TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 11/16 pp589–598
DOI: 10.26599/TST.2020.9010060
Volume 27, Number 3, June 2022

CDCAT: A Multi-Language Cross-Document Entity and Event Coreference Annotation Tool

Yang Xu, Boming Xia, Yueliang Wan, Fan Zhang, Jiabo Xu, and Huansheng Ning*

Abstract: A tool for the manual annotation of cross-document entity and event coreferences that helps annotators to label mention coreference relations in text is essential for the annotation of coreference corpora. To the best of our knowledge, CROss-document Main Events and entities Recognition (CROMER) is the only open-source manual annotation tool available for cross-document entity and event coreferences. However, CROMER lacks multi-language support and extensibility. Moreover, to label cross-document mention coreference relations, CROMER requires the support of another intra-document coreference annotation tool known as Content Annotation Tool, which is now unavailable. To address these problems, we introduce Cross-Document Coreference Annotation Tool (CDCAT), a new multi-language open-source manual annotation tool for cross-document entity and event coreference, which can handle different input/output formats, preprocessing functions, languages, and annotation systems. Using this new tool, annotators can label a reference relation with only two mouse clicks. Best practice analyses reveal that annotators can reach an annotation speed of 0.025 coreference relations per second on a corpus with a coreference density of 0.076 coreference relations per word. As the first multi-language open-source cross-document entity and event coreference annotation tool, CDCAT can theoretically achieve higher annotation efficiency than CROMER.

Key words: event coreference; entity coreference; manual annotation tool; natural language processing

1 Introduction

Coreference resolution refers to the clustering of expressions that refer to the same entity or event in a text, whether within a single document or across a document collection[1]. Coreference resolution, as a means for extracting the deep semantics of natural language, is an essential Natural Language Processing (NLP) task and one of the most downstream NLP tasks. Coreference resolution is widely used in question answering system[2], natural language generation[3], and measure of academic articles similarity[4].

When identifying the coreference relations between mentions in texts, researchers mainly focus on two kinds of mentions: entities and events. An entity (or named entity) can be the name of a person, place, or institution, or a proper noun, to name a few[5]. An event is a specific occurrence involving participants[6]. Although different definitions of entity and event have been established by

• Yang Xu and Huansheng Ning are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, and also with the Beijing Engineering Research Center for Cyberspace Data Analysis and Applications, Beijing 100083, China. E-mail: xuyang_mail@sina.cn; ninghuansheng@ustb.edu.cn.
• Boming Xia and Fan Zhang are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China. E-mail: boming.xia@outlook.com; fanzhang@xs.ustb.edu.cn.
• Yueliang Wan is with Research Institute with Run Technologies Company, Ltd., Beijing 100192, China, and also with the Beijing Engineering Research Center for Cyberspace Data Analysis and Applications, Beijing 100083, China. E-mail: yueliang@bjrun.com.
• Jiabo Xu is with the School of Information Engineering, Xinjiang Institute of Engineering, Urumqi 830091, China. E-mail: xujiabo_math@aliyun.com.
* To whom correspondence should be addressed.
Manuscript received: 2020-11-17; revised: 2020-12-02; accepted: 2020-12-17

a range of open tasks, organizations, and conferences, such as the ACE[6, 7], TAC[8], and TimeML[9], there is no generally accepted definition for the term entity or event.

A mention usually refers to an object in the real world. In this paper, we refer to this object as an "*instance*" following the denotation used in Event Coreference Bank (ECB)+ corpus[10]. If two mentions refer to the same instance, they are co-referred. Tasks related to coreference relations in texts have attracted widespread attention since the Sixth Message Understanding Conference in 1995, and many coreference corpora have since become available.

However, in most of those corpora which are usually in English, the labeled relations are either entity or event coreferenced, and are merely intra-documents. Due to the high cost of manual annotation, only a few of these corpora are cross-documents that have both entity and event coreference relations labeled.

