that paper.

These EFRSs are then used to manually or periodically (e.g., every few hours, nightly or weekly) continue the training of the NER algorithm by executing more training iterations until convergence is reached (see Figure 4.1). Section 5 describes in more detail how the EFRSs boost the heuristic expansion and filtering steps of TSE-NER.

5 TSE-NER heuristics boosting with Coner

This section describes the synergy between the TSE-NER and the Coner pipeline; boosting the performance of TSE-NER heuristics by integrating aggregated human feedback on a set of typed named entities.

5.1 Term & Sentence Extraction Coner Boost (TSECB)

As described in Section 3.2, TSE-NER expands the set of seed terms and set of training sentences every iteration, to balance and increase the size of the training data. Coner boosts this process by adding relevant Coner user added typed named entities with relevance scores (EFRS) higher than 0.5 for that facet to the seed term set for the next iteration. Any typed named entities with a relevance score (EFRS) equal to or lower than 0.5 are ignored for the expansion boosting step.

5.2 Filtering Coner Boost (FCB)

The TSE-NER heuristic filtering step, as described in Section 3.5, relies on a setup of filtering techniques that can be employed individually or in ensemble to filter out noisy data and improve precision. The TSE-NER ensemble filtering resulted in the highest combined facet F-score in TSE-NER's performance evaluation [27]. We introduce an ensemble filter boosting mechanism that relies on the assumption that named entities rated as relevant for a facet by the majority of evaluators is regarded as a true positive for that facet and should not be filtered out. With this assumption, Coner overrules the ensemble filtering decision by always keeping a typed named entity with a relevance score (EFRS) higher than 0.5 and always removing a typed named entity with relevance score (EFRS) lower than 0.5. So manual human feedback supersedes all heuristic decisions if an EFRS score is present. However, if an entity's relevance score is not present, or exactly 0.5 or could not be calculated according to the restrictions of the CFA, then the stand-alone ensemble filtering result is adhered to. We will provide more detailed information on how human feedback conflicts with heuristic decisions in our evaluation in Section 6.

6 Evaluation

In the previous sections we proposed the Coner pipeline, an extension onto the existing TSE-NER algorithm, with the goal to incorporate human feedback in the iterative NER classifier training process. To evaluate the effectiveness of our approach, we focus mainly on answering the following three research questions in this evaluation:

Research Question 1 (RQ1):
What is the nature of the elicited human feedback?

– Hypothesis: Human feedback elicited in the Coner Interactive Document Viewer is time efficient and results in a sufficiently high inter-annotator agreement to calculate EFRSs for a majority named entities.
– Motivation: The Coner viewer is built to minimise the cognitive load of inputting feedback by evaluators, thus aimed to reduce time spend giving feedback on each named entity. We measured time efficiency in average time spend per entity feedback instance. Furthermore each each named entity occurrence is highlighted in it's full context of a paper, to help align evaluators' understanding of the contextual semantic meaning of that particular named entity. We measured inter-annotator agreement in Cohen's Kappa between pairs of evaluators that rated a joint subset of entities.

Research Question 2 (RQ2):
In how far does human feedback confirm or conflict with TSE-NER heuristics?

– Hypothesis: Humans are more selective compared to TSE-NER heuristic filtering in recognising true positives to keep in the filtering process and are superior in identifying false positives to discard.
– Motivation: The nature of the expanded training data and extracted entities from the corpus is noisy and the semantic meanings of a multitude of named entities are ambiguous due to the difference in contextual occurrence in papers. As opposed to algorithmic heuristic decision making, a human's conscious reasoning is better capable of recognising semantics of specific named entity occurrence situations. It is crucial that this preliminary research question is answered before we can do any type of performance comparison. We hypothesise that if no conflict between human feedback and TSE-NER heuristics is discovered whatsoever, it is unlikely that the overall TSE-NER will be positively boosted by Coner.

Research Question 3 (RQ3):
How does incorporating human feedback on entities into the TSE-NER expansion and filtering steps improve the overall performance with respect to precision, recall, and F-measures?

– Hypothesis: Coner boosts the performance measures both in terms of precision and recall compared to TSE-NER.
– Motivation: The main goal of Coner is to boost TSE-NER's performance by utilising humans' superior ability to judge relevance of entities in their full context compared to machines, thus we hypothesise that Coner will indeed improve the performance measures and help improve faceted keyword search in digital libraries.

To answer these research questions, and to perceive if our hypothesises are correct, we conducted two user experiments, both revolving around receiving human feedback on papers' extracted named entities and allowing users to add new named entities that they deem relevant for the *dataset* or *method* facets during the manual annotation phase. However each experiment was conducted with different entity extraction filtering techniques and instructions giving to evaluators.

The first experiment, as described in Section 6.2, focuses on answering RQ1 and RQ2 by analysing human feedback on unfiltered extracted entities with evaluator instructions to focus on adding new relevant entities in addition to giving feedback on automatically extracted entities. This setup allows us to clearly compare different filtering methods, because we have access to the unfiltered extracted entities, human filtered entities and TSE-NER heristics filtered entities.

The second experiment in Section 6.3 tackles RQ3 and employs a smart entity selection technique to only select entities for visualisation that were recognised by both facet NERs, from now on called *doubly classified*. We choose this selection mechanism above traditional uncertainty sampling, where we would select entities that could not be clearly filtered out or kept by ensemble heuristic filtering; a conflict between different filters can result in a 50/50 majority vote between filters, thus rendering the verdict for that entity useless. We motivate this decision by looking at our biggest bottleneck and limitation; human feedback is expensive in terms of time and incentivisation. In order to make human feedback scalable, we aspire to maximise the potential knowledge gain of each user feedback instance on a named entity. A doubly classified named entity almost certainly indicates a fault in the trained models, because in reality an entity is rarely part of two facets, i.e., an entity doesn't describe a *method* and a *dataset*, but in the best case scenario belongs to one of the facets. Section 6.2.5 further discusses our smart entity selection mechanism based on tangible results of our first experiment.

6.1 Training Data

For evaluating our approach, we relied on a corpus similar to the one collected and maintained in [26, 29, 30] and already used in [27], because it covers a wide range of interesting subjects within the data science and natural language processing domains. To create our corpus we selected 11,598 papers from 11 conferences: 1613 papers from the Joint Conference on Digital Libraries (JCDL), 280 papers from the International Conference on Theory and Practice of Digital Libraries (TPDL), 230 papers from the Text Retrieval Conference (TREC), 828 papers from the European Conference on Research and Advanced Technology in Digital Libraries (ECDL), 634 papers from the Extended Semantic Web Conference (ESWC), 827 papers from the International Conference on Web and Social Media (ICWSM), 444 papers from the International Conference on Very Large Databases (VLDB), 442 papers from the Annual Meetings of the Association for Computational Linguistics (ACL), 2100 papers from the International World Wide Web Conference (The Web Conference), 2100 papers from the the International Conference on Software Engineering (ICSE) and 2100 papers from the International Conference on Research and Development in Information Retrieval (SIGIR).

