CoRefi: A Crowd Sourcing Suite for Coreference Annotation

Aaron Bornstein[1,2] **Arie Cattan**[1] **Ido Dagan**[1]
[1]Computer Science Department, Bar Ilan University, Ramat-Gan, Israel
[2]Microsoft Corporation, Tel-Aviv, Israel
abornst@microsoft.com arie.cattan@gmail.com dagan@cs.biu.ac.il

Abstract

Coreference annotation is an important, yet expensive and time consuming, task, which often involved expert annotators trained on complex decision guidelines. To enable cheaper and more efficient annotation, we present COREFI, a web-based coreference annotation suite, oriented for crowdsourcing. Beyond the core coreference annotation tool, COREFI provides guided onboarding for the task as well as a novel algorithm for a reviewing phase. COREFI is open source and directly embeds into any website, including popular crowdsourcing platforms.

COREFI Demo: aka.ms/corefi Video Tour: aka.ms/corefivideo
Github Repo: https://github.com/aribornstein/corefi

1 Introduction

Coreference resolution is the task of clustering textual expressions (*mentions*) that refer to the same concept in a described scenario. This challenging task has been mostly investigated within a single document scope, seeing great research progress in recent years. The rather under-explored cross-document coreference setting is even more challenging. For example, consider the following sentences originating in two different documents in the standard cross-document coreference dataset ECB+ (1):

1. *A man suspected of **shooting** three people at an accounting firm where he had worked ...*
2. *A gunman **shot** three people at a suburban Detroit office building Monday morning.*

Recognizing that both sentences refer to the same event ("shooting","shot") at the same location ("accounting firm", "Detroit office") can be very useful for downstream tasks, particularly across documents, such as multi-document summarization (2; 3) or multi-hop question answering (4; 5).

High-quality annotated datasets are valuable to develop efficient models. While Ontonotes (6) provides a useful dataset for generic single-document coreference resolution, large-scale datasets are lacking for cross-document coreference (1; 7; 8) or for targeted domains, such as medical (9). Due to the complexity of the coreference task, existing datasets have been annotated mostly by linguistic experts, incurring high costs and limiting annotation scale.

Aiming to address the cost and scalability issues in coreference annotation, we present COREFI, an embeddable web-component tool suite that supports an end-to-end crowdsourcing process (Figure 1), while providing several contributions over earlier annotation tools (Section 4). COREFI includes an automated onboarding training phase, familiarizing annotators with the tool functionality and the task decision guidelines. Then, actual annotation is performed through a simple-to-use and efficient interface, providing quick keyboard operations. Notably, COREFI provides a reviewing mode, by which an additional annotator reviews and improves the output of an earlier annotation. This mode is

Preprint. Under review.

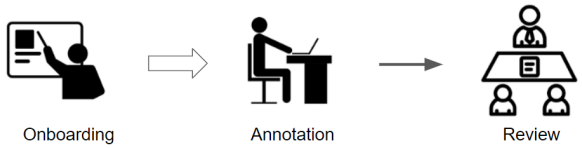

Figure 1: COREFI's end-to-end annotation process

enabled by a non-trivial algorithm, that seamlessly integrates reviewing of an earlier annotation into the progressive construction of the reviewer's annotation.

By open sourcing COREFI, we hope to facilitate the creation of large-scale coreference datasets, especially for the cross-document setting, at modest cost while maintaining quality.

2 The COREFI Annotation Tool

COREFI provides a suite for annotating single and cross-document coreference, designed to embed into crowdsourcing environments. Since coreference annotation is an involved and complex task, we target a *controlled crowdsourcing* setup, as proposed by Roit et al. (10). This setup consists of selecting designated promising crowd workers, identified in preliminary trap-tasks, and then quickly training them for the target task and testing their performance. This yields a pool of reliable lightly trained annotators, who perform the actual annotation of the dataset.

COREFI supports both the annotator training (onboarding) and annotation production phases, as illustrated in Figure 1. The training phase (Section 2.4) consists of two crowdsourcing tasks, first teaching the tool's functionality and then practicing guided annotation, interactively learning basics of the annotation guidelines. The annotation production phase also consists of two crowdsourcing tasks: first-round coreference annotation, providing a user-friendly interface designed to reduce annotation time (Section 2.2), and a reviewing task, in which an additional annotator reviews and improves the initial annotation (Section 2.3).