The most widely used cross-document entity and event coreference corpus is the ECB series (i.e., ECB[11], EECB[12], and ECB+[10]), in which the texts are all in English, and only events of interest are labeled. The lack of corpora in other languages limits related research in this field, resulting in the need for a manual annotation tool for creating corpora in different languages or a particular domain.

To the best of our knowledge, the CROss-document Main Events and entities Recognition tool (CROMER)[†] is the only open-source manual annotation tool for cross-document entity and event coreference[13]. However, CROMER supports only English and Italian corpora, and requires another intra-document coreference annotation tool known as Content Annotation Tool (CAT)[‡] to label cross-document mention coreferences, which is currently unavailable[14]. Furthermore, the annotation procedure requires that annotators read the corpus twice, which reduces efficiency.

To address the above challenges and problems, we present a new cross-document entity and event coreference manual annotation tool, Cross-Document Coreference Annotation Tool (CDCAT). CDCAT is open-source and can handle different input/output formats, preprocessing functions, languages, and annotation systems. It is also efficient-annotators can label a reference relation with just two clicks and best practice tests have revealed that annotators can reach an annotation speed of 0.025 coreference relations per second on a corpus with a coreference density of 0.076 coreference relations per word. As the first multi-language open-source cross-document entity and event coreference annotation tool, CDCAT also achieves higher efficiency than CROMER.

2 Related Work

Many manual annotation tools[14, 15] have been developed for use in intra-document coreference annotation. However, to the best of our knowledge, CROMER is the only open-source manual annotation tool for cross-document entity and event coreference[13].

CROMER is an excellent software program that supports multi-user operation and links between mentions and external knowledge graphs. The well-known corpus, ECB+, is annotated by CROMER. However, there are also a few drawbacks associated with CROMER:

- CROMER is designed only for English and Italian and has no specific interface to support other natural languages. Moreover, uploading corpora in other natural languages sometimes introduces errors.
- As a document-level annotation tool, CROMER links a set of documents to instances that are mentioned at least once in each of those documents. The only way for CROMER to annotate mention-level coreferences is by the use of CAT, i.e., a free-for-research user-friendly intra-document mention coreference annotation tool. Annotators must label the intra-document mention coreferences using CAT and then feed the CAT output into CROMER. However, CAT has recently become unavailable, which means that no open-source cross-document mention coreference annotation tool is currently available.
- The use of two different software tools (CROMER and CAT) rather than one integrated tool leads to some problems. In the top-down annotation strategy, annotators create all instances of interest, which results in the loss of other information. In the down-top annotation strategy, annotators must read the corpora twice: a preliminary reading to create all the instances and label the intra-document mention coreferences, and a careful rereading for labeling cross-document coreference relations. This process limits the level of annotation efficiency.

The above drawbacks motivated our development of a new and more effective multi-language cross-document

[†] https://github.com/hltfbk/CROMER/

[‡] http://dh.fbk.eu/resources/cat-content-annotation-tool

coreference annotation tool.

3 CDCAT

As shown in Fig. 1, CDCAT consists of a platform and four kinds of plugins: input plugins, preprocessing plugins, a CDCAT graphical user interface (GUI), and output plugins, with the platform managing all the labeled data, and the plugins not saving any labeled data. This platform-plugin structure makes it easy to use third-party software. As such, CDCAT can support different types of input, preprocess, and output. After the installation and configuration of CDCAT, the general workflow consists of four steps: (1) input, (2) preprocessing, (3) manual annotation, and (4) output.

3.1 Platform

The primary function of the platform is to manage labeled corpus data. As shown in Fig. 2, the platform has three parts: text, nodes, and instances.