The maximum number of papers included in the corpus from a single conference was limited to 2100, to enforce a degree of balance in the spread of papers across all 11 conferences and to build a corpus of the same size as used to evaluate the baseline

TSE-NER algorithm [27]. As described in Section 4.1, we selected a subset of papers from the MongoDB database collection of publications owned by the Web Information Systems Research Group[25] of the faculty of EEMCS at the Delft University of Technology and collected and maintained by [26, 29, 30]. An ElasticSearch[26] index is created for all publications in the collection to enable quick publication search and iteration. Metadata extraction & analysis, PDF fetching and PDF analysis has been performed by our Coner Document Analyser module with the $cda_{limited}$ setup enabled (more details in Section 4.1).

6.2 Experiment 1: Human Feedback on Unfiltered Entities

The goal of this experiment is to give a deeper insight in the nature of human feedback through our crowdsourcing solution and how human judgement of entity relevance confirms or conflicts with automatic heuristic filtering. Our experiment's analysis consists of three parts; assessment of human feedback, a qualitative inspection of most and least relevant entities according to human feedback and a comparison between stand-alone TSE-NER heuristics and TSE-NER heuristics with incorporated Coner human feedback. We discuss our interpretation of the acquired results in Section 6.2.5.

First, we trained the TSE-NER algorithm on the training data analogously to [27], with the algorithm setup as following:

- Train models for two facets: *Dataset* and *Method*.
- Seed term set size of 50 chosen manually to strike balance between increased classifier performance in terms of F-scores with more seeds, as apparent from evaluation in [27] and less human effort with less seeds. Seed term set of this size could be created by a single or a few domain experts in a few hours or less and thus does not immobilise the lightweight design behind TSE-NER.
- Run one iteration of the TSE-NER training process. We did not train the TSE-NER for multiple iterations until convergence, because we want to measure the potential divergence in performance between TSE-NER heuristics and Coner in the iterations to follow.

We used the trained NERs for each facet as input for our collaborative Coner pipeline, as seen in Figure 4.1. The CDA module (Figure 4.2) of the Coner pipeline annotated all corpus documents with unfiltered recognised *method* and *dataset* named entities and generated a final selection of 10 papers to receive human feedback on. We conducted this experiment on unfiltered extracted entities to enable us to generate a clean comparison between different filtering mechanisms on unfiltered extracted entities; Coner human filtering vs different TSE-NER filtering setups vs TSE-NER filtering enhanced by Coner human filtering. Setup of CDA threshold parameter values, as described in Section 4.1: $conf_{nr} = 11$, $min\_entities_{nr} = 15$, $top\_papers_{nr} = 50$ and $candidate\_papers_{nr} = 5$.

6.2.1 Coner Human Feedback Assessment

In this section we summarise the document selection process, look into the nature of user feedback itself and evaluate how it conflicts with or supports the TSE-NER heuristics.

[25] http://www.wis.ewi.tudelft.nl/

[26] https://www.elastic.co/

Document Selection

We wanted to maximise the efficiency of the time spend by each evaluator giving feedback on named entities, to ensure each instance of feedback leads to the highest knowledge gain achievable in the current setup. Therefore we used our novel Coner Document Analyser, as described in Section 4.1, to select papers of interest. We selected 10 papers as a final selection, after manual inspection, namely [33, 18, 20, 3, 14, 42, 22, 5, 32, 6]. The papers were selected from multiple conferences of interest, being The Web Conference (3 papers), ACL (3 papers), ICWSM (2 papers) and VLDB (2 papers) and with diverse research subjects, to minimise bias towards any type of paper structure, methodology introduced or results presented.

The 10 documents selected for this evaluation contain overall 255 distinct NER recognised *dataset* unfiltered entities, and 85 distinct NER recognised *method* unfiltered entities. The average number of times each paper has been cited is 581.

Evaluators & Human Feedback Evaluation

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. The 10 human evaluators are randomly and uniformly assigned to the documents such that each document is processed by at least 3 evaluators. We asked the evaluators to check all recognised named entities of the *method* type or *dataset* type for correctness, or ignore them if they were unsure. Also, evaluators were instructed to focus their efforts on highlighting any word or sequence of words they encounter in the documents that they deemed to be of type *dataset* or *method* as newly recognised named entities. We informed the evaluators that the highlighted entities are unfiltered and have a noisy nature as result, thus many of them should be filtered out. As seen in Table 6.1, in total 60 *dataset* and 113 *method* entities were added by users. On average, each document contained 45.2 distinct entities of either type (after users added named entities). Remarkably, 13.5% of annotated named entities were classified as relevant to both facets, which in practice is highly implausible. For NER extracted entities, it is possible that both models trained for different facets recognise the same technical named entities, thus there is a degree of uncertainty about the semantic meaning of a named entity in that specific context. Surprisingly, even some user added entities were selected for both facets. This scenario is possible when users add the same named entity multiple times across different papers, and either a difference in user expertise or entity paper occurrence context results in recognition of different facets. Some examples of doubly classified entities are given in Section 6.2.2 and further analysis is conducted in Section 6.2.5.

	Dataset	Method	Dataset & Method
User added	60 (19.0%)	113 (57.1%)	3 (4.9%)
NER extracted	255 (81.0%)	85 (42.9%)	58 (95.1%)
Total	**315 (69.7%)**	**198 (43.8%)**	**61 (13.5%)**

Table 6.1: Number of unfiltered entities in evaluated viewer papers. Total number distinct named entities for both facets combined is 452. Some entities are classified as both facets by NER, so the sum of total percentages of *dataset* and *method* entities is more than 100%

We need at least feedback of three users on each named entity in the viewer to enable the correct calculation of entity type labels through majority vote in the CFA module of Coner, as described in Section 4.3. We obtained this minimum threshold of three users' feedback on the recognition correctness on 271 *dataset* entities (94.8%) and 158 *method* entities (94.0%). The average number of feedback instances on user added named entities of either type was just above 3 and on NER recognised entities just above 5. The lower amount of feedback instances on user added named entities can be attributed to the fact that evaluators can only give feedback on user entities after they have been added, therefor not every evaluator will see every user added entity if they go through every document once.