2.1 Design Choices

Our first major design choice regards the annotation flow. As elaborated in Section 4, two different coreference annotation flows were prominent in prior work. The local pair-based approach aims at annotation simplicity, often motivated by a crowdsourcing setting. Here, an annotator has to decide for a pair of mentions whether they corefer or not, or to proactively find such pairs of corefering mentions. Since coreference is annotated at the level mention pairs, it might require, in the worst case, comparing a mention to all other mentions in the text.

In the cluster-based flow, annotators assign mentions to coreference *clusters*. Here, a mention needs to be compared only against the clusters accumulated so far, or otherwise be defined as starting a new cluster. Indeed, the number of coreference clusters is often substantially lower than the number of mentions, particularly in the cross-document setting, where the same content gets repeated across the multiple texts. For example, in the most popular dataset for cross-document coreference, ECB+ (1), the number of clusters is about one third of the number of mentions (15122 mentions split into 4965 coreference clusters, including singletons). In COREFI, we adopt the cluster-based approach since we aim at exhaustive coreference annotation across documents, whose complexity would become too high under the pairwise approach. At the same time, we simplify the annotation process and functionality, making it crowdsourceable.

Our second design choice regards detecting referring mentions in text. As elaborated in Section 4, coreference annotation tools, particularly cluster-based (e.g. (11; 12)), often require annotators to first detect the target mentions before annotating them for coreference. Conversely, recent local pair-based decision tools (13; 14) delegate mention extraction to a preprocessing phase, presenting coreference annotators with pre-determined mentions. This simplifies the task and allows annotators to focus their attention on the coreference decisions.

As we target exhaustive crowdsourced coreference annotation, we chose to follow this recent facilitating approach. In addition to the input texts, COREFI takes as input an annotation of the targeted

mentions, while optionally allowing annotators to fix this mention annotation. In our tool suite, we followed the approach of Prodigy,[1] where corpus developers may implement their own automated (non-overlapping) mention extraction recipes, or use a separate manual annotation tool for mention annotation, according to their desired mention detection guidelines (which often vary across projects). The resulting mentions can then be fed into COREFI for coreference annotation. We provide an example mention extraction recipe that detects as mentions common nouns, proper nouns, pronouns, and verbs (for event coreference). Such mention detection is consistent with approaches that consider reduced mention spans, mostly pertaining to syntactic heads or named entity spans (15).

2.2 Annotation

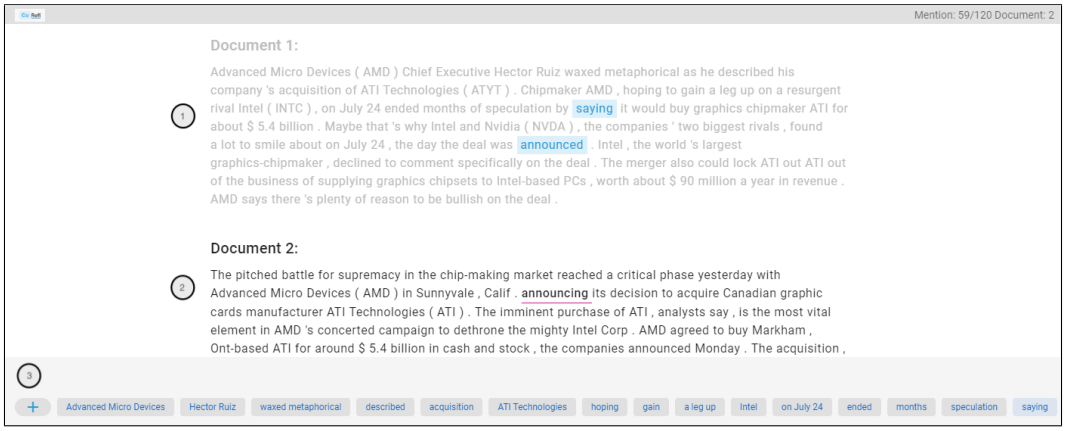

Figure 2: Annotation Interface of COREFI, presenting text and mentions from the ECB+ dataset, used in our pilot study (Section 3). The current mention to assign is underlined in purple (2). The selected cluster is highlighted in blue in the cluster bank (3), along with its mentions in the text (1).