- The text is the plain text of the corpora.
- A node, which consists of a parent node and a number of child nodes, represents a range of characters. There is a default root node, and each character in the text corresponds to a node. More nodes can be added to construct a more complex corpus tree. A node also has configurable labels and two fixed labels: the label "*path*", which is the path from the root node to the node itself, and the label "*coref.instance*", which links the node to an instance. A configurable label is a label that annotators can choose to add or not. The label "*mention.type*" is a typical configurable label for nodes. Typically, researchers make a distinction between entity and event mentions. However, it can be difficult to determine whether some mentions refer to an entity or an event. For the sake of flexibility and scalability, CDCAT has no hard rules for distinguishing between

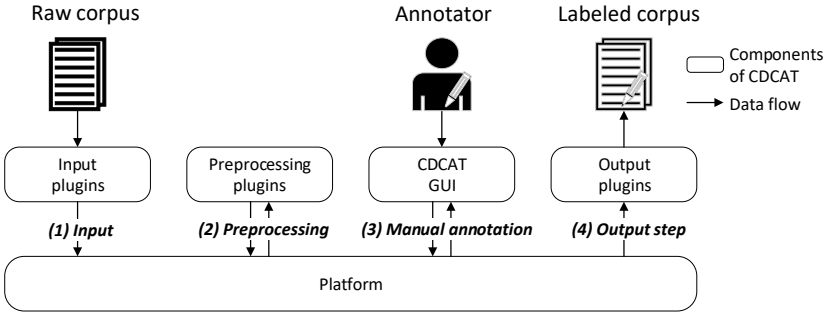

Fig. 1 Components and work flow of CDCAT.

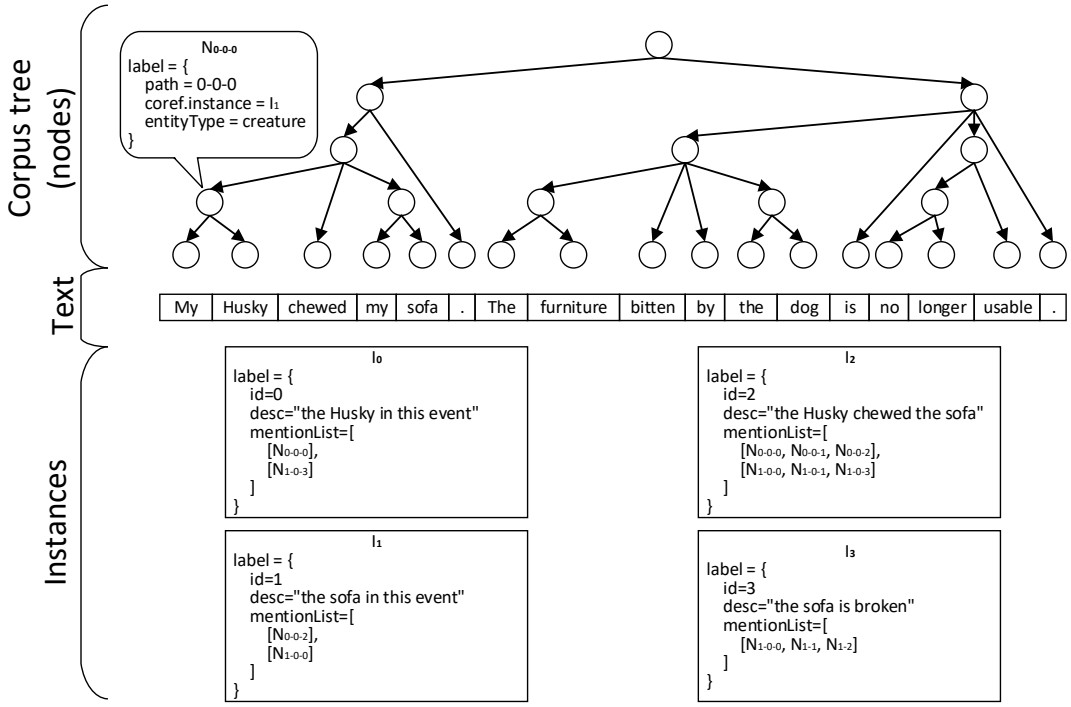

Fig. 2 Structure of annotation data.

entities and events. Accordingly, annotators can assign different values to the label "*mention.type*" of entities or events.

• A mention usually refers to an instance in the real world. An instance has some configurable labels and three fixed labels: the label "*id*", the label "*desc*" (readable description of the instance), and the label "*mentionList*" (a list of nodes that refer to this instance).

