ECCE: Entity-centric Corpus Exploration Using Contextual Implicit Networks

Julian Schelb
julian.schelb@uni-konstanz.de
University of Konstanz
Konstanz, Germany

Maud Ehrmann
maud.ehrmann@epfl.ch
École polytechnique fédérale de Lausanne
Lausanne, Switzerland

Matteo Romanello
matteo.romanello@unil.ch
University of Lausanne
Lausanne, Switzerland

Andreas Spitz
andreas.spitz@uni-konstanz.de
University of Konstanz
Konstanz, Germany

ABSTRACT

In the Digital Age, the analysis and exploration of unstructured document collections is of central importance to members of investigative professions, whether they might be scholars, journalists, paralegals, or analysts. In many of their domains, entities play a key role in the discovery of implicit relations between the contents of documents and thus serve as natural entry points to a detailed manual analysis, such as the prototypical 5Ws in journalism or stock symbols in finance. To assist in these analyses, entity-centric networks have been proposed as a language model that represents document collections as a cooccurrence graph of entities and terms, and thereby enables the visual exploration of corpora. Here, we present ECCE, a web-based application that implements entity-centric networks, augments them with contextual language models, and provides users with the ability to upload, manage, and explore document collections. Our application is available as a web-based service at http://dimtools.uni.kn/ecce.

CCS CONCEPTS

• **Computing methodologies** → *Information extraction*; • **Information systems** → *Document representation.*

KEYWORDS

Entity network, cooccurrence network, corpus exploration

ACM Reference Format:
Julian Schelb, Maud Ehrmann, Matteo Romanello, and Andreas Spitz. 2022. ECCE: Entity-centric Corpus Exploration Using Contextual Implicit Networks. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487553.3524237

1 INTRODUCTION

An ever-expanding wealth of information is created, stored, and disseminated in the form of unstructured text. In recent years, scalable neural language models have enabled the processing of this information at previously infeasible scales. However, in smaller settings where big language data is unavailable or not directly applicable, such language models alone cannot yet fully replace human investigative skills in knowledge acquisition. Consequently, semi-supervised approaches that include humans in the loop are often necessary in deriving insights from document collections.

A common thread in such investigations is the initial exploration of a corpus, which tends to rely on the visualization of documents, topics, or entity relations as networks or graphs to link information within and across documents. The benefits of graph-based indexing and visualization tools are tremendous and have been emphasized in such diverse domains as news investigation [5, 17], the political and social sciences [8], or the digital humanities [6].

One versatile type of language model that supports graph-based exploration are implicit entity models [16, 17], which have been demonstrated to work well in the extraction of entity relations [14] and topics in the news domain [15]. Here, we introduce ECCE, a web-based application that uses an implicit network representation for the interactive exploration of document collections containing English text. In contrast to previous contributions, it takes a domain-agnostic approach and works as an end-to-end web tool in which users may upload, annotate, and explore their own corpora.

Contributions. We make three primary contributions.

- We implement a web service for end-to-end implicit network exploration of user-defined documents and corpora.
- We demonstrate how implicit entity networks can be improved by taking advantage of contextual language models.
- We propose on-demand clustering of edge contexts to support interactive exploration by minimizing pre-computation.

2 RELATED WORK

Due to the diversity of domains in which text data are analyzed, numerous related applications can be found in the literature.

Interactive annotation. Several tools facilitate the interactive annotation of corpora via online user interfaces. Examples include ET, which enables users to edit, annotate, and query corpora and assess the quality of annotations [1]. Similarly, TALEN is a web interface for interactively annotating low-resource entities in corpora [10]. In contrast to ECCE, the above tools provide annotation capability as the primary function, while we use it solely as the means to facilitate the entity-centric exploration of the corpus.

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04.
https://doi.org/10.1145/3487553.3524237

Knowledge base integration. To utilize or extend structured information, some approaches include domain-specific knowledge bases. CurEx [9] identifies entities in unstructured documents, determines the relationship strength based on cooccurrence, and links them to entities in a knowledge base to generate an explorable entity network. SEMANNOREX supports the semantically enriched search of news by linking entities to an ontology whose structure is then used to derive semantic similarity scores for retrieval operations [7]. In contrast to ECCE, these tools do not focus on implicit entity relations described within a corpus.