The evaluators showed quite varying task completion times for giving feedback on entities contained in a document, with an average of 19.8 seconds to provide feedback for both facets of a single entity, while the fastest evaluator only needed 6.3 seconds and the slowest 34.3 seconds. We discussed with some evaluators about their experience giving feedback in the Coner viewer, and it became clear that not only is there a difference in domain expertise and background knowledge, but also users' feedback approach itself differs. The slowest users choose to read all the text around named entities, to get some deeper insight of the context in which a named entity occurs, while the faster evaluators only skimmed through the text and gave feedback on entities along the way.

The evaluators were not forced to rate all occurrences of recognised entities, thus relevance feedback was only given when an evaluator was certain of his/her input to a certain degree. The average percentage of extracted entities (highlighted in the Coner Viewer) each evaluator gave feedback on in one paper is 65.9%, with practically similar percentages for *dataset* and *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 facet relevance (e.g. the named entity `Microsoft` can reference a dataset created by Microsft or the actual company itself), therefore feedback input was not always given on these entities. Second, some bigram or trigram *method* entities were recognised with additional useless trailing words (e.g. `question taggings have`), therefore also not receiving feedback from some evaluators. Furthermore, some evaluators simply switched papers or closed the browser before finishing a paper.

Table 6.2 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 6.2 indeed show the effectiveness of collaborative feedback on TSE-NER. Note that not all entities have a majority vote of feedback instances from evaluators, as it is possible to either have an equal amount of votes for relevant and irrelevant or less than three votes for any entity of either type. Interestingly enough, not all of the named entities added by users were rated as relevant for their intended type; false positives for 25.9% *dataset* and for 11.7% *method* user added entities. This means that it is crucial to also receive user feedback from evaluators on named entities other users added to ensure the quality of human feedback. It is inescapable that evaluators have differences in their knowledge and expertise of each scientific domain, which influences their decision making. Judging whether a word or sequence of words in a piece of scientific text is a named entity of a specific type is very much a subjective process. Some evaluators noted that they only deemed newly introduced names of datasets or methods in that publication as a named entity of such type (e.g. `Slashdot Zoo Corpus, Exp4 Algorithm`), while others were more broad and judged general dataset and method names as relevant (e.g. `Twitter, Wikipedia dataset, data mining, linear regression`). Also, the Coner viewer version used during this experiment does not allow yet for users to remove an entity after they added it, thus a evaluator's lack of fluency with the viewer's user interface resulted in some entities added that were later clearly marked as irrelevant by the other users (e.g. `hits, hotel`). However, it is possible to change your feedback on an entity in the viewer. Only 115 times an evaluator changed his/her feedback input on an entity's facet, which translates to 2.8% of total feedback events. Changing feedback input can be explained by correcting previous miss clicks or a change in opinion about entity relevance when getting more semantic context.

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

Table 6.2: Comparison of false positive rates, resulting from users' majority vote on relevance of unfiltered extracted entities (Coner Filtering), for both newly user added and NER extracted entities for two types of entities: *Dataset* and *Method*

Inter-annotator Agreement

Both percent agreement between evaluators and inter-annotator agreement are interesting measures to explore how much evaluators' feedback overlap. But both approaches have their strengths and limitations. Percent agreement is favourable because it is easy to interpret, a simple percentage of overlap between two evaluators, but it does not take guessing of entities' type into consideration. This experiment was conducted in a private lab setting, all evaluators were graduate or post-graduate level and have a background in Computer Science. This means there is a slimmer chance of users guessing, compared to public crowdsourcing, an entity's type when they are not certain, but it could still result in an overestimate of true agreement among raters [25]. The Cohen's Kappa statistical value between 0 and 1 is less intuitive to read, but does take the possibility of evaluators guessing into account, which is essential when measuring crowdsourced feedback.

We measured the inter-annotator agreement between evaluators using Cohen's Kappa. We calculated the average Cohen's Kappa between the 10 evaluators for each entity facet by calculating the average of Cohen's Kappas for each user pair that shared a joint subset of evaluated named entities of size 20 or larger. On average, Cohen's Kappa for *dataset* entities is 0.51, while for *method* entities it is 0.63. We retrace that inter-annotator agreement score for *method* entities is higher, because a bigger percentage of evaluated named entities were added by users for the *method* facet compared to the *dataset* facet, as seen in Table 6.1. Named entities added by users have on average a much lower false positive rate compared to named entities automatically extracted by the NER (Table 6.2).

6.2.2 Qualitative Entity Inspection

This section gives an insight in the nature of named entities rated as most or least relevant for a facet by evaluators. Table 6.3 gives an overview of entities with lowest relevance scores for *dataset* and *method* facets. The nature of most dataset sampled entities rated as irrelevant is; should be method name instead or part of the named entity is actually a dataset name, but it contains useless trailing words (e.g. `digg interface` instead of `digg`). For method recognised entities they are rated as being dataset names instead or words are related to general method or programming approach, but not quite names of long-tail named entities.

	User added	NER extracted

Dataset	`traditional source,`
	`high-dimensional vector space,`
	`x-tree, news`
		`digg interfaces,`
		`logistic regression,`
		`interconnection index,`
		`evaluation results,`
		`buneman, acyclic subgraph`

Method	`t-density, hdvs,`
	`long-tail distribution,`
	`leaving out one feature`
		`digg, flickr,`
		`wikipedia, dynamic programming,`
		`system description,`
		`signed clustering`

Table 6.3: Sample of *Dataset* and *Method* annotated entities examples with *lowest Coner relevance scores* from users

See Table 6.4 for an overview of entities with highest relevance scores. Most *dataset* entities are names of datasets, but curiously also the entity `datasets` itself receive a high relevance score. For some *method* entities it is unclear wether method names should be selected with the trailing words `algorithm, classifier, model`, etc. or just the name of the method itself.

Surprisingly, for some user added entities it was not clear from the human relevance scores what type it belongs to. For example the named entity `slash` was added by an evaluator as *dataset* entity, but then receive a majority vote relevance score of 0.5 for both the *dataset* and *method* facets. The entity `slash` appears in the [18] paper, and is part of the dataset name `slashdot`. Table 6.5 presents an overview of other example entities that received feedback from an adequate number of evaluators, but with divided opinions on entity type relevance. These entities with uncertainty relevance scores fall into three main categories:

- Entities that are related to the facet, but are too generic, e.g. `signed networks, news article, news feed, data base, etc.` for *dataset* and `algorithm, decision rule and used search algorithm` for *method*. This could be explained by a difference in domain expertise or interpretation of what belongs to a certain type between evaluators.
- User added entities with some type of syntax error, like letter(s) missing from the entity (`igned symmetric spectral ranking`) or the inclusion of a line-break hyphen (`wall street jour-nal`). The resulting uncertainty in relevance scores can again be interpreted as a diversion in subjective judgement of entity types.