There is a two-way reference link between a node and an instance. A node has the label "*coref.instance*" to represent a link from the node to the corresponding instance. Correspondingly, there must be a link to this node in the label "*mentionList*" of the instance. Two nodes are considered to co-refer if they refer to the same instance.

Most annotation systems are supported by the ability to edit configurable labels. Annotators can add a new label to nodes or instances and configure the optional values of the label. The example in Fig. 2 illustrates the basic idea of the node–instance system. Annotators must design a specific annotation system prior to making annotations, i.e., they must decide the kind of mentions that should be labeled and the corresponding labels of nodes or instances required. The config file of CDCAT must then be edited to determine what kind of labels to give the nodes and instances.

3.2 Input

The input of the CDCAT is a corpus, which can either be labeled or not. As CDCAT supports many input plugins, the corpus can have different formats. The corresponding input plugin transforms the input corpus into a corpus tree and passes it to the platform. Table 1 lists all the supported input formats and the corresponding annotation formats.

3.3 Preprocessing

CDCAT is designed for manual annotation. However, some optional auto annotation models are provided as preprocessing plugins to generate potential nodes and reduce the labor required by annotators. Using these plugins, annotators can simply select the generated nodes and label them in the manual annotation step; otherwise, annotators must first create a node for each target mention.

Table 1 Supported input formats and the formats to be supported.

Input format	Supported or not
string/text (plain text)	Supported
string/text (NLTK tree string)	Supported
string/text (CoNLL)	To be supported
string/text (Stanford)	To be supported
file (pickle)	Supported
file (plain text)	Supported
folder	To be supported

It is highly recommended that tokenization and syntactic analysis are performed by calling the corresponding plugins which can create a node for every token and syntactic constituent.

However, it is impossible to support every possible kind of processing plugin because corpora in different natural languages require different processing plugins. For example, the English and Chinese tokenization strategies are different. As such, developers must select a third-party auto annotation model for particular annotation tasks in the target language and convert it into a preprocessing plugin. The conversion process is discussed in Section 4.2.

3.4 Manual annotation

To perform manual annotation, the annotator first calls the GUI function and inputs the corpus tree as a parameter, and then CDCAT GUI will appear. There are five windows in the CDCAT GUI, as shown in Fig. 3:

(1) Content Window: Display the content.

(2) Center Window: Show the text of the current article.

(3) Node Information Window: Describe the node information of the current mention.

(4) Instance Window: List all the instances in order of use history, i.e., an instance is moved to the top of the Instance Window after being clicked. This window is not editable. Annotators cannot group instances or change their order. The instance order is dynamically updated after clicks.

(5) Instance Information Window: Provide information for the current instance.

Loading the whole corpus at once consumes much memory and time. Therefore, a corpus is labeled based on the article unit. Any node with a label "*article: True*" is recognized as an article. All the parent nodes of every article node form a subtree, and this subtree comprises the content of the corpus, as shown in the Content Window.

To present the text of an article, the annotator clicks on the article in the Center Window. Then, the annotators can begin the annotation process, which typically has

Fig. 3 Five windows in the CDCAT GUI for selecting a mention and creating a node.

four steps:

(1) Select a mention: Select a range of characters in the Center Window. The selected characters turn red, as shown in Fig. 3[§], which indicates that these characters are considered as "current mention".

(2) Create and edit the node: If there is already a node for the current mention, the node information appears immediately in the Node Information Window. Otherwise, the annotators must click the "add an annotation node for this mention" button in the Node Information Window to create a node for the current mention. The node information will then be listed in the Node Information Window. Annotators can view and edit annotation information in the Node Information Window, as shown in Fig. 4. The special node label "*coref.instance*" links to an instance. Two mentions that link to the same instance are considered to be co-referred. If there is no corresponding instance, annotators must create a new instance.

(3) Create and edit an instance: Click the "+" button in the Instance Window to create an instance. A new empty button to represent the new instance is then added to the Instance Window, and its information will be listed in the Instance Information Window, as shown in Fig. 4. Annotators can view and edit the annotation information of the current instance in the Instance Information

Fig. 4 Editing current mention or current instance.