Figure 2 shows the annotation interface of COREFI.

As initialization, the first candidate mention is automatically assigned to the first coreference cluster, which is placed in the "cluster bank", appearing at the bottom of the screen ((3) in Figure 2). In this bank, each cluster is labeled by the text of its first mention. The annotator is then shown the subsequent mentions, one at a time, with the current mention to assign underlined in purple (2). For each mention, the annotator decides whether to accept it as a valid mention (doing nothing) or to modify its span (easily highlighting the correct span and pressing the 'F' key, for "Fix"). Similarly, the annotator may introduce a new span, missing from the input. To simplify annotation, the tool allows only non-overlapping spans.

The annotator then makes a coreference decision, by assigning the current mention to a new or existing cluster. An existing cluster can be rapidly selected either by selecting it in the cluster bank or by selecting one of its previously-assigned mentions in the text. Once a cluster is selected, it is highlighted in blue along with all its previous text mentions ((3) and (1) in the figure). Rather than assigning mentions to clusters through a slower drag and drop interface (11; 12) or buttons (16; 17), annotation is driven primarily by faster keyboard operations, such as SPACE (assign to an existing cluster) and CTRL+SPACE (new cluster), with quick navigation through arrow keys and mouse clicks.

At any point, the annotator can re-assign a previously assigned mention to another cluster or view any cluster mentions. COREFI supports an unlimited number of documents to be annotated, presented sequentially in a configurable order. Finally, COREFI guarantees exhaustive annotation by allowing task submission only once all candidate mentions are processed.

2.3 Reviewing

To promote annotation quality, annotation projects typically rely on multiple annotations per item. One approach for doing that involves collecting such annotations in parallel and then merging them

[1] https://prodi.gy/

in some way, such as simple or sophisticated voting (18). Another approach is sequential, where one or more annotations are collected initially, and are then *manually* consolidated by an additional, possibly more reliable, annotator (a "consolidator" or "reviewer") (10).

In our case, coreference annotation is addressed as a global clustering task, where an annotator generates a complete clustering configuration for the input text(s). Automatically merging such multiple clustering configurations, where cluster assignments are mutually dependent, might become unreliable. Therefore, in COREFI we follow the sequential manual reviewing approach. To that end, we introduce a reviewing task, which receives as input a previously annotated clustering configuration and allows an additional annotator to review and improve it.

The reviewing task follows the same flow of the annotation task, making it trivial to learn for annotators that already experienced with COREFI annotation. At each step, the reviewer is presented with the next mention in the reviewed configuration, and may first decide to modify its span. Next, the reviewer has to decide on cluster assignment for the current mention. The only difference at this point is that the reviewer is presented with candidate cluster assignments which reflect the original annotator assignment (as explained below), displayed just above the cluster bank (Figure 3).

In fact, it is not trivial to reflect the cluster assignment by the original annotator to the reviewer, since that assignment has to be mapped to the current clustering configuration of the reviewer. Ambiguity may arise, resulting in multiple candidate clusters, since an early cluster modification by the reviewer can impact the interpretation of downstream cluster assignments in the original annotation.

To illustrate this issue, consider reviewing a cluster assigned by the original annotator, consisting of three mentions, $\{A, B, C\}$. When presented with the mention A, the reviewer agrees that it starts a new cluster. Then, when reaching B, the reviewer is presented with $\{A\}$ as B's original cluster assignment. Suppose the reviewer disagrees with the annotator that A and B corefer and decides to assign B to a new cluster. Now, when reviewing the mention C, it is no longer clear whether to attribute C's original assignment to $\{A\}$ or $\{B\}$. Hence, the reviewing tool presents both $\{A\}$ and $\{B\}$ as candidate clusters that reflect the original annotator's assignment. The reviewer may then choose either of them, or override the original annotation altogether and make a different assignment. Similar ambiguities arise when the reviewer splits an original mention span and assigns its parts to separate clusters.

