CollabKG: A Learnable Human-Machine-Cooperative Information Extraction Toolkit for (Event) Knowledge Graph Construction

Xiang Wei, Yufeng Chen, Ning Cheng, Xingyu Cui, Jinan Xu and Wenjuan Han✉
Beijing Key Lab of Traffic Data Analysis and Mining,
Beijing Jiaotong University, Beijing, China
{22120436, chenyf, ningcheng, 22120357, jaxu, wjhan}@bjtu.edu.cn

Abstract

In order to construct or extend entity-centric and event-centric knowledge graphs (KG and EKG), the information extraction (IE) annotation toolkit is essential. However, existing IE toolkits have several non-trivial problems, such as not supporting multi-tasks, and not supporting automatic updates. In this work, we present `CollabKG`, a learnable human-machine-cooperative IE toolkit for KG and EKG construction. Specifically, for the multi-task issue, `CollabKG` unifies different IE subtasks, including named entity recognition (NER), entity-relation triple extraction (RE), and event extraction (EE), and supports both KG and EKG. Then, combining advanced prompting-based IE technology, the human-machine-cooperation mechanism with Large Language Models (LLMs) as the assistant machine is presented which can provide a lower cost as well as a higher performance. Lastly, owing to the two-way interaction between the human and machine, `CollabKG` with learning ability allows self-renewal. Besides, `CollabKG` has several appealing features (*e.g.*, customization, training-free, and label propagation) that make the system powerful and high-productivity. We holistically compare our toolkit with other existing tools on these features. Human evaluation quantitatively illustrates that `CollabKG` significantly improves annotation quality, efficiency, and stability simultaneously.

Keywords: Information Extraction, Human-machine Cooperation, Knowledge Graph Construction Toolkit

1. Introduction

Entity-centric and event-centric knowledge graphs (KG and EKG, collectively referred to as (E)KGs) are structured semantic knowledge bases for describing concepts and their relations in the physical world (Zou, 2020; Guan et al., 2022). These (E)KGs are playing an increasingly important role in many downstream tasks and applications, such as search engine (Zhao et al., 2021; Yang et al., 2020), question-answering (Bao et al., 2016; Souza Costa et al., 2020) and commonsense reasoning (Lin et al., 2019). With the dynamic changes in the Internet content, existing (E)KGs still need to be completed in the general domains, and even need to be constructed from scratch in emerging and specialized domains (Kejriwal, 2022; Chen et al., 2020). In this regard, the information extraction (*i.e.*, IE) is an effective way to construct or complement (E)KGs (Luan et al., 2018; Li et al., 2020).

There are tons of existing open-source tools for IE labeling, both automatically and manually. However, these tools still have some non-trivial issues that hinder the applicability and effectiveness of real-world applications. First, there are various IE tasks, such as named entity recognition (NER), entity-relation triple extraction (RE), and event extraction (EE). However, most tools only support one or two of these tasks (Nghiem and Ananiadou, 2018; Islamaj et al., 2020; Stewart et al., 2019; Li et al., 2021). To our knowledge, very few open-source manual annotation toolkits function EE annotation. Hence, KG and EKG often use individual tools instead of a unified one while parts of KG and EKG share similar architecture. Secondly, most tools only support either automatic or manual labeling (Stenetorp et al., 2012; Zhang et al., 2022; Bikaun et al., 2022; Jin et al., 2021; Tang et al., 2020). However, a human-machine cooperative system has proven to outperform both standalone agents and humans working alone (Bien et al., 2018). Lastly, even if some tools support both automatic and manual labeling, the machine itself cannot learn from human annotations as feedback (Klie et al., 2018; Abrami et al., 2019). Besides the above three major issues, minor issues exist like requiring large amounts of data for training, which makes labeling time-consuming and not suitable for low-resource scenarios (Jin et al., 2021). Therefore, it is crucial to build an IE toolkit that is multi-tasking, human-machine-cooperative, training-free, *etc*.

Based on the above clues, we propose `CollabKG`[1] (Fig.1), a learnable human-machine-cooperative IE annotation toolkit for KG and EKG construction. The main contributions are described as follows:

1 `CollabKG` is an open-source IE annotation toolkit that unifies NER, RE, and EE tasks, integrates KG and EKG, and supports both English and Chinese languages.

✉ Corresponding author.

[1] Video: https://www.youtube.com/channel/UCsadiRvhW9dsmn4KtRDCaFg Code: https://github.com/cocacola-lab/CollabKG

LREC-COLING 2024, pages 3490–3506
20-25 May, 2024.

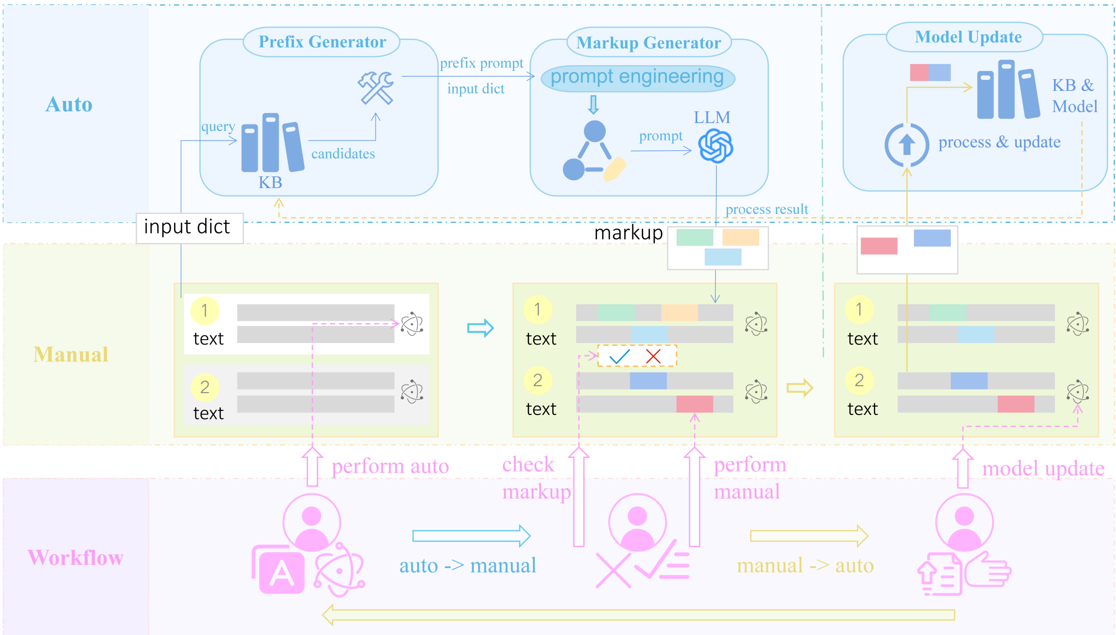

Figure 1: Illustration of the human-machine-cooperative workflow for `CollabKG`.

2 `CollabKG` combines automatic and manual labeling to build a learnable human-machine cooperative system. In particular, humans benefit from machines and meanwhile, manual labeling provides a reference for the self-renewal of the machines online. Additionally, `CollabKG` is designed with many other appealing features (Sec.2) making it more powerful and productive.

3 Extensive human evaluation suggests that `CollabKG` can significantly improve the effectiveness and efficiency of manual annotation, as well as reduce variance (Sec.4).

2. Core Functions

We compare `CollabKG` quantitatively with existing open-source IE annotation tools in Tab.1 and perform the following summary about the functions:

Unification: Only 22% of the tools support all three IE tasks for KG and EKG construction.

Human-machine-cooperation: Only 22% of the reviewed tools support both automatic and manual labeling, but they do not support learnability or unification feature.

Learnability: Only on 11% of the reviewed tools, human labeling can provide a reference for the machine to help annotate more effectively and efficiently.

Other functions: Only our toolkit `CollabKG` along with Quickgraph support **annotation propagation** and **document clustering**[2]. But the annotation propagation of Quickgraph cannot be applied to Chinese texts. Additionally, CollabKG is **training-free**, suitable for low-resource scenarios, and flexible for **customization** (Sec.2.2 and App.H).

2.1. Implementation of Unification

`CollabKG` is designed to unify three IE tasks including NER, RE, and EE (Fig.2 and App.D). We uniformly model and integrate the schemes into a series of triples `(S-Type:S,R,O-Type:O)`. To fit the unified scheme, we design the annotation mode of `Entity` and `Relation`. `Entity` mode considers entities in NER, subjects, and objects in RE, arguments, and triggers in EE. Similarly, `Relation` mode considers relations in RE and roles in EE.

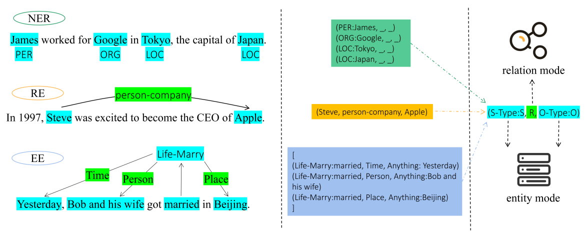

Figure 2: Illustration of unifying NER, RE, and EE tasks.

[2]AP: Sub-string offset matching to perform relation/entity propagation (Bikaun et al., 2022). DC: Clustering documents to promote annotator consistency (Bikaun et al., 2022).