	User added	NER extracted

Dataset	`twitter dataset,`
	`wikipedia dataset,`
	`slashdot zoo corpus,`
	`google news,`
	`mashable, very large database`
		`digg, flickr,`
		`wikipedia, datasets,`
		`network data, bugzilla`

Method	`k-nearest neighbour, exp3`
	`nearest neighbour search,`
	`digg promotion algorithm,`
	`subjectivity classifier,`
	`signed symmetric spectral ranking`
		`similarity search,`
		`spectral ranking,`
		`reinforcement learning,`
		`markov process,`
		`linucb, linucb algorithm,`
		`logistic regression,`
		`acyclic subgraph`

Table 6.4: Sample of *Dataset* and *Method* annotated entities examples with *highest Coner relevance scores* from users

- Entities that do now belong to either of the types no matter how the interpretation, e.g. `vldb and web services`.

The entities in this sample indicate that type assignment of entities through human feedback is undoubtedly a subjective process, which emphasises that feedback from multiple evaluators is paramount to the quality of the relevance scores.

	User added	NER extracted

Dataset	`signed networks, slash,`
	`news article, news feed,`
	`online news feed, wall street jour-nal`
		`data base, data collection,`
		`elements storing,`
		`hdvss, news source`

Method	`10-foldcross validation, algorithm,`
	`igned symmetric spectral ranking,`
	`fmf, used search algorithm`
		`decision rule, laplacian,`
		`vldb, web services`

Table 6.5: Sample of *Dataset* and *Method* annotated entities examples with *uncertainty Coner relevance scores* from users

We also see a lot of doubly recognised entities, which are entities that were classified as both facets by TSE-NER, e.g. entities that appeared both in Table 6.3 and Table 6.4; `digg, flickr, wikipedia, logistic regression, acyclic subgraph`. For most of these doubly facet extracted entities, it becomes undeniably clear with human feedback which facet each entity actually belongs to. We will further investigate this doubly classification phenomenon in the discussion in Section 6.2.5. Finally, occasionally evaluators select new entities when they were also already extracted by the NER, like NER extracted `wikipedia` and user still added `wikipedia dataset`.

6.2.3 Comparison Filtering Techniques: Coner vs TSE-NER

Table 6.6 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 feedback from at least three evaluators in the 10 viewer papers; 315 *dataset* and 198 *method* entities. As illustrated in Table 6.6, the FCB filtering technique is more strict than EMV alone for the *dataset* facet, but less strict for the *method* facet. This can be explained by the larger percentage of user added named entities for the *method* facet compared to the *dataset* facet, with user added named entities having a much lower average false positive rate compared to NER extracted entities (Table 6.2).

	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 6.6: Comparison of entity retention rate (percentage of unfiltered entities kept by filter) between Coner and TSE-NER filter techniques of all viewer entities for both facets (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

To get a deeper 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 6.7); if an entity is kept by a filter for a facet, but was voted as irrelevant for a facet 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 it's retention of entities), which means there are a significant number of entities that were recognised as irrelevant for a facet by human judgement, but TSE-NER heuristic filtering was unable to do so.

	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 6.7: Percentage of false positives in the remaining filtered entity sets of TSE-NER filtered heuristics compared to Coner human filtered entities for two types of entities: *Dataset* and *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

We also considered the false negatives which were excluded by the filtering techniques but were labelled as relevant by majority of evaluators (Table 6.8). The PMI filtering as explained in [27] achieved the highest precision among the TSE-NER filtering techniques in their evaluation. Table 6.8 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 ensemble filtering heuristic, which is regarded as most effective in terms of F-Score by [27], the average false negatives rate is 57.8%. Also, in Table 6.6 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.

	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 6.8: Percentage of false negatives in the remaining filtered entity sets of TSE-NER filtered heuristics with regards to Coner filtered entities for two types of entities: *Dataset* and *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

Finally, Table 6.7 and Table 6.8 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 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.

6.2.4 Comparison NER Performance: Coner vs TSE-NER

In this section we measure and compare the precision, recall and F-score of TSE-NER with and without the Coner feedback on TSE-NER extracted entities. We use the same test dataset of manually annotated text snippets, for both *dataset* and *method* facet models, already employed in the baseline work [27].

Table 6.9 compares the performance of TSE-NER with and without Coner feedback in terms of precision, recall and F-Score. We filtered all TSE-NER extracted and Coner human added technical named entities from the 10 viewer documents (315 *dataset* and 198 *method* entities) with two different setups. For TSE-NER we used the ensemble filter setup, because evaluation from [27] clearly illustrates that the F-score for ensemble filtering is higher than other filtering techniques. The PMI filtering technique did achieve the highest precision among the filtering techniques [27], but we aim to optimise both precision and recall, thus a higher F-Score is more relevant for our performance goals.

For Coner we employed the TSECB module, to improve recall by enhancing the training set, and the FCB module, to boost precision of the trained model, as described in Section 5. After that we ran the TSE-NER training cycle for the second iteration with the two different setups and measured the performance of the resulting NER models with our gold standard test dataset. From Table 6.9 we can clearly conclude that the recall and F-Score increased, with a tiny decrease in precision, for both facets when employing the Coner pipeline.

	Dataset (P/R/F)	Method (P/R/F)
TSE-NER	**0.77**/0.40/0.53	**0.78**/0.12/0.21
Coner	0.74/**0.52**/**0.61**	0.77/**0.14**/**0.24**

Table 6.9: Comparison of performance of *TSE-NER* and *Coner* in terms of Precison/Recall/F-Score for two type of unfiltered entities: *Dataset* and *Method*

6.2.5 Discussion

One of our goals of this experiment was to identify the nature of human feedback through our online crowdsourcing document viewer, introduced earlier as RQ1. On average it took 19.8 seconds to judge the relevance of a named entity for both facets, which is the equivalent of rating about 180 named entities in 1 hour of website usage. Even in our limited lab setting with 10 evaluators using Coner for less than 1 hour, we received sufficient feedback (average of 4 feedback instances per entity) on 94.4% of named entities in order to calculate the Coner relevance score (EFRS). All of the results were achieved with an average inter-annotator agreement between pairs of evaluators with joint entity subsets of 0.51 for *dataset* and 0.63 for *method*, which fits our expectation for this type of noisy context dependent data. We retrace the difference to the divergence of amount of user added entities for *dataset* versus *method* entities, with more user added entities for the *method* facet (Table 6.1). Named entities added by users have on average a much lower false positive rate compared to named entities automatically extracted by the NER (Table 6.2).