To address this challenge, we formulate an algorithm that maps an original cluster assignment to a set of candidate clusters in the current clustering configuration of the reviewer. Generally speaking, the algorithm considers the cluster to which the current mention was assigned in the original annotation, and tracks all earlier token positions in that cluster. These token positions are then mapped back to clusters in the current reviewer's clustering configuration, which become the candidate clusters presented to the reviewer. The algorithm pseudocode is presented in Appendix A, along with a comprehensive example of its application.

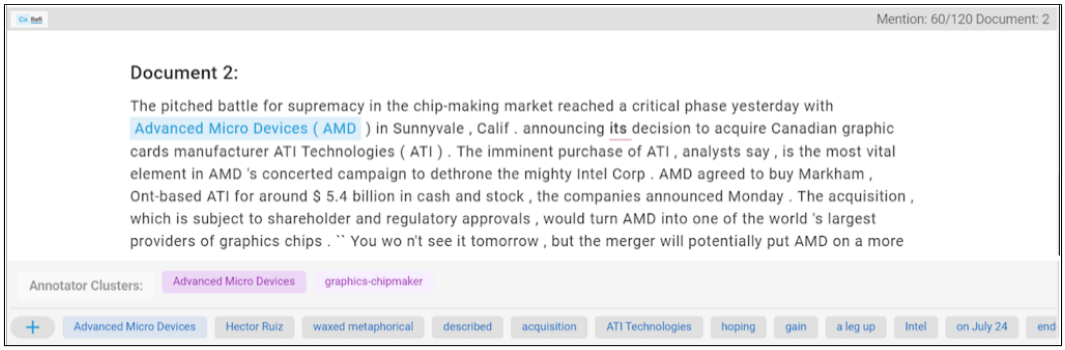

Figure 3: Reviewing interface of COREFI. Candidate clusters found by the reviewing algorithm are shown in purple.

2.4 Onboarding

In the proposed controlled crowdsourcing scheme of (10), annotators were trained in an "offline" manner, reading slides and receiving individual feedback. We propose augmenting this phase with automated training, delivered through two crowdsourcing tasks, described below.

2.4.1 Walk-through Tutorial

During this task, a trainee is walked through the core concepts and functionality of COREFI, such as the "current mention" and the "cluster bank" (Figure 2), and through the annotation operations. These functionalities are presented through a series of intuitive dialogues. To ensure that each feature is correctly understood, the user is instructed to actively perform each operation before continuing to the next (see Appendix B).

2.4.2 Guided Annotation

After acclimating with COREFI's features, users are familiarized with the coreference decision guidelines through a guided annotation task, practicing annotation while receiving automated guiding feedback. If an annotation error is made, the trainee is notified with a pre-prepared custom response, which guides to the correct decision before allowing to proceed. Additionally, following certain decisions, specific important guidelines can be communicated (see Appendix C for examples).

The content of the guided annotation task and the automated responses are easily configurable using a simple JSON configuration schema. This allows tailoring them when applying COREFI for different datasets and annotation guidelines.

Augmenting the controlled crowdsourcing scheme (10) with automated training provides key benefits. First, since feedback is automated, the amount of required personalized manual feedback is reduced. Second, annotators benefit from an immediate response for each decision, allowing them to understand their mistakes earlier and improve in real time. We suggest that, for optimal learning of annotation guidelines, these benefits should be coupled with the additional training means of controlled crowdsourcing. These include the provision of guideline slides, for learning and later for reference, and some personalized manual feedback during the training phase.

2.5 Implementation Benefits

COREFI was developed using the web component standard and the VUE.JS framework.[2] allowing, it to easily embed into any website, including crowdsourcing platforms. Additionally, COREFI provides output in the standard CoNLL coreference annotation format, enabling training state of the art models and scoring with the official coreference scorer (19). All COREFI features are easily configurable with HTML encoded JSON and support any UTF-8 encoded language.

3 Pilot Study

To further assess COREFI's effectiveness in a crowdsourcing environment, we performed a small-scale trial on Amazon Mechanical Turk, employing 5 annotators, focusing on the coreference annotation functionality (rather than mention validation). To allow objective assessment of annotation quality, we experimented with replicating coreference annotations from the ECB+ dataset (1), the commonly used dataset for cross-document coreference over English news articles (20; 21; 22; 23; 24; 25). Accordingly, we considered the ECB+ gold mentions as input, requesting crowdworkers to assign them to coreference clusters. Focusing on the controlled crowdsourcing setting, we hired five annotators that were previously selected for annotation by (10).