	NER	RE	EE	Auto	Manual	Learn	TF.	AP.	DC.
`CollabKG` **(ours)**	✓	✓	✓	✓	✓	✓	✓	✓	✓
DeepKE (Zhang et al., 2022)	✓	✓	-	✓	-	-	-	-	-
CogIE (Jin et al., 2021)	✓	✓	✓	✓	-	-	-	-	-
Quickgraph (Bikaun et al., 2022)	✓	✓	-	-	✓	-	-	✓	✓
BRAT (Stenetorp et al., 2012)	✓	✓	✓	-	✓	-	-	-	-
WebAnno (Yimam et al., 2013)	✓	-	-	-	✓	-	-	-	-
SLATE (Kummerfeld, 2019)	✓	✓	✓	-	✓	-	-	-	-
INCEpTION (Klie et al., 2018)	✓	✓	-	✓	✓	-	-	-	-
TeamTat (Islamaj et al., 2020)	✓	✓	-	-	✓	-	-	✓	-
TextAnnotator (Abrami et al., 2019)	✓	✓	-	✓	✓	-	-	-	-
FITAnnotator (Li et al., 2021)	✓	-	-	✓	✓	✓	-	-	-
APLenty (Nghiem and Ananiadou, 2018)	✓	-	-	✓	✓	✓	-	-	-
Redcoat (Stewart et al., 2019)	✓	-	-	-	✓	-	-	-	-
SALKG (Tang et al., 2020)	✓	✓	-	-	✓	-	-	-	-
RESIN (Wen et al., 2021)	✓	✓	✓	✓	-	-	-	-	-
REES (Aone and Ramos-Santacruz, 2000)	-	✓	✓	-	✓	-	-	-	-
FLAIR (Akbik et al., 2019)	✓	-	-	✓	-	-	-	-	-
OpenNRE (Han et al., 2019)	-	✓	-	✓	-	-	-	-	-
ODIN (Valenzuela-Escárcega et al., 2015)	-	-	✓	✓	-	-	✓	-	-

Table 1: Illustration of core functions for `CollabKG` and comparison with recent existing open-source IE toolkits. **Auto**: Automatic labeling. **Manual.**: Manual labeling. **Learn**: Learnable. **TF.**: Train free. **AP.**: Annotation Propagation. **DC.**: Document Clustering.

2.2. Implementation of Human-Machine-Cooperation

`CollabKG` is designed to support manual labeling and automatic labeling simultaneously.

Attr.	Description
`isEntity`	Entity or relation
`suggested`	Markup state (suggested or accepted)
`_id`	Identifier for current markup
`name`	Name for current entity or relation type
`labelId`	Id for current entity or relation type
`source`	Id for subject associated with current relation
`target`	Id for object associated with current relation
`start`	Start position for current entity
`end`	End position for current entity
`entityText`	Span for current entity

Table 2: Attributes in the `markup`. **Attr.**: Attribute.

Manual Labeling We use a dictionary structure called "`markup`" (Bikaun et al., 2022) to record and manage tags. We divide markups into entity markups and relation markups. The attribute `isEntity` is used to distinguish them. Moreover, there are many other attributes that are classified as common attributes or private attributes as shown in Tab. 2. Specially, common attributes mainly include `suggested`, `_id`, `name`, `labelId` while private attributes include `source`, `target`, `start`, `end`, `entityText`, *etc*.

Automatic Labeling We adopt ChatIE (Wei et al., 2023), a SOTA approach for zero-shot information extraction based on prompting ChatGPT. It is training-free and suitable for low-resource scenarios. Most importantly, ChatIE is flexible for unification because the type list allows customization (App.E). Given the above advantages, we use ChatIE as the backbone of our automatic annotation algorithm. Then, we improve the algorithm in EE task by introducing a new trigger words extraction stage through the inclusion of the trigger-related prompt template. For example, the following prompt template is used:

```
When the event type of the given
sentence above is "<event-type>",
please recognize the corresponding
trigger.  The trigger is the word or
phrase that most clearly expresses
event occurrences.\nOnly answer the
trigger, no extra word.  The trigger
is:.
```

2.3. Implementation of Learnability

In Fig.1, we present the bi-directional interaction workflow between manual and automatic labeling.

Auto → Manual Users can click a specific button to make the system perform automatic labeling. `CollabKG` first collects current information to form the input dict, including task, text, and language. Then, the prefix generator searches the knowledge base (which stores high-frequency manual markups) with the current text to generate a prefix prompt. The prefix prompt is fed in the model to enhance domain knowledge, such as *Note: the type of Google is ORG; the relation between James and Google is person-company; ....* Finally, the markup generator generates the final prompt, predicts results, and converts them to markups for human reference. After these steps, the attribute of these markups is set to the `suggested` state, indicating that these labels are pending. Users can make checking that either click the `Apply` button

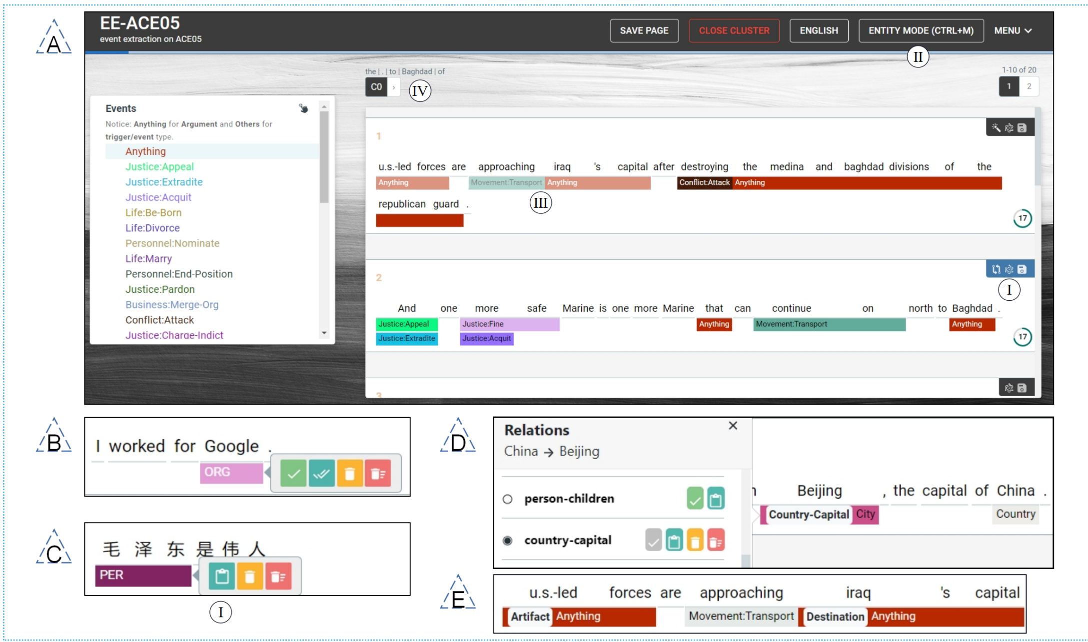

Figure 3: Main Interface of Annotation. B-C for NER, D for RE and A or E for EE.

to accept them or Delete button to discard them. By utilizing this workflow, users can refer to the automatic annotation results as a reference.

Auto ← Manual CollabKG deploys a self-renewal function. It processes high-frequency and informative manual markups and converts them to specific schemes. Then, it updates CollabKG by adding these specific schemes into the knowledge base, which will be used to build a more powerful prefix generator in later iterations. We refer to this process as "**learnability**". With a single click, the annotators can store the currently annotated texts as well as the convinced NER/RE/EE markups in the knowledge base. During the next automatic annotation process, the prefix generator queries the knowledge base using the current text to get similar texts and their markups. Then, the prefix generator will fill the texts and their markups into the prefix template. For example, the following prefix template is used: `Note: the type of [entity] is [entity type]; the relation between [subject] and [object] is [relation] ... in [text]..`

This learnability brings many benefits, such as identifying emerging concepts, domain-specific terms, ambiguous words, *etc*. For example, give a text from an electronic product report "*The middle class likes using Apple.*", the model may not recognize *Apple* as ORG. But if humans manually mark *Apple* as ORG in another sentence "*New Yorker really like Apple phones.*" and CollabKG perform model updating. Then the prefix prompt "*Note: Apple is ORG in ...;*" will make the model successfully recognize it.

3. Toolkit Usage

Core functions and the corresponding implementation make CollabKG powerful and highly productive. Next, we show the toolkit usage by introducing the UI (Fig.3) and description.

3.1. Annotation

Manual Labeling There are entity and relation modes. In entity mode, the user selects the span and clicks the corresponding type to complete the annotation (Fig.3 B-C, **NER**). In relation mode, the user clicks on the subject and then selects the relation type associated with the corresponding object (Fig.3 D-E, **RE** and **EE**). Moreover, switching between the two modes requires only one click on the toggle (Fig.3 A.II) and multi-label and nested markups are supported.

Automatic Labeling By clicking the corresponding button (Fig.3 A.I), the predicted results of the machine (*i.e.*, ChatIE) will be transformed into highlighted markups in **semitransparent** color (Fig.3 A.III) to be displayed on the annotation interface. Users can further choose to accept or discard them by operating the tooltips containing actionable icons *apply one*, *apply all*, *delete one*, *delete all* (Fig.3 B).

Learnability By clicking the corresponding button (Fig.3 A.I), the markups in the accept state will be passed into the backend to update model (Sec.2.3).

Label Propagation Hundreds of entities or relations can be suggested in other sentences with a single click (Fig.3 C.I) and users can make checks by operating the tooltips.

	NER					RE					EE				
	P	*R*	*F1*	*Var*	*Time*	*P*	*R*	*F1*	*Var*	*Time*	*P*	*R*	*F1*	*Var*	*Time*
BRAT	64.2	62.0	63.0	0.51	00:42:34	48.1	44.2	45.8	0.69	01:46:50	36.0/70.5	27.5/47.5	31.1/56.7	0.79/0.53	01:34:36
CollabKG **(Auto)**	**86.3**	65.1	74.2	-	-	**85.5**	51.5	64.2	-	-	**44.2**/82.4	38.2/68.9	41.0/75.0	-	-
CollabKG **(Human)**	64.2	62.0	63.0	0.51	00:41:27	48.1	44.2	45.8	0.69	01:27:17	36.0/70.5	27.5/47.5	31.1/56.7	0.79/0.53	01:12:49
CollabKG	81.7	**79.3**	**80.4**	**0.39**	**00:40:42**	70.8	**71.9**	**70.9**	**0.41**	**01:24:01**	43.4/**83.4**	**45.5/71.3**	**44.2/76.9**	**0.37/0.09**	**01:11:21**

Table 3: Results on three tasks. For EE, the left/right numbers represent Arg-C/Trig-C. Refer to App.C for detailed results.