Graph-based topic modelling. Topics are a staple of document analysis, and for the visual exploration of topics, named entities often play an important role. ContraVis demonstrates the use of visual topic modeling for the comparison of corpora [8]. InfraNodus provides network analysis capabilities for word cooccurrence networks and employs community detection to identify topics [12]. The most closely related topic-centric approach to ECCE is TopExNet [15], which visualizes graph topics in entangled news streams, yet does not allow the user to work with their own document data.

Exploration of implicit relations. Implicit relations between entities tend to be a focus of corpus exploration. Receptor is a platform for the extraction of implicit relations between entities and events in sensitive corpora via graph search [11]. In the news domain, focusing on the joint occurrence of entities quickly leads to the detection of events, which typically describe (co)occurrences of persons at a specific place and time. TiCCo is a tool for the graph-based temporal exploration of such events in news [3]. Similarly, EVELIN [14] supports the graph-based retrieval of entities, their relations and implicit descriptions thereof from news articles and Wikipedia texts through faceted search. Some other works focus on more narrow domains, such as the analysis of relations between companies that jointly occur in news articles [5].

In contrast to the above tools and methods, ECCE focuses not just on the extraction and exploration, but on a comprehensive pipeline in which the user annotates and explores their documents.

3 THEORETICAL BACKGROUND

ECCE is built using implicit entity networks [16] and contextual implicit entity networks [17] for corpus representation and indexing. We briefly describe the underlying intuition of these models.

Implicit Entity Networks (IEN). IENs were proposed as a joint representation of entities, terms, and the sentences and documents that contain them to enable interactive entity-centric retrieval [16]. Conceptually, IENs are entity cooccurrence networks in which nodes $\mathcal{V}$ correspond to entities in the documents. Each entity is associated with an entity type and entities are unique by entity name and type. Edges $\mathcal{E}$ between entities are weighted by the inverse exponential of the cooccurrence distance δ (typically counted in units of sentences). To represent the full corpus, parallel edges between entities $v, w \in \mathcal{V}$ are aggregated over all instances $\mathcal{I}_{v,w}$ in which they cooccur to derive an edge weight ω as

$$\omega(v, w) = \sum_{i \in \mathcal{I}_{v,w}} \exp -\delta_i(v, w). \tag{1}$$

Due to the exponentially decaying weights, it is sufficient to consider cooccurrences within some context window of fixed size c. While the method performs well for the interactive exploration of documents, the aggregation over all cooccurrence contexts is problematic when entities cooccur in multiple different contexts.

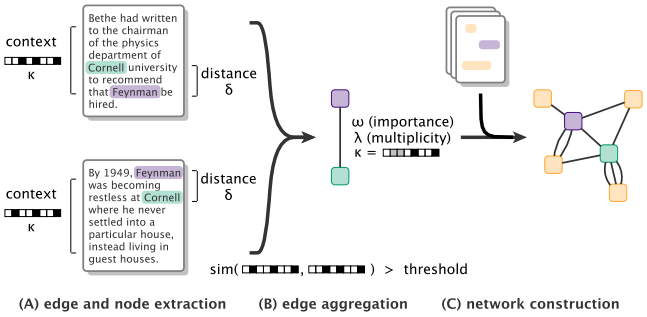

Figure 1: Schematic overview of contextual implicit entity network construction. (A) Entities are extracted as nodes, edge weights are derived from cooccurrence distances δ. The context is used to compute text embeddings κ. (B) Edges with similar contexts are aggregated, their weights are combined to ω, and the contexts are averaged. (C) A joint network representation is created from all cooccurrences in the corpus.

Contextual Implicit Entity Networks (CIEN). To address the ambiguity of entity cooccurrences in differing contexts, CIENs are constructed to take into account the context [17]. Instead of aggregating all parallel edges, individual edges between two given entities are attributed with a context embedding κ that is derived from the text in the cooccurrence window. Edge contexts are clustered using some suitable similarity measure (e.g., cosine similarity) and algorithm (the number of contexts is a-priori unknown, so density-based clustering such as DBSCAN [2] is sensible). Only edges with similar contexts are aggregated, such that the resulting network retains some parallel edges. For an overview, see Figure 1.

Neural context embeddings. In the original implementation of CIENs [17], edge contexts are derived by averaging static word2vec embeddings of all words in the context window. To take advantage of advances in neural language modeling, we instead compute contextual sentence embeddings with a transformer model [13] directly from the entire context window as input.