[§] Sample text in Figs. 3 and 4 is from http://www.xinhuanet.com/world/2019-03/12/c_1210079099.htm.

Window.

(4) Coreference annotation: To enter edit mode, double click the button after the label "*coref.Instance*" in the Node Information Window. Search for the corresponding instance in the Instance Window; click on the instance to link the current node to this instance, and then exit the edit mode. There can be more than one mention linked to the same instance, and these mentions are co-referred.

In the above steps, five clicks (double click is counted as one) are needed to label a coreference relation. These include a click for the selection of a mention, a click for the creation of a node, a click for the creation of an instance, a double click to enter edit mode of the coreference label, and a click to select the corresponding instance. However, many strategies can be implemented to speed up the annotation process. One strategy is the use of preprocessing plugins to automatically create nodes. Another strategy is to use shortcuts, whereby after Step (1), annotators can click the "→" button in the Instance Window. This operation creates a new node and instance based on a current mention, and links this node to this instance. When using this strategy, only two clicks are required.

3.5 Output

The labeled corpus is saved on the platform and supports output formats like pickle files.

4 Result and Analysis

CDCAT, which is implemented in Python 3.6, is an open-source software available from GitHub[¶]. The CDCAT GUI is Python web software based on the Flask framework, the details of which are provided on GitHub.

4.1 Annotation effectiveness

In CROMER, as noted above, annotators who follow the down-top strategy must skim the corpus and create instances, and then reread the corpus carefully to link the nodes to the corresponding instances, which limits annotation efficiency.

In CDCAT, annotators must read corpora just once, during which they can create a node and an instance based on a mention, and link the node to the instance with just two clicks. This annotation strategy is more efficient than that of CROMER. Table 2 shows some common operations and the ideal average annotation speed.

Table 2 Common operations, the number of mouse clicks required, and the ideal average annotation speed.

Operation	Click count/ average speed
Create a node based on a mention	Two click/ 3.60 seconds per operation (depend on the length of the target mention)
Create an empty instance	One click/ 1.36 seconds per operation
Create a node and instance based on mention, and label the reference relation	Two clicks/ 3.55 seconds per operation
Link a node to an instance	Two clicks/ 2.56 seconds per operation (ignore the time of searching the target instance in Instance Window)
Label a coreference relation from scratch	Six clicks/ 9.02 seconds per operation (ignore the time of searching the target instance in Instance Window)

We note that some factors may interrupt skilled annotators during actual tasks and cause the annotation speed to be much slower than ideal. To evaluate the actual annotation speed, as a test corpus, we used a Chinese coreference corpus containing news from xinhuanet.com regarding Boeing 737 MAX 8 planes crashes. In the annotation of the test corpus, a number of factors slowed the annotation speed, which are listed in Table 3. Corresponding optimizations led to the development of a new version of CDCAT (denoted as CDCAT1 and CDCAT2, respectively).

As shown in Table 3, the editable Instance Window is the main difference between CDCAT1 and CDCAT2. Annotators who spend time organizing instances in the Instance Window can save time when searching for instances. Conversely, less time spent organizing instances may result in long instance search times. Annotators must balance these two factors to determine how they will organize instances and how long this will take.

The annotation speeds of three experiments are compared. In Experiment 1, the annotator using CDCAT1 spent no time organizing instances. In Experiment 2, the annotator using CDCAT2 organized instances mainly by events. In Experiment 3, the annotator, who also used CDCAT2, organized instances mainly by mention types. Examples of instance organization in the three experiments are shown in Figs. 6 – 8, respectively.

The instances in Figs. 6–8 are in Chinese because the

[¶] https://github.com/Zhuo-Ren/cdcat

Table 3 Major factors that influence the annotation speed and corresponding solutions.