Like we hypothesised for RQ2, Table 6.2 and Table 6.6 indicate that on average human feedback is indeed more selective than the most effective ensemble TSE-NER filtering heuristic. Furthermore, Table 6.7 and Table 6.8 illustrate that by leveraging human feedback 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 steps, i.e., false negatives. We saw that even the most precise (PMI) and effective (EMV) filtering heuristics result in high false positives and false negatives rates compared to human feedback.

We already demonstrated the effectiveness of the Coner performance boosting mechanism in Table 6.9. Critically, even with our limited availability of resources (evaluators and time) we revealed that human feedback on unfiltered named entities can boost recall up to 23.1% in our TSE-NER training setup. However, a small decrease in precision of up to 3.9% (0.77 to 0.74) was also apparent, indicating that the filtering boosting mechanism needs further fine tuning. However, these results are promising, especially for the boost of the expansion step, as recall has improved significantly.

Table 6.1 and the manual qualitative inspection overview in Section 6.2.2 gave us a clue about the phenomenon of doubly classified entities; a single entity recognised by TSE-NER trained classifiers for both facets. This scenario is in practice highly unlikely, because a technical named entity in a given sentence is either of type *dataset* or *method*. Out of the 452 unfiltered viewer entities 13.5% was doubly classified. We further investigated to see if this pattern also appears for entities recognised in the entire corpus, as described in Section 6.1. For *ensemble* heuristic filtered entities the percentage of doubly classified entities is 14.0% and for *PMI* filtered entities 18.5%. For unfiltered corpus recognised entities it is even higher; 43.1%. The filtering techniques already greatly reduced the percentage of doubly recognised entities compared to unfiltered entities, but even with ensemble filtering, it appears that there is a semantic overlap between the *word2vec* embedding spaces of both facets. Naturally, humans are much better at separating the semantic spaces of doubly classified entities, because we see the full context in which named entities occur in documents and are superior at extracting unambiguous semantic meanings from natural text compared to machines.

These issues motivated us to setup a second user experiment, where the named entities shown in the viewer are selected in a more intelligent fashion; we only visualise high impact named entities that are doubly classified by TSE-NER, to fully take advantage the evaluators' strength to judge and separate entity typing in specific contexts. This mechanism aims to maximise the value of each human feedback effort to help scale our crowdsourcing approach by reducing our biggest bottleneck of the amount of human time necessary to achieve performance improvement. Another more common traditional approach to smart entity selection is to perform uncertainty sampling from our ensemble filtered entities; solely select entities for evaluation that received a 50/50 majority vote from TSE-NER ensemble filters, therefore indicating conflicts between different filters and resulting in uncertainty. The vision behind that methodology is to utilise human judgement to clear up these cases of entity relevance uncertainty and decide if it is a true positive or true negative for that particular facet. We choose our custom doubly classified methodology for two reasons. We assume and reason that uncertainty entities have a lower chance than doubly classified entities of being relevant for any of the facets. Doubly classified entities have been recognised confidently by a majority of filters for both classifiers. Almost without fail human feedback will judge the entity as irrelevant for at least one of the facets, thus resulting in a high confidence true positive for one facet and true negative for the other. Traditional uncertainty sampling only allows us to include or exclude an entity based on human feedback for one facet (the one it was extracted as), we cannot argue anything about the relevance for other facets, as that entity was not recognised for that facet in the first place. So arguably, feedback on each doubly classified entity results in increased potential knowledge gain and a bigger filtering correction compared to traditional uncertainty sampling.

The following Section 6.3 evaluates our smart selection entity evaluation setup. The explicit entity feedback relevance scores calculated from human input are incorporated into the expansion and filtering phases of TSE-NER to evaluate whether performance has increased, and thus tackles RQ3.

6.3 Experiment 2: Human Feedback on Doubly Classified Entities

The purpose of this experiment is to obtain knowledge about how our smart doubly classified entity selection mechanism affects the boosting ability of Coner on TSE-NER's extraction performance in terms of recall, precision and F-score. This experiment is based on the hypothesis that receiving human feedback only on entities that were classified by TSE-NER trained classifiers for both facets (typed as both *dataset* and *method*) will lead to higher potential knowledge gain per human feedback interaction, and thus increase the trained NER classifiers' overall performance with our Coner boosting technique.

Our previous results in Section 6.9 revealed that human feedback collected on unfiltered entities in a limited lab setting where evaluators were instructed to focus on adding new entities, resulted in a recall increase of up to 23.1%. In our previous experiment we did not employ a smart entity selection mechanism, thus it was up to the users to decide which entities to provide feedback on. Most users choose to give feedback on the majority of highlighted entities, even when irrelevance for a facet was unambiguous and clear (e.g. `whereas, and, without, although, anyone`). Part of these clearly irrelevant and noisy named entities would have already been filtered by TSE-NER heuristics, resulting in an partial overlap between human filtering and TSE-NER heuristic filtering. Unsurprisingly, precision did not increase when employing the Coner boosting mechanism, as the knowledge gain of feedback on each entity was suboptimal. In order to outperform the previous setup, we introduce the doubly classified entity preprocessing step in the CDA module; we only consider entities for analysis that where recognised by NER classifiers for both facets and kept by both facet ensemble filtering heuristics. Furthermore evaluators were instructed to focus more on filtering out existing entities, to tailor the trained TSE-NER classifiers towards higher precision performance.

The training process is similar to the one describes in Section 6.2 (two facets of *dataset* and *method*, 50 seed terms and one iteration), but with one major difference; we only selected and visualised candidate named entities for human feedback that were detected by both trained NERs. Our experiment's analysis consists of three sections; assessment of human feedback in Section 6.3.1, analysis of Coner boost performance in Section 6.3.2 and our interpretation of the achieved results in Section 6.3.3.

6.3.1 Coner Human Feedback Assessment

Similarly to our document selection in Section 6.2, we utilised our novel CDA module with smart entity selection enabled to annotate and pick 28 papers from 4 conferences in our document corpus; 13 papers from VLDB, 9 papers from The Web Conference, 4 from SIGIR and 2 fro ICWSM.