On-demand aggregation of edge contexts. In the default implementation, parallel edges in CIENs are aggregated during precomputation when documents are added, which is expensive due to the large number of cooccurrences that scales quadratically with the number of entities inside a context window. In practice, we observe that the majority of relations will never be explored by the user, so an IEN representation instead of a CIEN representation is sufficient. Thus, as a compute-efficient alternative, we propose the on-demand aggregation of edges, in which only a single edge is added between any two adjacent entities in the graph visualization until the context for this entity pair is explored, at which time the contexts for the selected edge are aggregated as needed.

4 SYSTEM ARCHITECTURE

Based on the above methodology, we describe the system architecture for extracting, displaying, and interacting with contextual implicit entity network representations of a document corpus. For an overview of the system, see Figure 2.

4.1 Document Database

For the storage of documents and corpora, we use MongoDB as a text-centred NoSQL database. Documents are grouped into corpora and assigned unique object identifiers, which can be used by users to retrieve, manipulate or delete data at a later time. Data is transferred to and from the database in JSON format. We also use the database for caching document annotations to avoid costly repeated computation and the loss of manual annotation data.

4.2 NLP and ML Modules

For processing the data, we use four major modules: (1) text preprocessing and named entity annotation, (2) sentence embedding, (3) implicit network extraction, and (4) edge clustering.

Text preprocessing and annotation. For segmentation, tokenization, stop word detection, and named entity recognition we use the Python package *spaCy*. Out of its tag set, we use PERSON, ORG, GPE, NORP, LOC and WORK_OF_ART as default. The user may also activate tagging for CARDINAL, DATE, MONEY, PRODUCT, TIME, PERCENT, QUANTITY, EVENT, ORDINAL, FAC, LAW, and LANGUAGE.

Sentence embedding. We employ a contextual language model to compute context embeddings for entity cooccurrences. Specifically, we use the pre-trained version of sentence-BERT [13] that is implemented in the Python package *sentence transformers* with the *multi-qa-distilbert-cos-v1* model from Huggingface.

Implicit network extraction. For the extraction of IENs (before context embeddings are added), we implemented a Python version of the original algorithm [16], which we also make available as a separate Python package[1]. The network data is passed to the frontend in JSON format. By default, we use a context window of size $c = 2$ sentences to each side for determining cooccurrences, but the user may adjust this value.

Edge clustering. Following the approach originally proposed for clustering edges in CIENs, we use DBSCAN [2] as implemented in the *sklearn* Python package. We specify the cosine distance as metric, and set $\varepsilon = 0.25$ and a minimum sample size of 1.

4.3 Backend

The backend operates as a RESTful web service that connects the database and the NLP modules to the frontend by providing API endpoints for corpus management, document annotation, and entity network extraction. It is implemented using the Python package *Flask*. The programming logic is completely decoupled from the frontend and can serve HTTP requests by external applications using the standardized API endpoints. Thus, the backend allows for modules to be distributed and shared across machines, which is particularly useful regarding scalability when considering the GPU requirements of language models in the NLP module.

4.4 Web Frontend

The frontend of ECCE is implemented as a web interface using *Vue.js* and consists of three primary views: document and corpus management, annotation interface, and network exploration (for details, see Section 5). For interactive UI elements and implementing the responsive layout, we use the *BootstrapVue* library. To visualize

[1] https://pypi.org/project/implicit-word-network/

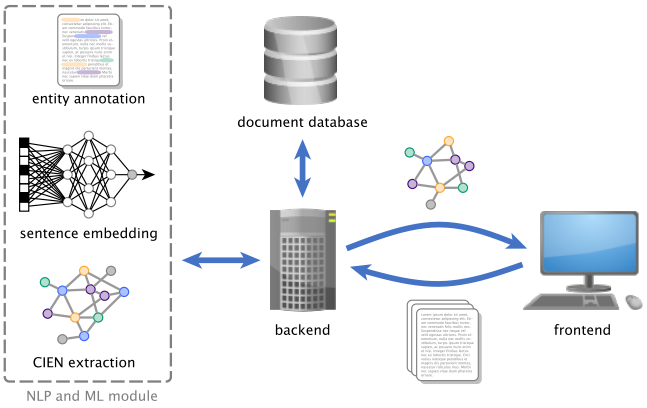

Figure 2: Schematic view of the application architecture.

the CIEN, we use the JavaScript library *D3*. Communication between the frontend and the backend is handled via HTTP requests. The application itself is stateless and the server-side storage and retrieval of documents relies on unique identifiers, for which we use MongoDB's object IDs.