Document Clustering Clustering enables aggregation of texts with similar semantics (Fig.3 A.IV) so that annotators can focus more on a certain class of concepts and thus increase productivity.

3.2. Dashboard and Project Creation

CollabKG is designed to enhance power and user-friendliness. CollabKG offers dashboard with several features such as real-time (E)KG visualization, double-checking, statistic and downloading. See App.G for details. The project creation process of CollabKG is clear, and the operation is user-friendly and well-guided. During project creation, users can customize the task scheme, pre-processing, uploading pre-annotation files, and more (App.H).

4. Human Evaluation

4.1. Evaluation Method

We designed a series of comparison experiments. We randomly sampled 50 instances from the conllpp (Wang et al., 2019b), NYT11-HRL (Takanobu et al., 2019), and ACE05 (Christopher Walker and Maeda, 2006) English datasets for the NER, RE, and EE tasks, respectively. For each task, we hired ten humans and randomly divided them equally into the control group (without the automatic labeling module) and the experimental group (with the automatic labeling module). During the annotation, we record the annotation time, perform performance evaluation, as well as calculate the intra-group variance (Pang et al., 2020) to assess effectiveness, efficiency, and variance.

Participants: All participants were senior students (female 4-6 per task) from universities who had passed the qualifications of English language ability (*e.g.*, IELTS). Every participant was paid a wage of $14.17/h and signed an informed consent form (App.F).

Metrics: Following the previous work (Wei et al., 2023), we adopt Micro F1 as the metric. For NER, the predicted entity is correct only if its whole span and type are correct. For RE, an extracted triple is considered as correct if the whole span of both head and tail entities, as well as the relation, are all correct. For EE (Lin et al., 2020), an argument is correctly identified only if its whole span, role label, and event type match the ground truth (Arg-C). A trigger is correctly identified only if its whole span and event type match the golden trigger (Trig-C).

We calculate the intra-group variance as follows: $var = 1 - \frac{1}{|D| \cdot C_{|G|}^{2}} \sum \sum_{i<j \in G} get\_f1\_list(g_i, g_j)$, where G denotes the experimental or control group, g denotes the sequence of annotation results, C denotes the combinatorial number, and D denotes the dataset. Moreover, get_f1_list denotes the micro F1 between the labeling results of the two participants (one for golden and the other for prediction). The more similar the labeling of the two participants, the larger the F1 value.

4.2. Results

The results are shown in Tab.3. CollabKG (**Auto**) denotes pure automatic labeling. CollabKG (**Human**) denotes humans using CollabKG without assistance from automatic labeling. CollabKG denotes complete human-machine cooperation. Results show that CollabKG significantly improves annotation quality, efficiency, and stability.

On NER, RE, and EE tasks, the average improvement w.r.t. effectiveness is 18.75%. The average improvement w.r.t. variance is 0.315. Time is most affected by external or human factors, but nonetheless, speedups of 1.8%, 3.9%, and 2.1% were achieved on NER, RE, and EE tasks, respectively. In addition, compared to BRAT, the annotation speed of CollabKG exceeds 4.6%, 27.2%, and 32.6% for NER, RE and EE tasks, respectively. It is worth noting that the effectiveness of BRAT is the same as CollabKG (Human) since both are only manually annotated. Thus CollabKG also outperforms BRAT by 18.75% on average.

5. Conclusion

We present CollabKG an open-source IE annotation tool for KG and EKG construction. Ultimately, we conducted an extensive human evaluation to quantitatively demonstrate that CollabKG can significantly improve annotation quality, efficiency, and stability as well as qualitative comparisons with other existing open-source tools.

6. Acknowledgements

The work is supported by the Talent Fund of Beijing Jiaotong University (2023XKRC006), CHN RAILWAY Grant Number (L2023G013), the National Key R&D Program of China (2020AAA0108001), and the National Nature Science Foundation of China (No. 61976016, 61976015, and 61876198).

7. Ethics Statement

All participants in this study will be fully informed of the nature of the study and will be required to provide informed consent prior to participation. All personal data will be kept confidential and anonymous.

8. Bibliographical References

Giuseppe Abrami, Alexander Mehler, Andy Lücking, Elias Rieb, and Philipp Helfrich. 2019. Textannotator: A flexible framework for semantic annotations.

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Chinatsu Aone and Mila Ramos-Santacruz. 2000. REES: A large-scale relation and event extraction system. In *Sixth Applied Natural Language Processing Conference*, pages 76–83, Seattle, Washington, USA. Association for Computational Linguistics.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers*, pages 2503–2514.

Nicholas Bien, Pranav Rajpurkar, Robyn L Ball, Jeremy Irvin, Allison Park, Erik Jones, Michael Bereket, Bhavik N Patel, Kristen W Yeom, Katie Shpanskaya, et al. 2018. Deep-learning-assisted diagnosis for knee magnetic resonance imaging: development and retrospective validation of mrnet. *PLoS medicine*, 15(11):e1002699.

Tyler Bikaun, Michael Stewart, and Wei Liu. 2022. Quickgraph: A rapid annotation tool for knowledge graph extraction from technical text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

BSI. 1973a. *Natural Fibre Twines*, 3rd edition. British Standards Institution, London. BS 2570.

BSI. 1973b. Natural fibre twines. BS 2570, British Standards Institution, London. 3rd. edn.

A. Castor and L. E. Pollux. 1992. The use of user modelling to guide inference and learning. *Applied Intelligence*, 2(1):37–53.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. 2020. Knowledge graph completion: A review. *Ieee Access*, 8:192435–192456.

J.L. Chercheur. 1994. *Case-Based Reasoning*, 2nd edition. Morgan Kaufman Publishers, San Mateo, CA.

N. Chomsky. 1973. Conditions on transformations. In *A festschrift for Morris Halle*, New York. Holt, Rinehart & Winston.

James W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

Umberto Eco. 1990. *The Limits of Interpretation*. Indian University Press.

Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo. 2022. What is event knowledge graph: a survey. *IEEE Transactions on Knowledge and Data Engineering*.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 169–174, Hong Kong, China. Association for Computational Linguistics.

Paul Gerhard Hoel. 1971a. *Elementary Statistics*, 3rd edition. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester. ISBN 0 471 40300.

Paul Gerhard Hoel. 1971b. *Elementary Statistics*, 3rd edition, Wiley series in probability and mathematical statistics, pages 19–33. Wiley, New York, Chichester. ISBN 0 471 40300.

Rezarta Islamaj, Dongseop Kwon, Sun Kim, and Zhiyong Lu. 2020. TeamTat: a collaborative text annotation tool. *Nucleic Acids Research*, 48(W1):W5–W11.

Otto Jespersen. 1922. *Language: Its Nature, Development, and Origin*. Allen and Unwin.

Zhuoran Jin, Yubo Chen, Dianbo Sui, Chenhao Wang, Zhipeng Xue, and Jun Zhao. 2021. Cogie: An information extraction toolkit for bridging texts and cognet. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 92–98.

Mayank Kejriwal. 2022. Knowledge graphs: Constructing, completing, and effectively applying knowledge graphs in tourism. In *Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications*, pages 423–449. Springer.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Jonathan K. Kummerfeld. 2019. SLATE: A super-lightweight annotation tool for experts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12, Florence, Italy. Association for Computational Linguistics.

Linfeng Li, Peng Wang, Jun Yan, Yao Wang, Simin Li, Jinpeng Jiang, Zhe Sun, Buzhou Tang, Tsung-Hui Chang, Shenghui Wang, et al. 2020. Real-world data medical knowledge graph: construction and applications. *Artificial intelligence in medicine*, 103:101817.

Yanzeng Li, Bowen Yu, Li Quangang, and Tingwen Liu. 2021. FITAnnotator: A flexible and intelligent text annotation system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 35–41, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Mariana Neves and Jurica Ševa. 2019. An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*, 22(1):146–163.

Minh-Quoc Nghiem and Sophia Ananiadou. 2018. APLenty: annotation tool for creating high-quality datasets using active and proactive learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 108–113, Brussels, Belgium. Association for Computational Linguistics.

Bo Pang, Erik Nijkamp, Wenjuan Han, Linqi Zhou, Yixian Liu, and Kewei Tu. 2020. Towards holistic and automatic evaluation of open-domain dialogue generation. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 3619–3629, Online. Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Charles Joseph Singer, E. J. Holmyard, and A. R. Hall, editors. 1954–58. *A history of technology*. Oxford University Press, London. 5 vol.

Tarcísio Souza Costa, Simon Gottschalk, and Elena Demidova. 2020. Event-qa: A dataset for event-centric question answering over knowledge graphs. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 3157–3164.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Michael Stewart, Wei Liu, and Rachel Cardell-Oliver. 2019. Redcoat: A collaborative annotation tool for hierarchical entity typing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 193–198, Hong Kong, China. Association for Computational Linguistics.

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753, Istanbul, Turkey. European Language Resource Association (ELRA).

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

S. Superman, B. Batman, C. Catwoman, and S. Spiderman. 2000. *Superheroes experiences with books*, 20th edition. The Phantom Editors Associates, Gotham City.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7072–7079.

Mingwei Tang, Cui Su, Haihua Chen, Jingye Qu, and Junhua Ding. 2020. Salkg: A semantic annotation system for building a high-quality legal knowledge graph. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2153–2159.

Marco A. Valenzuela-Escárcega, Gus Hahn-Powell, Mihai Surdeanu, and Thomas Hicks. 2015. A domain-independent rule-based framework for event extraction. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 127–132, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019a. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019b. CrossWeigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2019. A novel cascade binary tagging framework for relational triple extraction. *arXiv preprint arXiv:1909.03227*.