Factors related to the annotation speed	Corresponding solutions
The layout of the GUI. A larger scale of the Center Window is not good for the improvement of annotation effectiveness. On the contrary, the larger the scale, the longer it takes to move your mouse from mention to buttons. In annotation Step (4), annotators should search for the target instance in the Instance Window. This is the most time-consuming operation when there are more and more instances. Rename an instance with a more explicit description to make it easier for searching.	The new version, CDCAT2, is designed with a narrow Center Window and an editable Instance Window. Annotators can group instances and change the order of them as they wish to make it easier for annotators to find the target instance in annotation Step (4).
Annotators can not decide whether a mention should be labeled. It would be a waste of time if annotators label the mention and no more co-referred mention appears in the following text; if annotators ignore the mention and a co-referred mention appears in the following text, annotators have to go back and label the first mention. Misoperation and the corresponding rollback operation.	There is no optimization because those factors are about annotators, not the annotation tool. Accordingly, we try to equate those factors when comparing the annotation speed of two different annotation tools. Annotators are told which mention is co-referred with another and asked to annotate twice to get familiar with the tools.
The coreference density of the test corpus. A highly skilled annotator with a perfect annotation tool can not get a high annotation speed if there is only a few coreference relations in the test corpus.	There is no optimization because those factors are about the corpus, not the annotation tool. Accordingly, the coreference density of the test corpus is analyzed, as shown in Fig. 5.

test corpus is in Chinese, but they have been partially translated to enable comprehension by English readers. To measure the annotation speed, we used the average number of coreference relations that one annotator can label in one second. Figure 9 shows the average annotation speed for the n-th article. Figure 10 shows the average annotation speed of the first n articles, in which it is obvious that for the first few articles, the annotation speed of Experiment 1 is higher because the annotator did not spend time organizing instances

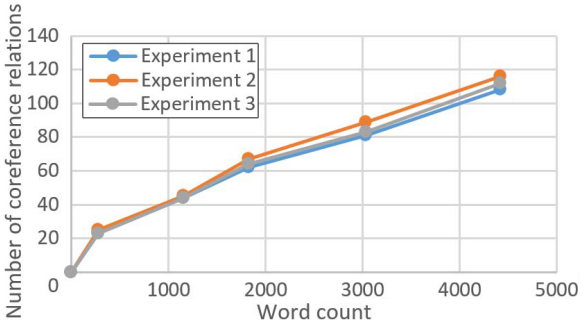

Fig. 5 Each word in the test corpus includes an average of 0.076 coreference relations.

Fig. 6 Organization of instances in Experiment 1.

and could readily find target instances among the small number of instances. However, as the number of words increases, the number of instances correspondingly increases, and the annotator must spend more time searching. In addition, the annotator in Experiment 2 did not achieve a higher speed than that in Experiment 1 because organizing instances by event takes more time than can be saved when searching for instances. The best practice strategy allows for more flexible organization of instances. The annotator who followed this strategy in Experiment 3 organized instances by location, time, institution, and any other type they observed, and thereby attained the highest annotation speed with an increased number of instances.

In conclusion, using CDCAT2 and the best practice strategy (Experiment 3), annotators reached an annotation speed of 0.025 coreference relations per

Fig. 7 Organization of instances in Experiment 2.

Fig. 8 Organization of instances in Experiment 3.

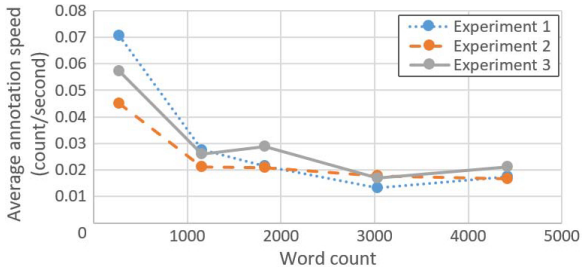

Fig. 9 Average annotation speeds for the *n*-th article.

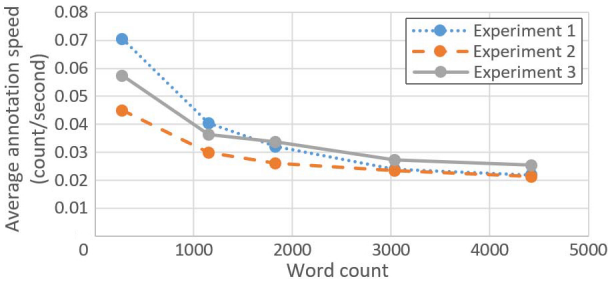

Fig. 10 Average annotation speeds for the first *n* articles.

second on a corpus with a coreference density of 0.076 coreference relations per word. An average speed when using CDCAT2 is 0.023 coreference relations per second.