5 FUNCTIONALITY AND DEMONSTRATION

In the following, we present the functionality of the application by describing a typical usage scenario, which is organized into three main stages as shown in Figure 3: (A) uploading documents and managing corpora, (B) annotating the documents, and (C) exploring the extracted contextual implicit entity network. ECCE is available as a web demonstration, including usage instructions (for the URL, we refer the reader to the abstract).

A: Document upload and corpus management. We designed ECCE to allow users to analyze their own documents and corpora, either by entering them as text input or by uploading plain text in a CSV file. All documents are stored in the underlying database and assigned a unique identifier that allows them to be retrieved and edited at a later time by providing this identifier. We deliberately rely on a loginless scheme to avoid the necessity of user management and allow easy yet secure data sharing between users.

B: Document annotation. Once the user has created or uploaded and edited all documents, they can proceed to the named entity recognition phase. Named entity annotations are suggested automatically, highlighted in the text, and color-coded by type. Furthermore, stop words and punctuation tokens are detected (so they can be discarded later during the construction of the entity network). Since even state-of-the-art entity recognition is known to be error-prone in the best of cases and on data from well-researched and popular domains [4], the user is also given the option to correct all annotations, which may be added, deleted, or given a different entity type. Where necessary, individual tokens can be merged to reflect compound or nested entities.

C: Entity network exploration. Once the user is satisfied with the annotations, the annotated documents are used to construct a CIEN of the entire corpus, which is then displayed to the user using a force-directed graph layout. Nodes are color-coded by entity type, and the frequency with which an entity occurs in the corpus is represented by node size using linear scaling. Entities

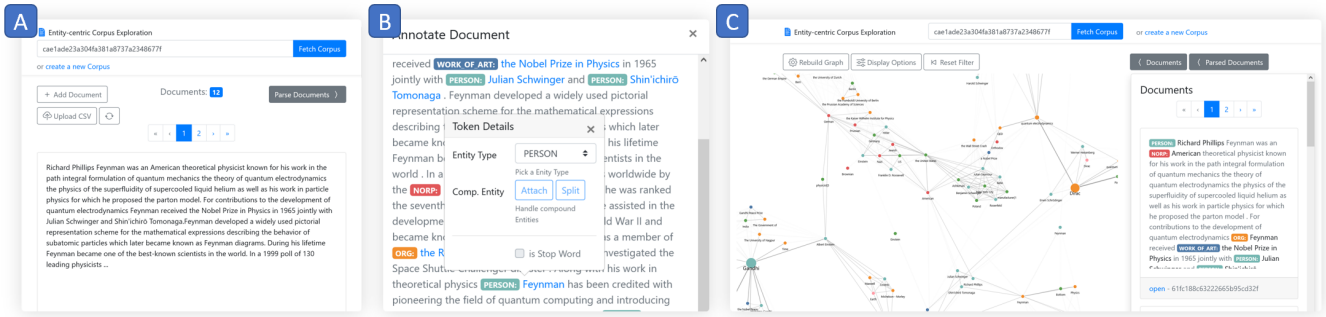

Figure 3: Web interface of ECCE. (A) Documents can be added, edited, and deleted individually or imported in bulk by uploading a text file. (B) Named entity annotations for the documents (computed with *spaCy*) are displayed and can be edited, deleted, or merged into compound entities. (C) The extracted entity network is visualized using force-directed layout, allowing the exploration of entity cooccurrences in context. Entity types are encoded as node colors, edge width denotes relation strength between entities. Parallel edges are clustered on-demand during exploration by using neural contextual sentence embeddings of their context as features. Document provenance can be viewed for occurrences (nodes) and cooccurrences (edges).

that cooccur in the corpus are connected by edges, whose width and opacity corresponds to the IEN edge weight, as introduced in Section 3. This enables the user to identify entities of high importance in the corpus, as well as the relations between them. When a node is clicked, further details about the corresponding entity are displayed, including the sentences and documents in which the entity occurs. When edges are selected, the cooccurrence contexts are automatically clustered on-demand as described in Section 4 and provenance information is displayed for all retrieved clusters of contexts in which the incident entities occur, again including sentences and documents. The text contents of documents are displayed in a separate pane, so the user may simultaneously explore entity relations in the network and in the documents.