Haoyang Wen, Ying Lin, Tuan Lai, Xiaoman Pan, Sha Li, Xudong Lin, Ben Zhou, Manling Li, Haoyu Wang, Hongming Zhang, Xiaodong Yu, Alexander Dong, Zhenhailong Wang, Yi Fung, Piyush Mishra, Qing Lyu, Dídac Surís, Brian Chen, Susan Windisch Brown, Martha Palmer, Chris

Callison-Burch, Carl Vondrick, Jiawei Han, Dan Roth, Shih-Fu Chang, and Heng Ji. 2021. RESIN: A dockerized schema-guided cross-document cross-lingual cross-media information extraction and event tracking system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 133–143, Online. Association for Computational Linguistics.

Chengbiao Yang, Weizhuo Li, Xiaoping Zhang, Runshun Zhang, and Guilin Qi. 2020. A temporal semantic search system for traditional chinese medicine based on temporal knowledge graphs. In *Semantic Technology: 9th Joint International Conference, JIST 2019, Hangzhou, China, November 25–27, 2019, Revised Selected Papers 9*, pages 13–20. Springer.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. WebAnno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, Sofia, Bulgaria. Association for Computational Linguistics.

Ningyu Zhang, Xin Xu, Liankuan Tao, Haiyang Yu, Hongbin Ye, Shuofei Qiao, Xin Xie, Xiang Chen, Zhoubo Li, and Lei Li. 2022. DeepKE: A deep learning based knowledge extraction toolkit for knowledge base population. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 98–108, Abu Dhabi, UAE. Association for Computational Linguistics.

Xuejiao Zhao, Huanhuan Chen, Zhenchang Xing, and Chunyan Miao. 2021. Brain-inspired search engine assistant based on knowledge graph. *IEEE Transactions on Neural Networks and Learning Systems*.

Xiaohan Zou. 2020. A survey on application of knowledge graph. In *Journal of Physics: Conference Series*, volume 1487, page 012016. IOP Publishing.

9. Language Resource References

Christopher Walker, Stephanie Strassel, Julie Medero and Kazuaki Maeda. 2006. *ACE05*. Linguistic Data Consortium, ISLRN 458-031-085-383-4. PID https://catalog.ldc.upenn.edu/LDC2006T06.

A. System Architecture

As shown in Fig. 4, we use modern full-stack framework MERN [3], Docker and Python to build `CollabKG`. It consists of four components wrapped in a Docker container, namely web client, server, NLP server, and database. The NLP server deploys ChatIE to obtain automatic annotation results. In addition, the database is crucial, storing and managing information such as projects, texts, users, *etc*. This is achieved by maintaining three collections including `Project`, `Text`, and `User`. `Project` stores the task details of the project, whether to perform the model update, semantic clustering, preprocessing, *etc*. `Text` manages samples, markups, *etc*. `User` stores the information of users like name and password.

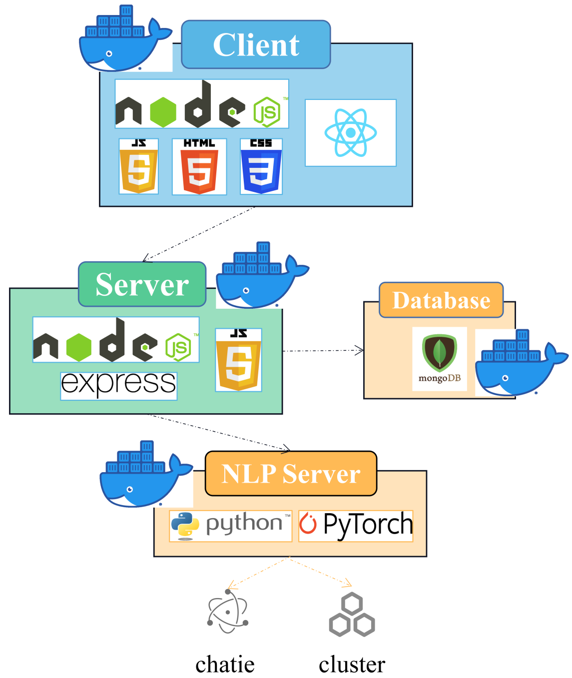

Figure 4: System Architecture.

B. Related Works

There are many existing open-source IE (*i.e.*, NER, RE, and EE) annotation tools for KG or EKG construction. We will describe them from several perspectives such as applicable tasks and annotation styles.

From the perspective of the three applicable tasks, FLAIR (Akbik et al., 2019), OpenNRE (Han et al., 2019), ODIN (Valenzuela-Escárcega et al., 2015), FitAnnotator (Li et al., 2021), WebAnno (Yimam et al., 2013), Redcoat (Stewart et al., 2019) and APLenty (Nghiem and Ananiadou, 2018)

[3]MongoDB-Express-React-Node `https://www.mongodb.com/mern-stack`

only focus on a single task. DeepKE (Zhang et al., 2022), Quickgraph (Bikaun et al., 2022), INCEpTION (Klie et al., 2018), TeamTat (Islamaj et al., 2020), TextAnnotator (Abrami et al., 2019), REES (Aone and Ramos-Santacruz, 2000) and SALKG (Tang et al., 2020) support two tasks. CogIE (Jin et al., 2021), BRAT (Stenetorp et al., 2012), SLATE (Kummerfeld, 2019) and RESIN (Wen et al., 2021) support all three tasks. However, CogIE and RESIN only support automatic labeling and require training, which is not suitable for low-resource scenarios where data is insufficient for training from scratch. BRAT and SLATE only support manual labeling. Moreover, BRAT sometimes is criticized for its difficulties in deployment (Neves and Ševa, 2019). SLATE is a command-line-based tool so it is not user-friendly.

From the perspective of the annotation styles (*i.e.*, automatic and manual labeling). INCEpTION, TextAnnotator, FitAnnotator, and APLenty support two annotation styles immediately. However, INCEpTION and TextAnnotator don't support the learnability function (namely, self-renewal). Although FitAnnotaor and APLenty utilize active learning to support two-way interaction, this is inflexible, not real-time, and requires a training process. In addition, as mentioned earlier, they only support a single task.

C. Detailed Results

To note, the standard labeling process includes multiple rounds of labeling, including validating and refining, iterating and improving. Validating and refining the annotations in the knowledge graph aims to ensure that the labels accurately represent the domain knowledge by double-checking. Iterating and improving denote continuously iterating and improving the knowledge graph based on feedback from multiple turns. In this work, all results are obtained in one round of annotation, so the numbers may seem to be low.

NER The results are presented in Tab. 4. The experimental group consisted of No. 6 to 10, while the control group consisted of No. 1 to 5. Before calculating the metrics we eliminated No.3 and No.8 because of their poor annotating quality.

RE The results are presented in Tab. 5. The experimental group consisted of No. 6 to 10. Before calculating the metrics we eliminated No.5 and No.6 because of their long annotation time or poor annotation quality. It is worth noting that since NYT11-HRL is obtained by remote supervision, the gold annotation does not cover all entities and relationships (Wei et al., 2019). Therefore, we re-examined and relabeled the 50 samples as the gold label.

EE The results are presented in Tab. 6. The experimental group consisted of No. 6 to 10. No. 5 participant was absent for personal reasons, so to align with the control group, we eliminated the results of No. 6 participant (poor annotation quality). It is worth noting that because there are so many tags in ACE05 (namely, 33), annotating them all would be too tricky. Therefore, we narrowed the tag range to 12 (within 50 samples).

	P	R	F1	Time
P1	56.4	58.5	57.4	00:40:24
P2	70.3	60.4	65.0	00:42:21
P3	48.1	58.5	52.8	00:41:03
P4	71.4	66.0	68.6	00:42:07
P5	58.8	63.2	60.9	00:40:56
P6	79.4	80.2	79.8	00:40:31
P7	87.8	81.1	84.3	00:40:32
P8	73.3	72.6	73.0	00:41:17
P9	83.8	78.3	81.0	00:40:09
P10	75.9	77.4	76.6	00:41:40

Table 4: Human evaluation results on NER. **P** denotes participant.

	P	R	F1	Time
P1	57.0	47.6	51.9	01:29:42
P2	50.0	50.5	50.2	01:32:39
P3	35.3	39.8	37.4	01:23:44
P4	50.0	38.8	43.7	01:23:01
P5	47.9	33.0	39.1	02:15:39
P6	55.4	60.2	57.7	01:21:40
P7	68.1	62.1	65.0	01:14:04
P8	80.9	73.8	77.2	01:21:28
P9	75.3	73.8	74.5	01:23:56
P10	58.8	77.7	67.0	01:36:39

Table 5: Human evaluation results on RE. **P** denotes participant.

D. Implementation of Unification

`CollabKG` is designed to unify three IE tasks including NER, RE, and EE, and integrate KG and EKG construction. To unify these tasks, we first observe their schemes and summarize the transformation rules among them. Then we uniformly integrate these three schemes utilizing these transformation

	P	R	F1	Time
P1	37.3/69.2	25.5/44.3	30.3/54.0	01:10:44
P2	43.4/75.0	30.0/49.2	35.5/59.4	01:27:25
P3	26.5/65.2	23.6/49.2	25.0/56.1	01:05:34
P4	36.6/72.5	30.9/47.6	33.5/57.4	01:07:34
P5	-	-	-	-
P6	37.7/84.6	41.8/72.1	39.7/77.9	01:04:51
P7	47.4/86.0	40.9/70.5	43.9/77.5	01:02:48
P8	44.9/82.7	48.2/70.5	46.5/76.1	01:33:12
P9	37.2/78.6	46.4/72.1	41.3/75.2	01:07:02
P10	44.0/86.3	46.4/72.1	45.1/78.6	01:02:21

Table 6: Human evaluation results on EE. The left and right numbers in Column **F1** represent Arg-C and Trig-C, respectively. **P** denotes participant.

rules and design an annotation format to fit the unified scheme.