We recruited 15 graduate-level/post-graduate-level volunteers and showed them the Coner workflow as introduced in our first experiment, but instructed them to mainly focus their efforts on filtering existing entities. Table 6.10 lists all the named entities resulting from the manual annotation session; 298 entities in total with 109 recognised entities for both facets, as we only selected doubly classified entities, and 189 added entities by users. Remarkably, even when instructed to focus on filtering entities, evaluators tend to still add new entities they encounter around other highlighted entities. Also, just like experiment 1, more methods were discovered than datasets giving us a suspicion that method occurrences are more frequent in most computer science papers.

	Dataset	Method	Dataset & Method
User added	52 (32.2%)	139 (56.0%)	2 (1.8%)
NER extracted	109 (67.7%)	109 (44.0%)	109 (98.2%)
Total	**161 (54.0%)**	**248 (83.2%)**	**111 (37.2%)**

Table 6.10: Number of doubly classified filtered entities in evaluated viewer papers. Total number distinct named entities for both facets combined is 298. Most entities are classified as both facets by NER, so the sum of total percentages of *dataset* and *method* entities is more than 100%

We calculated the following experiment statistics; average entity rating time of 15.3 seconds, 3.4% of feedback events identified as opinion change, average number of feedback inputs for user added entities was 3 and for extracted entities 6 (95.4% of entities received three or more ratings) and it took users on average 163 seconds to rate a single paper. The average entity rating time decreased from 19.8 seconds in the experiment 1 to 15.3 seconds, which we attribute to a new feature introduced in the Coner viewer; when rating an entity as YES for a type, the other facet's relevance is automatically rated as NO.

From the overview of false positives rates in Table 6.11 it is straightforward to see that the overall average false positive rate (FPR) has gone down compared to our first experiment's results as seen in Table 6.2. *Dataset* FPR went from 80.4% in experiment 1 to 60.9%, with lower FPR for both user added and NER extracted entities. However, the *method* FPR did increase from 27.4% in experiment 1 to 32.1%; we instructed evaluators to focus less on adding entities compared to experiment 1, thus the ratio between NER extracted and user added entities is slightly higher (respectively 85 vs 113 for experiment 1 and 109 vs 139 for experiment 2), and since the FPR of NER extracted entities is much higher than that of user added entities, it naturally drives up the total FPR value of the *method* facet. Surprisingly, the difference between FPR of unfiltered NER extracted *method* entities in experiment 1 and doubly filtered NER extracted *method* entities in experiment 2 is non existent; from 57.9% to 58.7%.

	Dataset (FP%)	Method (FP%)
User added	17.3%	10.9%
NER extracted	81.7%	58.7%
Total	**60.9%**	**32.1%**

Table 6.11: Comparison of false positive rates, resulting from users' majority vote on relevance of doubly classified filtered extracted entities (Coner Filtering), for both newly user added and NER extracted entities for two types of entities: *Dataset* and *Method*

6.3.2 Comparison NER Performance: Coner vs TSE-NER

In this section we measure and compare the precision, recall and F-score of TSE-NER with and without the Coner feedback on TSE-NER doubly classified entities. We use the same test dataset of manually annotated text snippets, for both *dataset* and *method* facet models, already employed in the baseline work [27]. Furthermore we conducted the performance analysis in similar fashion to the setup in Section 6.2.4, but with incorporation of Coner relevance scores of the 298 doubly classified and added entities resulting from the latest user experiment. Table 6.12 reveals an increase in precision for both *dataset* and *method* facet classifiers when incorporating the Coner innovative modules, while maintaining recall and F-score levels. The next section discussed the achieved performance improvements.

	Dataset (P/R/F)	Method (P/R/F)
TSE-NER	0.66/**0.62**/0.64	0.56/**0.21**/0.30
Coner	**0.70**/0.61/**0.65**	**0.59**/0.20/0.30

Table 6.12: 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*

6.3.3 Discussion

Surprisingly, the difference between FPR of NER extracted *method* entities in experiment 1 and 2 is non existent; from 57.9% to 58.7%, even when in theory entity preprocessing with ensemble filtering heuristics should significantly decrease the FPR rate in experiment 2 compared to experiment 1 on unfiltered entities. This finding further enforces the conclusion in [27] that filtering based on assumptions and heuristics is flawed, and confirms our hypothesis that humans are superior at judging relevance of entities for a specific facet.

Also, we noticed that our entity addition feature in the Coner viewer produces in rare cases entities with syntax errors. For example, when an evaluator selects an entity that includes a word that is split across lines and separated by a hyphen, it is included with text selection. Even with our built in ability to edit an entity's text before you add it, this scenario sometimes results in added entities like `in-ternet movie database imdb` and `huff-ington post` which currently cannot be automatically detected and corrected by the Coner system.

Even in a limited lab setting with 15 users we have shown that the NER's performance can be boosted in terms of precision or recall depending on the instructions we give evaluators and the smart entity selection setup. Recall increases when focused primary on adding entities besides filtering, which resulted in a longer per-entity rating time on average of 19.8 seconds in experiment 1 and 15.3 seconds in experiment 2. Arguably it takes more time to search for new relevant long-tail named entities to add than it takes to give feedback on already highlighted entities. Also in experiment 2 we introduced the automatic opposite facet rating feature, where it takes less clicks to rate YES for one facet and automatically rate NO for the other. With our introduction of the doubly entity selection method and user instructions to focus on filtering highlighted entities we showed the potential of the system with an increase of up to 5.7% in precision.

Experiments to evaluate the original TSE-NER algorithm in [27] showed that training a NER model for more iterations results in a decrease of precision performance, in return of a stable increase of recall. The Coner pipeline is already able to improve precision in our small lab setting, which is a promising preliminary result for the expected performance increase when Coner is deployed in a full production system, which would result in the continuous build up of a massive dictionary of irrelevant and relevant named entities for each facet. We describe this scenario in more details in the future work Section 8.

7 Conclusion

In this thesis we introduced Coner, a collaborative approach for long-tail named entity recognition in scientific publications. Coner extends TSE-NER, a previously established technique for iterative training of NER algorithms using distant supervision [27]. 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), only required a manually annotated small seed set of known named entities of the desired type as training input. Unfortunately, this reliance on automatic heuristic expansion and filtering also limited 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 collaborative feedback on entities is realised with an *interactive viewer component* (CIDV), which allows users to elicit feedback on the correctness of recognised entities and add new entities they deem relevant unobtrusively while reading the document.