4.2 Function extension

The data structure of nodes is implemented based on the nltk.tree.ParentedTree, a class implemented in the Natural Language Toolkit (NLTK) package. As such, theoretically, the nodes should work well with NLTK functions.

Developers can extend the functionality of CDCAT by developing new preprocessing plugins based on related third-party models. In a preprocessing plugin, the corpus tree in the platform is converted into the input format of the third-party model, and the output of the third-party model is converted into a corpus tree. Developers do not need to know the internals of CDCAT. They can simply use the interfaces of node and instance to implement the format conversion. Explanations and examples of interfaces are provided in a docstring.

4.3 Language extension

A corpus in any natural language can be supported by CDCAT as long as the corpus is Universal Character Set/Unicode Transformation Format 8 (UTF-8) encoded. However, different natural languages require different preprocessing plugins. For example, Chinese and English corpora have different tokenization strategies. Developers must convert a corresponding third-party model into a new plugin on the platform prior to starting

the annotation process.

The default CDCAT user interface is in English. All the button text is configured based on a configuration file. To translate the user interface into another natural language, the button text listed in the configuration file must simply be translated.

4.4 Annotation system extension

There are a number of default labels for nodes and instances, but other types of labels can be added to nodes and instances. By editing a config file, annotators can choose the labels that are shown in the CDCAT user interface and the optional values that these labels will have. For example, in ECB+, mentions are divided into event and entity categories, which can be divided further into 31 small classes. If an annotator labels a corpus following the annotation system of ECB+, the label "*mention.type*" must be added to the nodes and all 31 small classes set as the optional values of this label in the config file.

5 Discussion and Conclusion

This paper introduced CDCAT, the first multi-language open-source cross-document entity and event coreference annotation tool. CDCAT can handle a range of I/O formats, preprocessing functions, languages, and annotation systems. As no similar tool is available, our comparison with previous results was restricted to the workflow level. Annotators using CDCAT achieved higher efficiency than those using CROMER at the workflow level, and CDCAT was used in an annotation task on a test corpus. During the annotation of this corpus, a number of factors were identified as limiting the efficiency of CDCAT. We then developed a corresponding optimized version, CDCAT2. Using CDCAT2 and the best practice strategy, annotators reached an annotation speed of 0.025 coreference relations per second on a corpus with a coreference density of 0.076 coreference relations per word.

Possible improvements involving the use of hotkeys will be studied in future work. In addition, the labeling of instance relations will be supported by the next version of CDCAT.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61872038), and the Fundamental Research Funds for the Central Universities (No. FRF-GF-19-020B).

References

[1] S. Barhom, V. Shwartz, A. Eirew, M. Bugert, N. Reimers, and I. Dagan, Revisiting joint modeling of cross-document entity and event coreference resolution, in *Proc. 57th Ann. Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 4179–4189.

[2] H. J. Fan, Z. Y. Ma, H. Q. Li, D. S. Wang, and J. F. Liu, Enhanced answer selection in CQA using multi-dimensional features combination, *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 346–359, 2019

[3] Y. F. Gao, P. J. Li, I. King, and M. R. Lyu, Interconnected question generation with coreference alignment and conversation flow modeling, in *Proc. 57th Ann. Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 4853–4862.

[4] M. Liu, B. Lang, Z. P. Gu, and A. Zeeshan, Measuring similarity of academic articles with semantic profile and joint word embedding, *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 619–632, 2017.

[5] P. C. Ma, B. Jiang, Z. G. Lu, N. Li, and Z. W. Jiang, Cybersecurity named entity recognition using bidirectional long short-term memory with conditional random fields, *Tsinghua Science and Technology*, vol. 26, no. 3, pp. 259–265, 2021.