Transformation Rule NER aims to find entities with specific preset entity types from the given text. For example, given the preset type list `[PER, LOC, ORG, MISC]` and the sentence "*James worked for Google in Tokyo, the capital of Japan.*", these entities should be recognized are: `PER:James, ORG:Google, LOC:Tokyo, LOC:Japan`.

RE aims to find pairs of entities, predict the relations between them and form triples. For instance, given the sentence "*Mr.Johnson retired before the 2005 season and briefly worked as a football analyst for WBZ-TV in Boston.*", this triple is `(Person:Mr.Johnson, person-company, Organization:WBZ-TV)`. The first term in the triple is called `Subject`, the middle term is `Relation`, and the last term is called `Object`.

EE plays an important role in EKG construction. It aims to identify event types, triggers, arguments involved, and the corresponding roles. For instance, given the sentence "*Yesterday Bob and his wife got married in Beijing.*", we regard `Life:Marry` as event type, "*married*" as the trigger, "*Bob and his wife*" as `Person`, "*Yesterday*" as `Time` and "*Beijing*" as `Place`.

NER	E-Type:E $\longrightarrow$(E-Type:E, _, _)
RE	(S-Type:S, R, O-Type:O)
EE	{E-Type:T, R_1:A_1, ..., R_n:A_n } $\longrightarrow$ {(E-Type:T, R_n, _:A_n)}

Table 7: Unification of three tasks. $\longrightarrow$ represents the transformation. For NER, `E` denotes `Entity`. For RE, `S`, `R`, `O` represents `Subject`, `Relation`, `Object`, respectively. For EE, `E`, `T`, `R`, `A` represents `Event`, `Trigger`, `Role`, `Argument`, respectively while `n` denotes the number of arguments. `_Type` denotes the type. _ represents a pseudo token.

The schemes of the three tasks can be summarized in Tab. 7. Through the above descriptions, transformation rules among tasks are observed as follows. RE scheme remains unchanged. The NER scheme can be attributed to the entity part of the RE scheme. For EE, if we regard `Trigger` as `Subject`, `Argument` as `Object`, and `Role` as `Relation`, we find it easy to decompose the EE structure into a combination of multiple RE triples. With the transformation rules, we leverage RE as the center, transform NER and EE schemes and thus unify the three IE schemes.

Integration We uniformly model the scheme of the three IE tasks and integrate the schemes into a series of triples `(S-Type:S,R,O-Type:O)`. To design an annotation format to fit the unified scheme, we design `Entity` annotation mode and `Relation` annotation mode. `Entity` mode considers entities in NER, subjects, and objects in RE, arguments, and triggers in EE. Similarly, `Relation` mode considers relations in RE and roles in EE. It is worth noting that in this way, KG and EKG construction are also integrated besides of RE, NER and EE.

E. Automatic Labeling

NER	[entity type 1, ..., entity type n]
RE	{relation type 1: [subject1, object1],...}
EE	{event type 1: [argument role 1, ...],...}

Table 8: Type list format for three IE tasks.

We have adopted ChatIE (Wei et al., 2023) as our approach for zero-shot information extraction, based on ChatGPT. ChatIE has shown impressive performance, even surpassing some full-shot models across various datasets. Its flexibility for customization is also a notable advantage because the type list allows customization (refer to Tab. 8).

Therefore, we have adopted ChatIE as the backbone of our automatic labeling module, with a slight modification that adds the trigger-related prompt template so that it can extract trigger words according to the event type. This allows for the extraction of trigger words based on the event type. For example, the following prompt template is used: `When the event type of the given sentence above is "<event-type>", please recognize the corresponding trigger. The trigger is the word or phrase that most clearly expresses event occurrences.\nOnly answer the trigger, no extra word. The trigger is:.`

F. Human Evaluation Procedure

All participants were gathered in a conference room and asked to sign a consent form before being introduced to the task and annotation criteria. They were then given model accounts to begin annotating without the use of external tools. Participants in the experimental group received ChatIE assistance. Once they finished annotating, they notified us using a communication tool, and we recorded the time spent on annotation. We evaluated metrics and conducted statistic analysis using a pre-written script.

G. Display

Our toolkit offers a range of features to enhance the user-friendly display. Firstly, it displays annotation progress and counts the results across multiple dimensions, such as entity, relation, and triples (see Fig. 5). Secondly, users can view a KG or EKG and filter the results (see Fig. 6 and 7). Thirdly, the tool provides a double-checking function for each text (see Fig. 8). Finally, our tool supports the filtering and exporting functions (see Fig. 9). Filter function includes saving, loading, quality filtering (accepted or suggested), and so on.

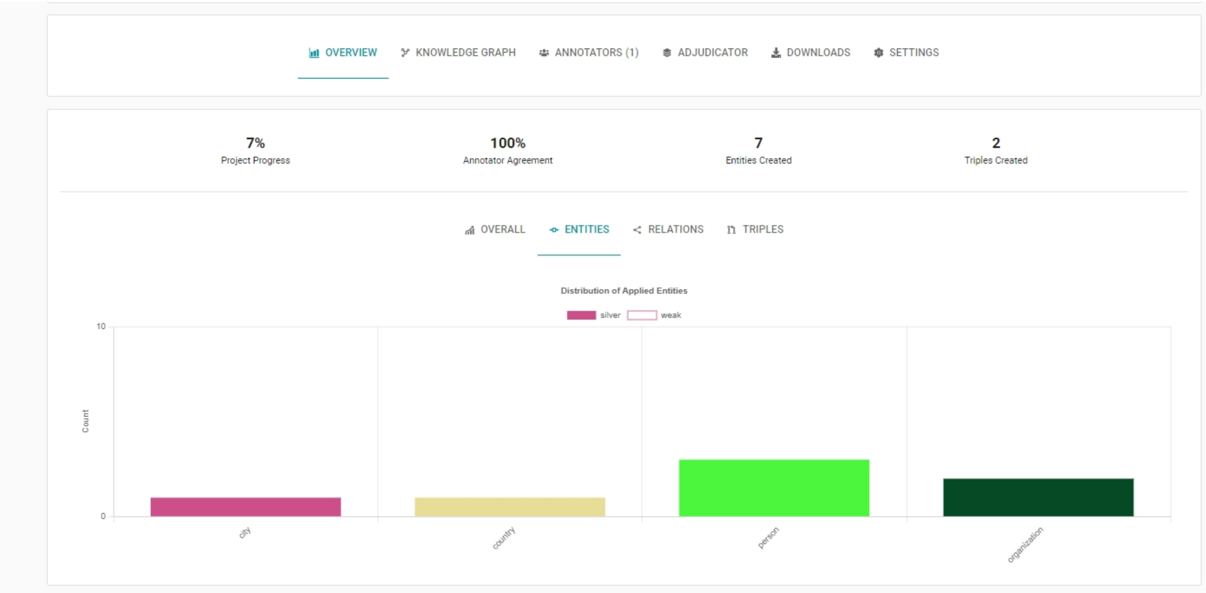

Figure 5: Overview of Dashboard.

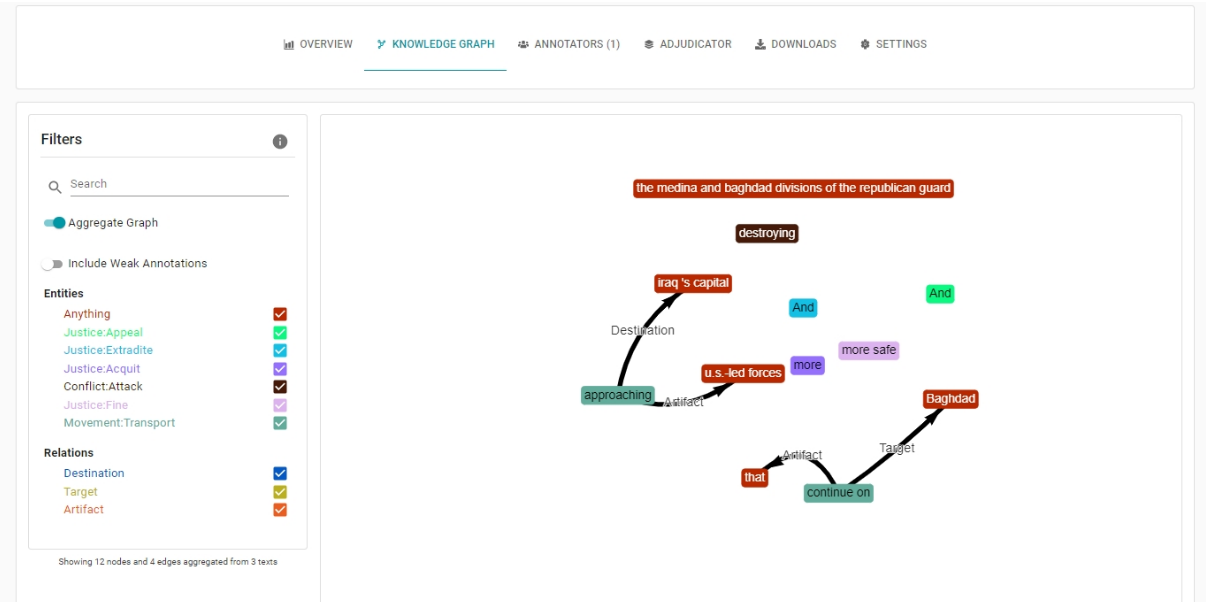

Figure 6: EKG display.