One of the advantages of Coner is that its user feedback on recognised entities requires only little cognitive effort (i.e., simple YES/NO feedback on recognised entities and text selection to add new entities), and can be easily elicited while users use the system without much interruption. Even when the cognitive effort is kept to an absolute minimum, proper *incentivisation* techniques are required to motivate the system's users to contribute feedback. During our studies we choose to set the scope of the user system to a private lab setting; only a limited group of evaluators from graduate/post-graduate level took part in our user experiments. Scaling Coner to a full scale public crowdsourcing solution requires further investigation, which is discussed in the future work Section 8.

In order to make human feedback scalable and maximise the potential knowledge gain for each instance of feedback, we introduced a smart entity selection process in our novel *document analyser module* (CDA). Instead of relying 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 actively steer this process by merely selecting entities that were doubly classified, so recognised as relevant for both facets, by the trained NER models. This approach is designed to make the most out of each user interaction and take advantage of the strength of humans to determine the semantic meaning of a named technical entity in a specific context of a paper. We then calculate explicit feedback scores for each entity (through majority vote) with our *feedback analyser module* (CFA) to facilitate the incorporation of human feedback in the TSE-NER algorithm.

In this first work incorporating the user base into the NER training process, we focus on augmenting the TSE-NER training cycle in two areas. First, augment the filter step of TSE-NER, allowing manual user feedback to overrule decisions which would have been taken by the heuristics, with the goal to boost precision. Manual feedback supersedes all heuristic 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. The TSE-NER expansion strategies aim to enhance the size and variety of the set of training terms and sentences, to improve overall recall of the trained NER model. We hypothesised that using user feedback to also *expand and boost the term and sentence set* (instead of only filtering) in each TSE-NER iteration should considerably increase the recall of the overall approach. We boost the expansion step by adding relevant user added entities to the seed term set for the next iteration of training. We evaluated both points of improvement in two user experiments.

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. In our lab experiments with a repository of 11,598 data science publications and 15 users, we could show that 76.1% of all entities detected by TSE-NER in the publications selected for evaluation were indeed false positives (94.3% for *dataset* and 57.9% for *method*). Even for named entities recognised and added by users the average false positive rate was 18.8%, accentuating the importance of human feedback from multiple evaluators on all types of entities. The average Cohen's Kappa between each pair of evaluators that shared a joint subset of evaluated named entities of size 20 or larger is 0.51 for *dataset* entities and 0.63 for *method* entities. We measured that on average 3.3% of total feedback events indicated a change of relevance rating for an entity, which can be explained by people changing their judgement on that entity or resolving a previous miss click. Crucially, it only took evaluators on average 15.3 seconds (depending on smart entity selection and evaluators' instructions) to rate a single entity's relevance for both facets, confirming our hypothesis that human feedback is indeed scalable with our lightweight and intuitive Coner viewer user experience.

Out of all the available filtering setups, the FCB achieved both the lowest average false positives rate (6.4%) and false negative rate (0.7%), which translates to a performance increase of up to 13.1% (0.53 to 0.61) in F-Score and 23.1% (0.40 to 0.52) in recall when incorporating human feedback on unfiltered entities in the TSE-NER training cycle.

Furthermore we illustrated that 14.0% of extracted entities were in fact doubly classified by both NERs even with the most optimal ensemble heuristic filtering setup, which emphasises the importance of incorporating human feedback on these doubly classified entities into the NER training cycle. These issues motivated us to build and evaluate a smart selection mechanism to only visualise entities to users which were doubly classified. With our smart entity selection implementation and a second round of manual annotation by 15 users we clearly demonstrated that performance was boosted with an increase of up to 5.7% in precision.

The single most striking conclusion is that even with our limited availability of human resources we revealed that TSE-NER's performance can be boosted by up to 23.1% in terms of recall, up to 5.7% in terms of precision and up to 13.1% in terms of F-score depending on the setup of our smart entity selection mechanism and instructions given to evaluators.

8 Future Work

We believe that this research has given rise to multiple directions in need of further investigation to explore the full potential of human collaborative named entity recognition. In future work it would be of great value to interact with the NER's surrounding user system in a more sophisticated way. Coner's full potential can be leveraged when it is integrated into an existing production system, like a larger scale digital library; every uploaded document is analysed by the CDA module and named entities selected by our smart entity selection mechanism are visualised in our Coner viewer. The vision is that the Coner viewer will be the default viewer for users to read any papers in the digital library, and because eliciting feedback is designed to be cognitively lightweight, receive continuous feedback from the system's users on a number of papers magnitudes bigger than our private lab experiment conducted so far. Digital libraries users, or so called "public scientists", are incentivised by the potential improvement of the digital library's search and exploration capabilities resulting from the improved trained NER models. Human time spend eliciting feedback on entities is still our biggest bottleneck in boosting the named entity extractors' performance, thus a larger number of evaluators should directly impact the overall performance of a future production system. Also, an automatic configuration of the Coner pipeline that retrains models periodically overnight, weekly or in other increments would be interesting, as it allows for a type of self learning where incremental results can steer the process of finding the most valuable papers and entities to get human feedback on. This setup will even allow for classifiers to keep up and conform to trends emerging in papers of a digital library grouped by research domains; when specific named entities of a topic on average occur more or less frequently over time, and we receive human feedback on these entities, the NER will fetch more training data for this topic and improve the retrieval of these entities through our self-learning setup.

In our evaluation we used human feedback on doubly faceted extracted entities to boost the expansion and filter steps, but this feedback could be further utilised to generate additional information on true negative examples in training data. When an entity is extracted for more than one facet, it is almost without fail a false positive for at least one of the facets and at most a true positive for one of the facets. If an entity is a false positive for certain facet(s), then that information could be leveraged to provide the next TSE-NER training cycle with high certainty entity facet information. Overall training data context is improved by incorporating both the true positive and true negative labels for different facets of a single entity. Furthermore, it is vital that more research is done on alternative methods of smart entity selection (besides doubly classified entities selection), i.e., based on uncertainty sampling, to explore potential enhancement in impact and scalability of human feedback.

Beyond those immediate challenges, several other interesting opportunities remain to be explored:

- A module that implements a configurable version of the Coner system that is more geared towards either recall or precision, so it can be used to give beneficial results for different applications of typed named entity recognition.
- Allow users of the Coner viewer to upload their own PDF documents of interest. This increases incentive and motivation to use the platform; not all users are not willing to read and give lightweight feedback on "random papers in their field of interest", so it is beneficial if they could upload documents they were going to read regardless. For this to work in practice, we believe it is absolutely vital that the user interface and experience of the Coner viewer is superior to alternative document viewers.
- The current version of the Coner viewer is a private crowdsourcing task solution, because feedback about facet relevance of entities is acquired by enlisting the services of a manually selected group of trusted people, who are motivated through incentives. The next step would be to scale the Coner viewer from a private crowdsourcing platform to a full fledged public crowdsourcing task solution. This would greatly improve the quantity of feedback received, but could negatively impact the quality of feedback due to several factors, as analysed in [2]. To a certain extend, this could be offset using appropriate *incentivisation* techniques, by motivating a 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 or research digital library setting, such issues need to be addressed by for example user reputation management [7] or different voting consensus techniques [11].