Since the implicit networks may grow to inconvenient sizes or densities when larger document collections are used as input, we provide display options for filtering the network view. By default, we display the 150 most frequently occurring entities and all edges, but thresholds for entity frequency and edge weight can be adjusted via two sliders. Alternatively, a filter may be applied to display only entities of a certain type. Navigation between the three stages is continuous to enable the user to update annotations or documents.

6 SUMMARY AND OUTLOOK

We presented ECCE, a web application for the exploration of user-defined text corpora by leveraging contextual implicit entity networks, which we adapted to take advantage of pre-trained contextual language models. Our work demonstrates the utility of implicit entity networks for transforming unstructured text into a structured and easily explorable network representation. By introducing on-demand clustering of cooccurrence edges based on context, we mitigate the runtime constraints of implicit entity networks and ensure interactive response times during the exploration.

Ongoing work. We are working on the integration of alternate entity recognition libraries and tag sets, and are including pre-trained language models for additional languages. Our aim is to further improve the versatility of ECCE as a tool for exploring text data originating from arbitrary domains and languages.

REFERENCES

[1] Elvis de Souza and Cláudia Freitas. 2021. ET: A Workstation for Querying, Editing and Evaluating Annotated Corpora. In *EMNLP '21*. https://doi.org/10.18653/v1/2021.emnlp-demo.5
[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD '96*. https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf
[3] Philip Hausner, Dennis Aumiller, and Michael Gertz. 2020. TiCCo: Time-Centric Content Exploration. In *CIKM '20*. https://doi.org/10.1145/3340531.3417432
[4] Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. Few-Shot Named Entity Recognition: An Empirical Baseline Study. In *EMNLP '21*. https://doi.org/10.18653/v1/2021.emnlp-main.813
[5] Thomas Kellermeier, Tim Repke, and Ralf Krestel. 2019. Mining Business Relationships from Stocks and News. In *MIDAS Workshop at PKDD '19*. https://doi.org/10.1007/978-3-030-37720-5_6
[6] Florian Kerschbaumer, Linda von Keyserlingk-Rehbein, Martin Stark, and Marten Düring. 2020. *The Power of Networks: Prospects of Historical Network Research*. Routledge. https://doi.org/10.4324/9781315189062
[7] Amit Kumar, Govind, and Marc Spaniol. 2021. Semantic Search via Entity-Types: The SEMANNOREX Framework. In *WWW '21 Companion*. https://doi.org/10.1145/3442442.3458607
[8] Tuan Le and Leman Akoglu. 2019. ContraVis: Contrastive and Visual Topic Modeling for Comparing Document Collections. In *WWW '19*. https://doi.org/10.1145/3308558.3313617
[9] Michael Loster, Felix Naumann, Jan Ehmueller, and Benjamin Feldmann. 2018. CurEx: A System for Extracting, Curating, and Exploring Domain-Specific Knowledge Graphs from Text. In *CIKM '18*. https://doi.org/10.1145/3269206.3269229
[10] Stephen Mayhew and Dan Roth. 2018. TALEN: Tool for Annotation of Low-resource ENtities. In *ACL '18*. https://doi.org/10.18653/v1/P18-4014
[11] Hitarth Narvala, Graham McDonald, and Iadh Ounis. 2020. Receptor: A Platform for Exploring Latent Relations in Sensitive Documents. In *SIGIR '20*. https://doi.org/10.1145/3397271.3401407
[12] Dmitry Paranyushkin. 2019. InfraNodus: Generating Insight Using Text Network Analysis. In *WWW '19*. https://doi.org/10.1145/3308558.3314123
[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP '19*. https://doi.org/10.18653/v1/D19-1410
[14] Andreas Spitz, Satya Almasian, and Michael Gertz. 2017. EVELIN: Exploration of Event and Entity Links in Implicit Networks. In *WWW '17 Companion*. https://doi.org/10.1145/3041021.3054721
[15] Andreas Spitz, Satya Almasian, and Michael Gertz. 2019. TopExNet: Entity-Centric Network Topic Exploration in News Streams. In *WSDM '19*. https://doi.org/10.1145/3289600.3290619
[16] Andreas Spitz and Michael Gertz. 2016. Terms over LOAD: Leveraging Named Entities for Cross-Document Extraction and Summarization of Events. In *SIGIR '16*. https://doi.org/10.1145/2911451.2911529
[17] Andreas Spitz and Michael Gertz. 2018. Exploring Entity-centric Networks in Entangled News Streams. In *WWW '18 Companion*. https://doi.org/10.1145/3184558.3188726