H. Project Creation Process

The project is created for each person on a dataset. The creation process can be divided into the following steps:

- *Configuration Setup:* Set up the details of the project: the name, the description, the configure of multi-task (NER, RE, EE), whether to perform `model update` (see Sec.2.3) and `text/document`

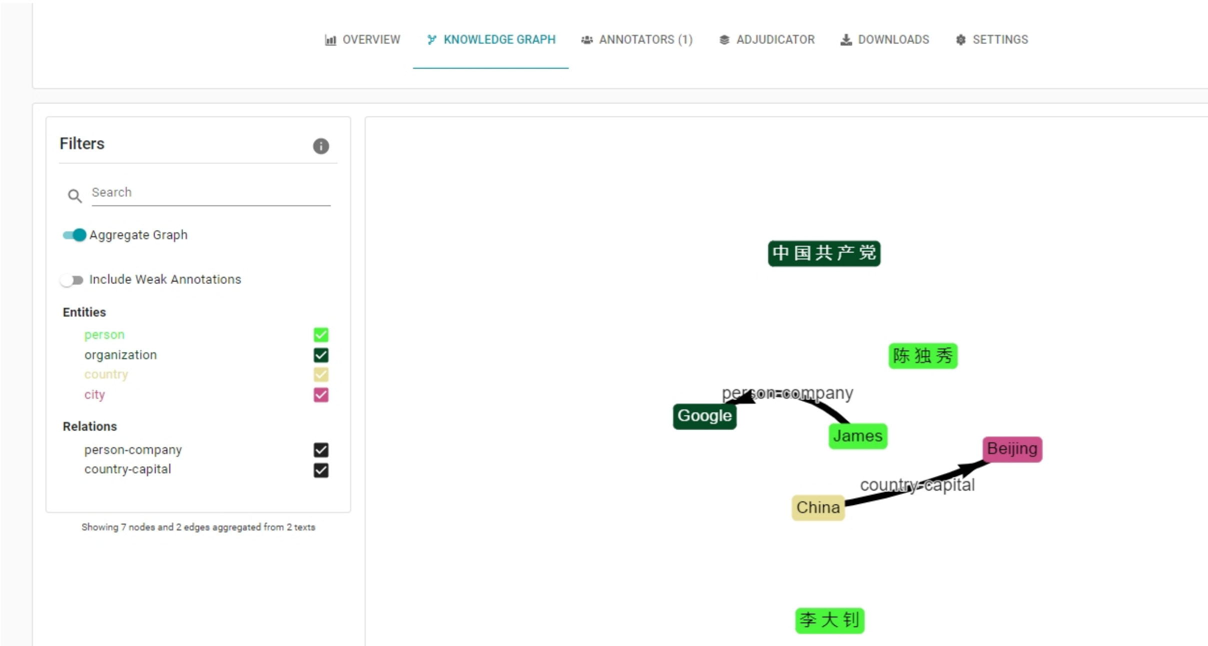

Figure 7: KG display.

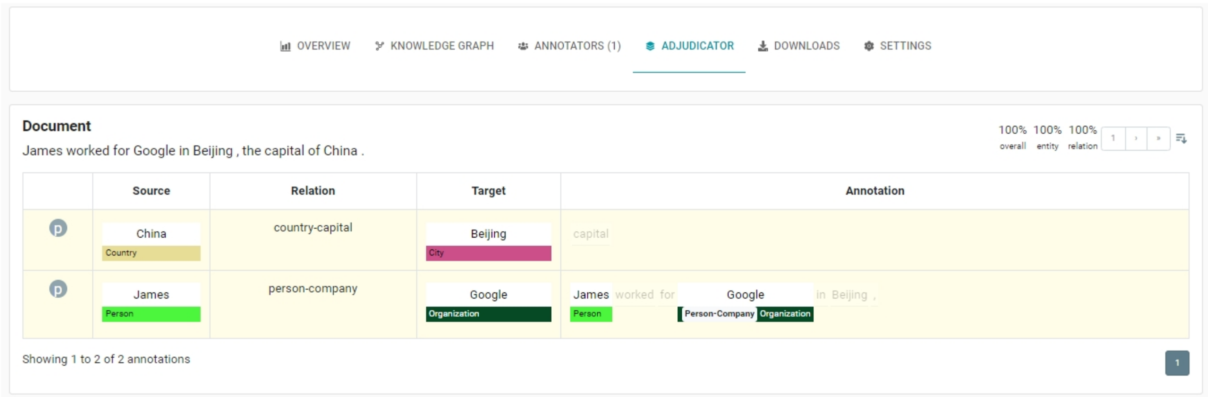

Figure 8: Double-checking display.

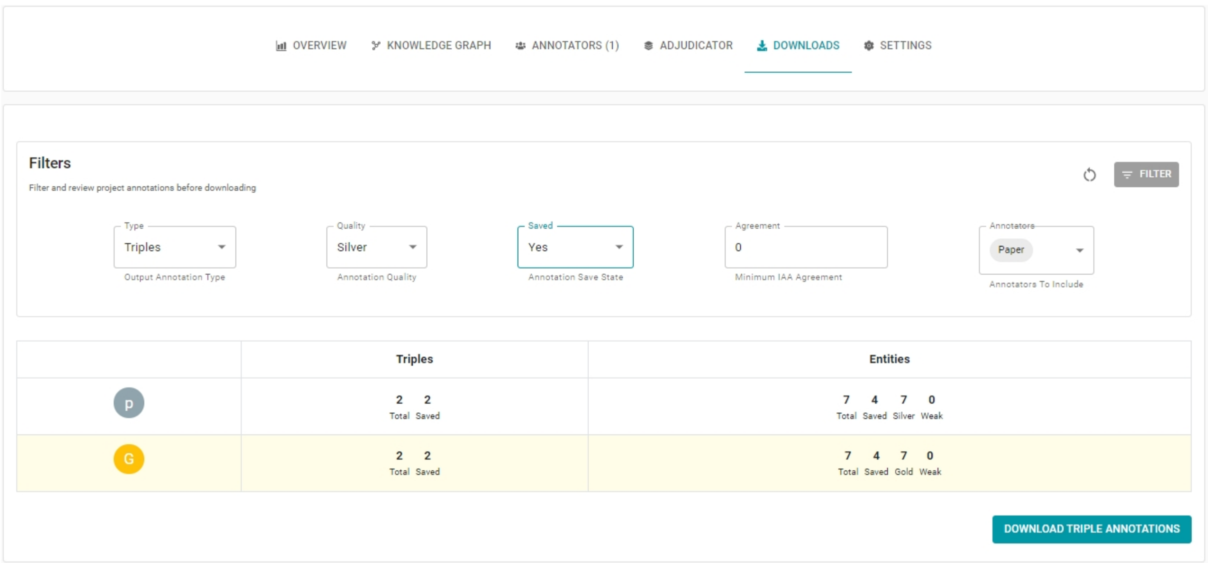

Figure 9: Download display.

`clustering` features (Fig. 10). Clustering enables aggregation of texts with similar semantics (Fig. 3 A.IV) so that annotators can focus more on a certain class of concepts and thus increase productivity. Our toolkit implements cohesive clustering by encoding documents with SBERT (Reimers and Gurevych, 2019) sentence embeddings.

- *Uploading Data:* It supports keyboard input and uploading files (Fig. 11).
- *Pre-processing:* Pre-processing function includes character casing, specified-character removal, and text de-duplication (Fig. 12).
- *Scheme Setup:* Build an ontology/scheme for the current task. Users can choose from the pre-set ontology or **customize** their own scheme. For RE, the `RELATION TYPES` format is `relation@[subject, object]`, where subject/object refers to head/tail entity type in triples (Fig. 13). For EE, the `ENTITY TYPES` is filled with a pseudo token (namely _), and the `RELATION TYPES` format is `event-type@[argument role 1, argument role 2, ...]` (Fig. 14). It is worth noting that `CollabKG` will complete the processing to convert `role` to `relation` and `event-type` to `entity-type` on the back-end. Unlike other IE annotation tools, our tool supports **`hierarchical labels`** (Fig. 15) and **`relation constraint`** (Fig. 16). The hierarchical labels facilitate the management of complex schemes. The relation constraint is a predetermination that a relationship can only occur between certain entity types. Consequently, this feature can narrow down the annotator's attention and improve the productivity and consistency of the annotators.
- *Preannotation:* Users can choose to upload pre-annotated entities and relations of the current corpus. This can reduce annotation effort by pre-applying tags based on external resources such as gazetteers (Fig. 17).
- *Review:* Summarize the current project. Hence users can check and make changes (Fig. 18).

Finally, when the user clicks the `CREATE` button, the project creation process is completed and will appear in the panel (Fig. 19).

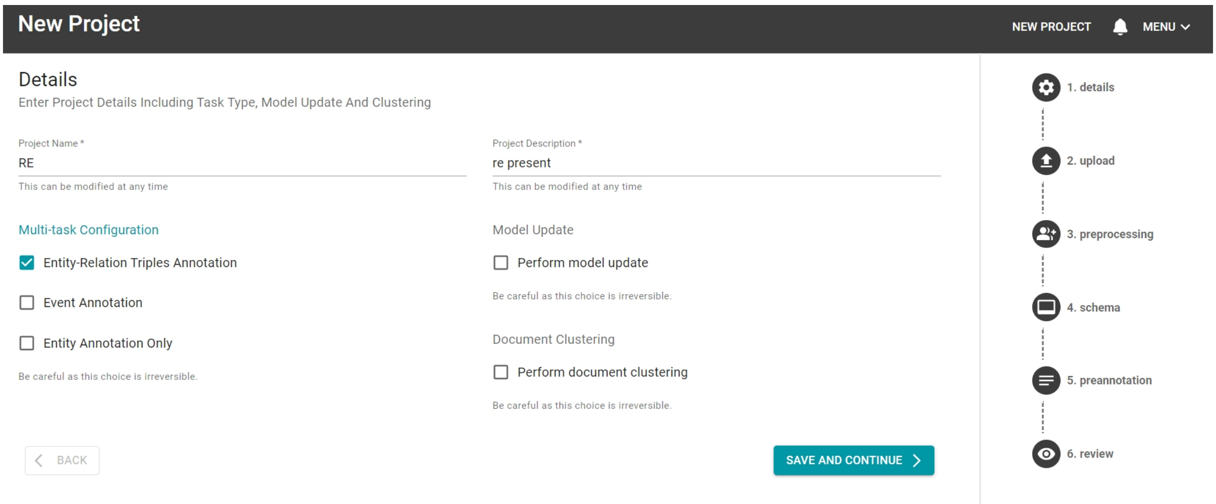

Figure 10: Detail of project creation.