References

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

[2] Omar Alonso, Daniel E Rose, and Benjamin Stewart. Crowdsourcing for relevance evaluation. In *ACM SigIR Forum*, volume 42, pages 9–15. ACM, 2008.

[3] Roja Bandari, Sitaram Asur, and Bernardo A Huberman. The pulse of news in social media: Forecasting popularity. *ICWSM*, 12:26–33, 2012.

[4] Marco Brambilla, Stefano Ceri, Emanuele Della Valle, Riccardo Volonterio, and Felix Xavier Acero Salazar. 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, 2017.

[5] Hai Leong Chieu and Hwee Tou Ng. Teaching a weaker classifier: Named entity recognition on upper case text. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 481–488. Association for Computational Linguistics, 2002.

[6] Sara Cohen, Jonathan Mamou, Yaron Kanza, and Yehoshua Sagiv. Xsearch: A semantic search engine for xml. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 45–56. VLDB Endowment, 2003.

[7] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)*, 51(1):7, 2018.

[8] Luca de Alfaro, Marco Faella, Vassilis Polychronopoulos, and Michael Shavlovsky. Incentives for truthful evaluations. *arXiv preprint arXiv:1608.07886*, 2016.

[9] Tome Eftimov, Barbara Koroušić Seljak, and Peter Korošec. A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PloS one*, 12(6):e0179488, 2017.

[10] Carsten Eickhoff and Arjen de Vries. How crowdsourcable is your task. In *Proceedings of the workshop on crowdsourcing for search and data mining (CSDM) at the fourth ACM international conference on web search and data mining (WSDM)*, pages 11–14, 2011.

[11] Kinda El Maarry, Ulrich Güntzer, and Wolf-Tilo Balke. 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, 2015.

[12] Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 80–88, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[13] Sean Louis Goldberg, Daisy Zhe Wang, and Tim Kraska. Castle: crowd-assisted system for text labeling and extraction. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[14] Tad Hogg and Kristina Lerman. Stochastic models of user-contributory web sites. In *ICWSM*, 2009.

[15] Kenichi Iwatsuki, Takeshi Sagara, Tadayoshi Hara, and Akiko Aizawa. Detecting in-line mathematical expressions in scientific documents. In *Proceedings of the 2017 ACM Symposium on Document Engineering*, pages 141–144. ACM, 2017.

[16] Ece Kamar and Eric Horvitz. Incentives for truthful reporting in crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '12, pages 1329–1330, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[17] Mayank Kejriwal and Pedro Szekely. 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, 2017.

[18] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, pages 741–750. ACM, 2009.

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

[20] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650. ACM, 2010.

[21] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier, 1994.

[22] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

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

[24] George Mathew, Amritanshu Agarwal, and Tim Menzies. Trends in topics at SE conferences (1993-2013). *arXiv preprint arXiv:1608.08100*, 2016.

[25] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282, 2012.

[26] Sepideh Mesbah, Alessandro Bozzon, Christoph Lofi, and Geert-Jan Houben. Describing data processing pipelines in scientific publications for Big Data injection. In *Workshop on Scholary Web Mining (SWM)*, Cambridge, UK, feb 2017.

[27] Sepideh Mesbah, Alessandro Bozzon, Christoph Lofi, and Geert-Jan Houben. Long-tail entity extraction with low-cost supervision. https://2018.eswc-conferences.org/paper_8/, 2018.

[28] Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Facet Embeddings for Explorative Analytics in Digital Libraries. In *Int. Conf. on Theory and Practice of Digital Libraries (TPDL)*, Thessaloniki, Greece, sep 2017.

[29] Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Facet embeddings for explorative analytics in digital libraries. In *International Conference on Theory and Practice of Digital Libraries*, pages 86–99. Springer, 2017.

[30] Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Semantic annotation of data processing pipelines in scientific publications. In *European Semantic Web Conference*, pages 321–336. Springer, 2017.

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[32] Tadashi Nomoto and Yuji Matsumoto. Supervised ranking in open-domain text summarization. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 465–472. Association for Computational Linguistics, 2002.

[33] Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 295–302. Association for Computational Linguistics, 2002.

[34] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[35] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. 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, 2004.

[36] Tarique Siddiqui, Xiang Ren, Aditya Parameswaran, and Jiawei Han. Facetgist: Collective extraction of document facets in large technical corpora. In *Int. Conf. on Information and Knowledge Management*, pages 871–880. ACM, 2016.

[37] Min Song, Hwanjo Yu, and Wook-Shin Han. Developing a hybrid dictionary-based bio-entity recognition technique. *BMC medical informatics and decision making*, 15(1):S9, 2015.

[38] Chen-Tse Tsai, Gourab Kundu, and Dan Roth. Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1733–1738. ACM, 2013.

[39] Suppawong Tuarob, Sumit Bhatia, Prasenjit Mitra, and C Lee Giles. Algorithmseer: A system for extracting and searching for algorithms in scholarly big data. *IEEE Transactions on Big Data*, 2(1):3–17, 2016.

[40] Daniel Vliegenthart, Sepideh Mesbah, Christoph Lofi, and Akiko Aizawa. Coner: A Collaborative Approach for Long-Tail Named Entity Recognition in Scientific Publications.

[41] Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. Perspectives on crowdsourcing annotations for natural language processing. *Language resources and evaluation*, 47(1):9–31, 2013.

[42] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 98, pages 194–205, 1998.

[43] Yazhou Yang and Marco Loog. Active learning using uncertainty information. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2646–2651. IEEE, 2016.

[44] Bweijunl Zheng, Wei Zhang, and Xiaoyu Fu Boqin Feng. A survey of faceted search. *Journal of Web engineering*, 12(1&2):041–064, 2013.

[45] Jingbo Zhu, Huizhen Wang, Benjamin K Tsou, and Matthew Ma. Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on audio, speech, and language processing*, 18(6):1323–1331, 2010.

[46] Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K Tsou. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational*

Linguistics-Volume 1, pages 1137–1144. Association for Computational Linguistics, 2008.