[6] C. Walker, S. Strassel, J. Medero, and K. Maeda, ACE 2005 multilingual training corpus, https://catalog.ldc.upenn.edu/LDC2006T06, 2005.

[7] S. D. Huang, S. Strassel, A. Mitchell, and Z. Y. Song, Shared resources for multilingual information extraction and challenges in named entity annotation, in *Proc. 1st Int. Joint Conf. Natural Language Proc.*, Hainan, China, 2004, pp. 112–119.

[8] N. Reimers and I. Gurevych, Event nugget detection, classification and coreference resolution using deep neural networks and gradient boosted decision trees, in *Proc. 8th Text Analysis Conf.*, Gaithersburg, MD, USA, 2015.

[9] J. Pustejovsky, J. M. Castano, R. Ingria, R. Sauri, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev, Timeml: Robust specification of event and temporal expressions in text, in *Proc. 5th Int. Workshop on Computational Semantics*, Tilburg, Netherlands, 2003.

[10] A. Cybulska and P. Vossen, Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution, in *Proc. 9th Int. Conf. Language Resources and Evaluation*, Reykjavik, Iceland, 2014, pp. 4545–4552.

[11] C. A. Bejan and S. Harabagiu, Unsupervised event coreference resolution with rich linguistic features, in *Proc. 48th Ann. Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 1412–1422.

[12] H. Lee, M. Recasens, A. Chang, M. Surdeanu, and D. Jurafsky, Joint entity and event coreference resolution across documents, in *Proc. 2012 Joint Conf. Empirical Methods in Natural Language Proc. Computational Natural Language Learning*, Jeju Island, Korea, 2012, pp. 489–500.

[13] C. Girardi, M. Speranza, R. Sprugnoli, and S. Tonelli, Cromer: A tool for cross-document event and entity coreference, in *Proc. 9th Int. Conf. Language Resources and Evaluation*, Reykjavik, Iceland, 2014, pp. 3204–3208.

[14] V. B. Lenzi, G. Moretti, and R. Sprugnoli, Cat: the celct annotation tool, in *Proc. 8th Int. Conf. Language Resources and Evaluation*, Istanbul, Turkey, 2012, pp. 333–338.

[15] P. Stenetorp, S. Pyysalo, G. Topic, T. Ohta, S. Ananiadou, and J. Tsujii, Brat: A web-based tool for nlp-assisted text annotation, in *Proc. 13th Conf. European Chapter of the Association for Computational Linguistics*, Avignon, France, 2012, pp. 102–107.

Yang Xu received the BEng degree from the Software College, Hebei Normal University, Hebei, China in 2014, and is pursuing the PhD degree in the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His research interests include event detection and analysis.

Boming Xia received the BEng degree from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China in 2018, where he is currently pursuing the master degree. His research interests include natural language processing and block chain.

Yueliang Wan received the PhD degree from the School of Computer Science, Beijing Institute of Technology, China in 2007. He is currently the director of the Research Institute with Run Technologies Company, Ltd., Beijing, China. He focuses on the content security in Internet, multimedia analysis, and Internet search and mining. His research interests include Internet multimedia retrieval, privacy protection, and data center network.

Fan Zhang received the master degree from Beijing University of Science and Technology in 2016, and is currently pursuing the PhD degree from the School of Computer and Communication Engineering. He focuses on network information security. His research interests include information encryption and information transmission.

Jiabo Xu received the BS degree from Yantai University, Shandong, China in 2004, and the PhD degree from Xinjiang University, Urumqi, China in 2011. He is currently a professor and the vice dean with the School of Information Engineering, Xinjiang Institute of Engineering. His current research interests include big data and Internet of Things.

Huansheng Ning received the BS degree from Anhui University, Anhui, China in 1996 and the PhD degree from Beihang University, Beijing, China in 2001. He is a professor and vice dean with the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His research interests include Internet of Things, cyber physical social systems, and cyberspace data and intelligence. He is a fellow IET and a senior member of IEEE. He has authored 7 books and over 200 papers.