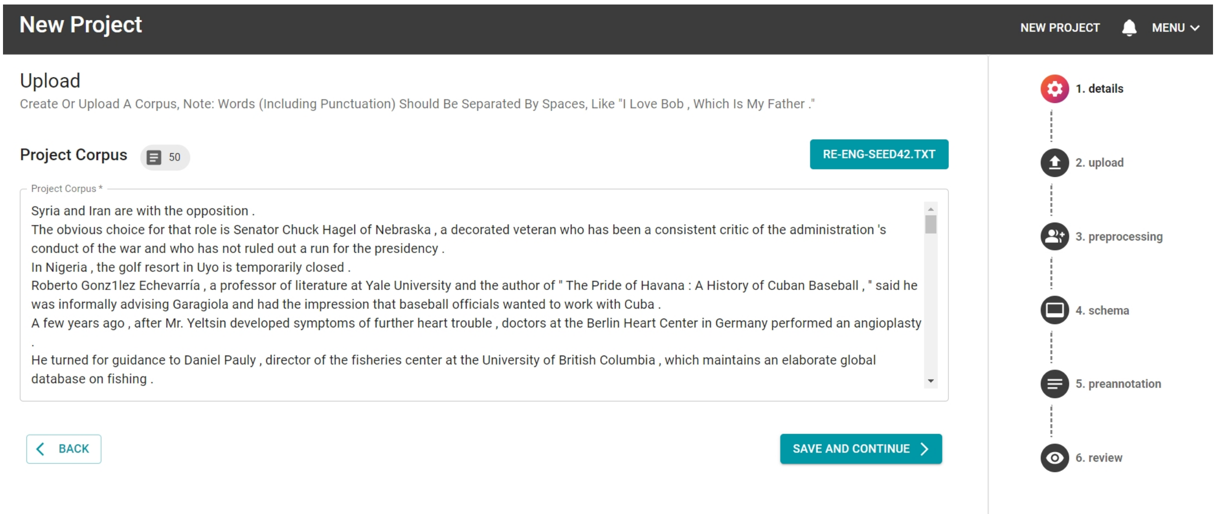

Figure 11: Uploading of project creation.

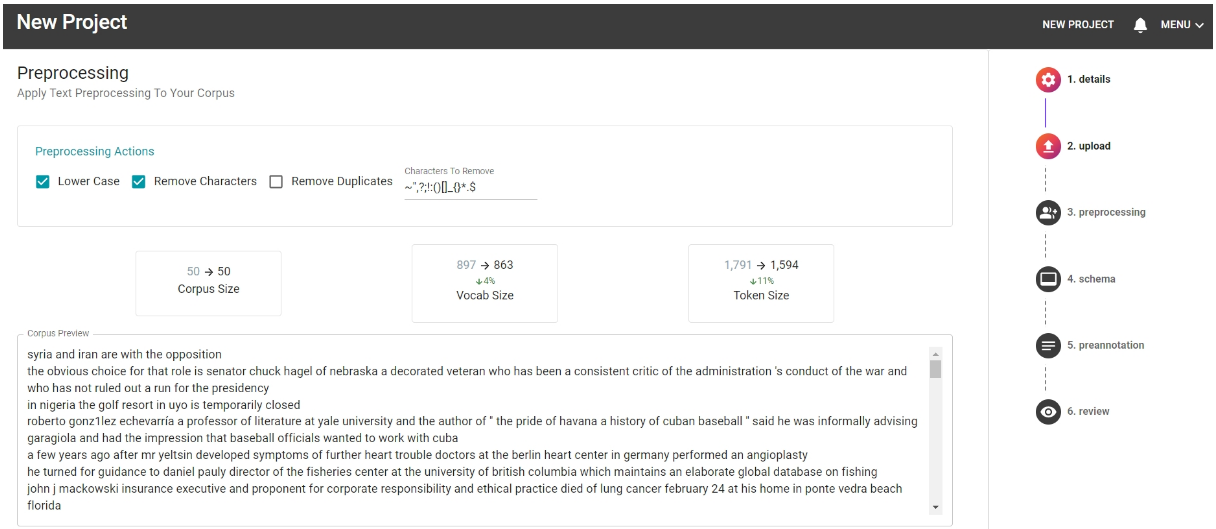

Figure 12: Preprocessing of project creation.

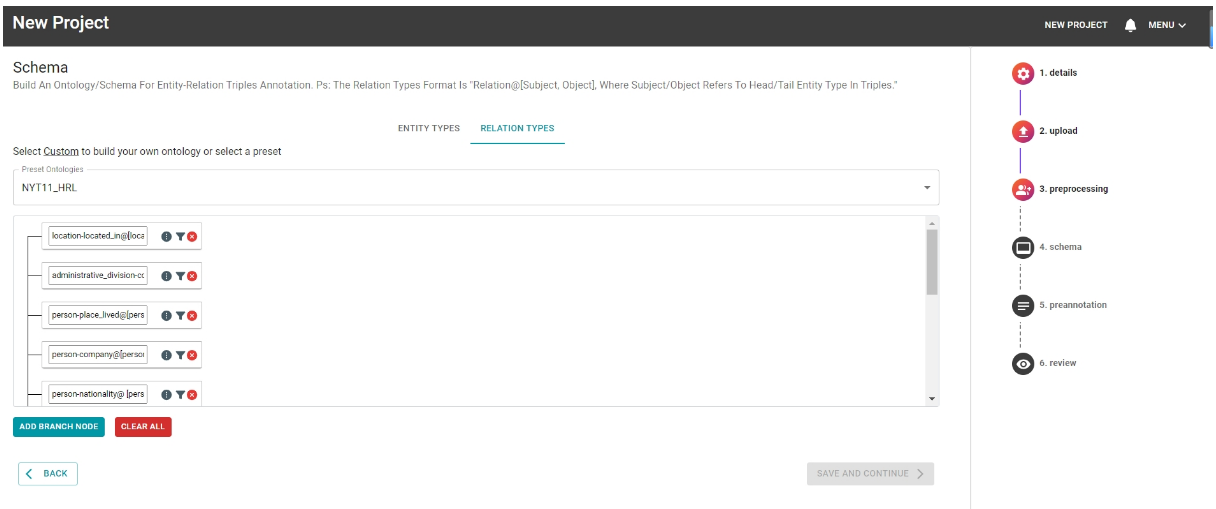

Figure 13: Scheme setup for RE of project creation.

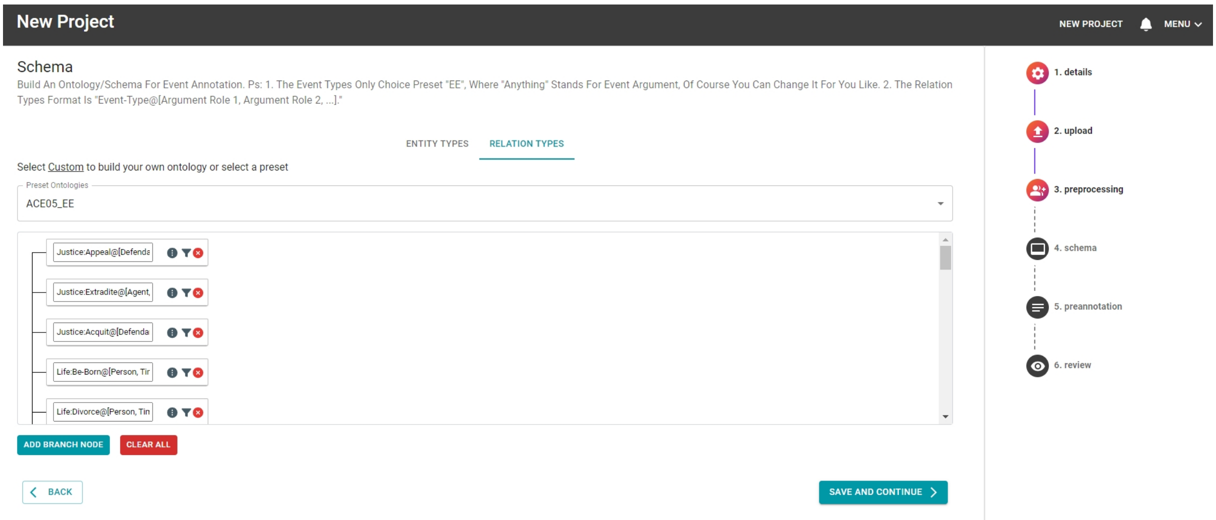

Figure 14: Scheme setup for EE of project creation.

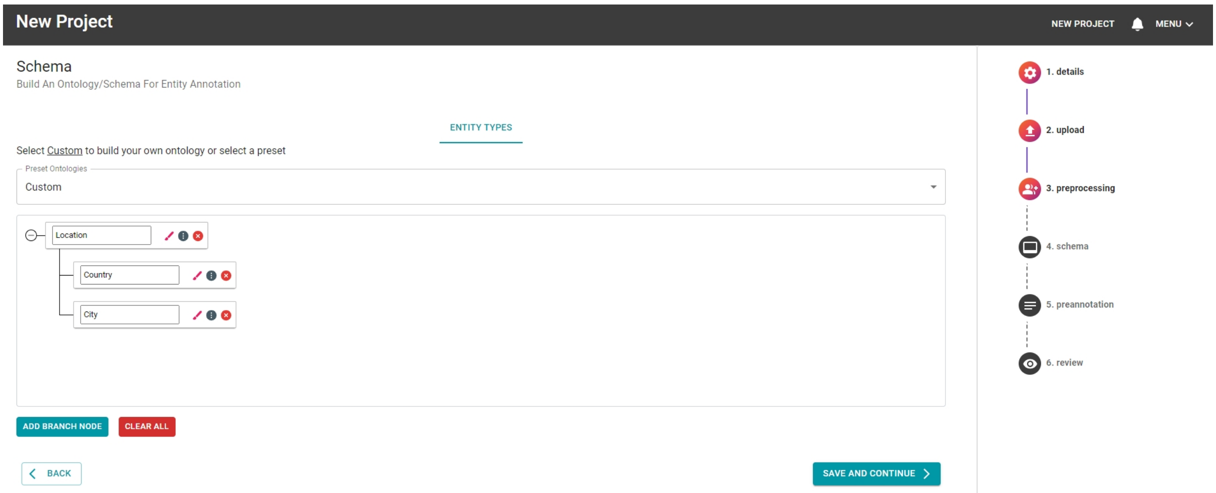

Figure 15: Hierarchical labels of project creation.

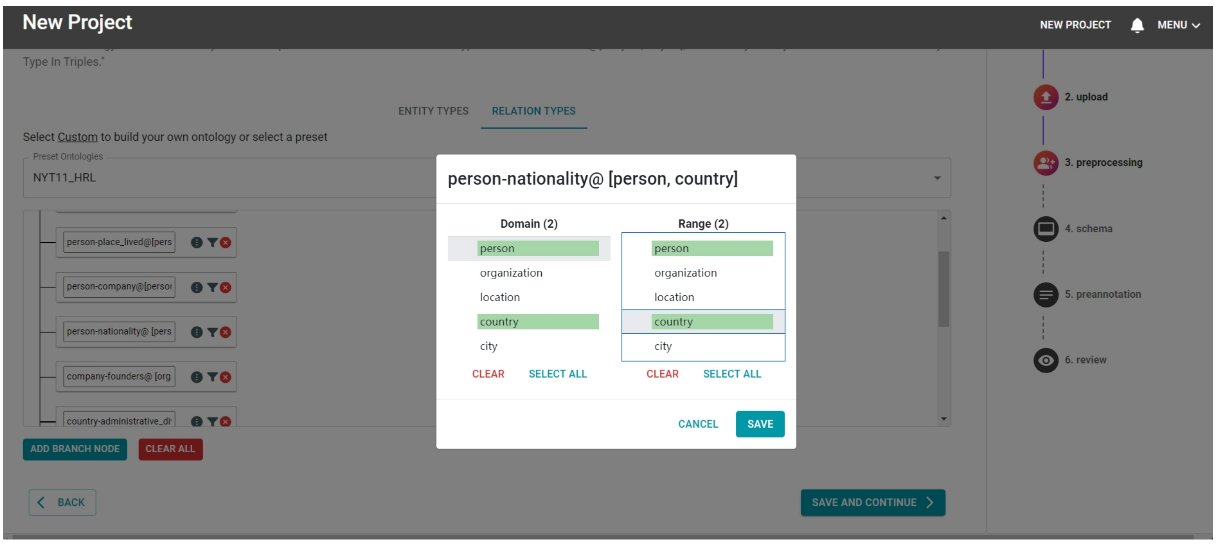

Figure 16: Relation constraint of project creation.

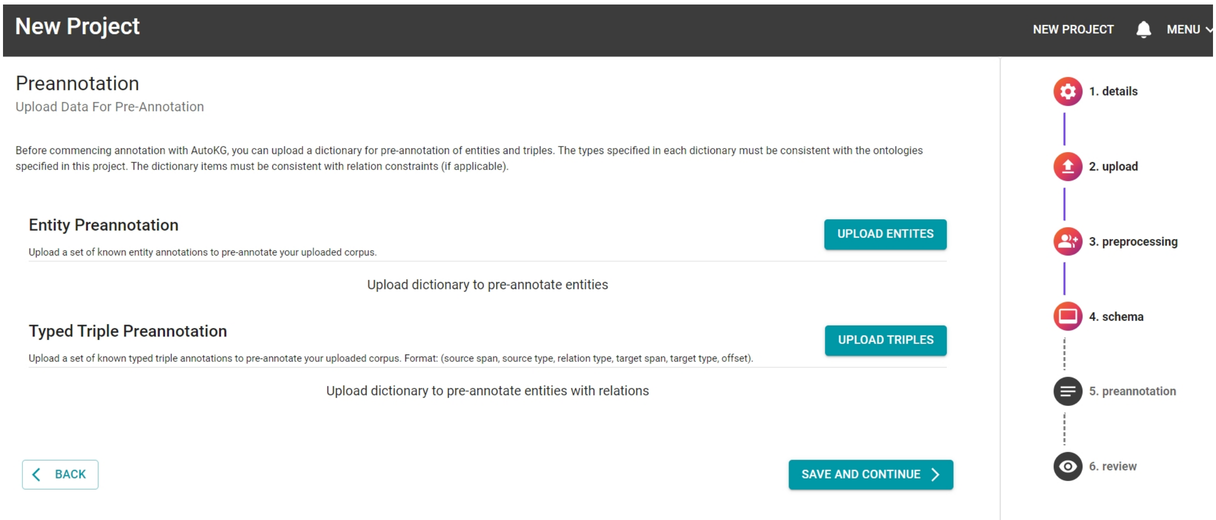

Figure 17: Preannotation of project creation.

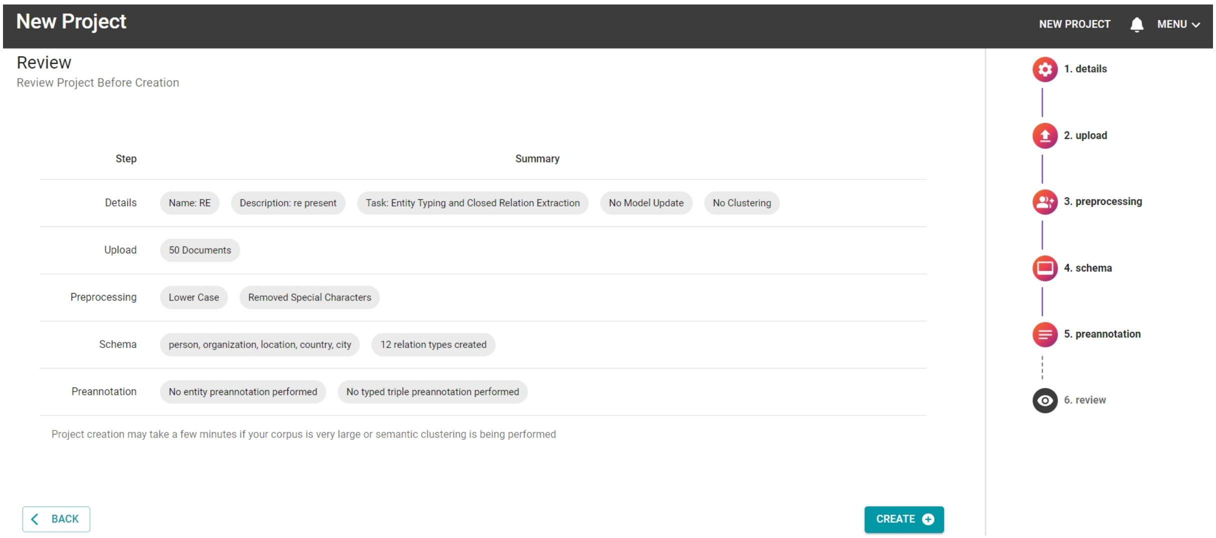

Figure 18: Review of project creation.

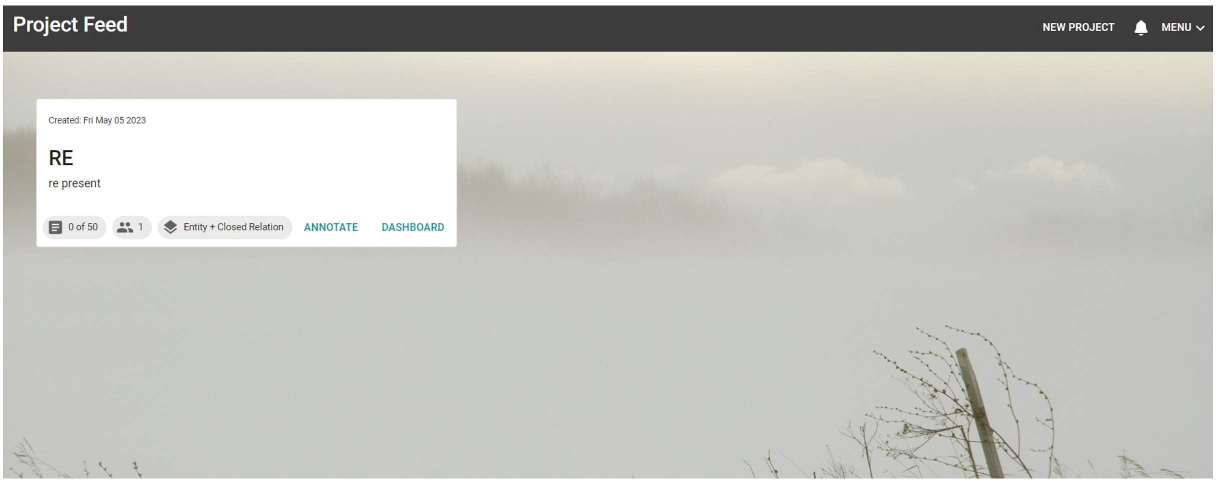

Figure 19: Feed of projects.