On-boarding Annotators were given COREFI's walk-through tutorial and guided annotation tasks, adapted to the ECB+ guidelines and applied to a part of an ECB+ subtopic (cluster of documents). These two tasks took altogether 11 minutes on average to complete, at a rate of $1.5 a task. Next, workers were asked to annotate an entire ECB+ subtopic (through the actual annotation task). We

[2] https://vuejs.org/

	MUC	B^3	CEAFe	CoNLL
A1	94.0	85.0	77.8	85.6
A2	94.8	91.2	85.4	90.5
A3	95.2	90.2	84.7	90.0
A4	94.8	86.9	76.0	85.9
A5	92.1	82.5	75.7	83.4

Table 1: F1 results of 5 annotators on 2 ECB+ subtopics.

	MUC	B^3	CEAFe	CoNLL
A1	87.0	69.1	62.6	72.9
R1	91.9	80.4	79.8	80.0
R2	88.0	73.4	73.3	78.2
A5	87.0	79.0	62.5	76.2
R1	92.9	87.3	73.0	84.4
R2	90.0	86.2	63.0	79.7

Table 2: F1 results of the reviewing trial.

provided them manual feedback for their mistakes, which consumed 30-40 minutes of researcher time per trained annotator.

Annotation After training, we paid workers $8 to annotate two additional subtopics in full (of about 150 and 200 mentions; in ECB+ only a few sentences are annotated per document, and these were presented for annotation). Each subtopic took 27 minutes on average to annotate, corresponding to an annotation rate of ~400 mentions per hour.

Table 1 presents the performance (F1) of each of the annotators, compared to the ECB+ gold annotations, averaged over the two subtopics. The results are reported using the common evaluation metrics for coreference resolution: **MUC** (26), **B^3** (27), **CEAFe** (28), and **CoNLL** — the average of the three metrics. Considering the decision volume and complexity, as well as the limited training (not providing guideline slides and a single practice round), we find that these results support COREFI's effectiveness for crowdsourcing.[3]. As previously mentioned, we expect that annotation quality may be further improved in an actual dataset creation project, by providing additional guided tasks, annotation guidelines slides, and additional manual feedback.

Reviewing For the reviewing trial, the two best annotators, A2 and A3, were selected as R1 and R2. Two additional annotators (A1 and A5) were each assigned a new unique subtopic, which was then reviewed by both R1 and R2. Table 2 presents the reviewing results, showing consistent improvements after reviewing and assessing the ease of using the reviewing functionality.

4 Related work

As mentioned in Section 2, prior tools for coreference annotation are based on two prominent workflows: pair-based, treating coreference as a pairwise annotation decision, and cluster-based, in which mentions are assigned to clusters. While targeting simplicity, only two pair-based tools supported crowdsourcing annotation, yet they were not applied for producing exhaustively annotated

[3]There are no comparable annotator performance evaluations in the literature. Ontonotes (6) reports an inter-annotator agreement for experts of 0.87 MUC scores, but these seem to include mention span decisions. The creators of ECB+ (1) formulates annotation as a multi-class classification problem and use Kappa to calculate inter-annotator agreement, reporting a Kappa score of 0.76. Kappa however is not applicable for our setting since the number of clusters are variable between annotators.

daasets: Phrase Detective (13), which was employed in a web-based game setting, and (14), which was applied in an active learning environment.

Pair-based tools differ in their annotation approaches. In certain tools, such as BRAT (29), Glozz (30), Analec (31), and MMAX2 (32), the annotator first determines mention span boundaries and then links a pair of mentions. Other Pair-based tools (13; 14) either provide annotators a single (pre-determined) mention, asking to find a coreferring antecedent, or provide a pair of mentions, asking to judge whether the two corefer. Notably, pair-based tools are less effective for exhaustive coreference annotation, for two reasons. First, they require comparing each mention to all other *mentions*, rather than to already constructed clusters. Second, local pairwise decisions lack awareness of previous cluster assignments, which might hurt annotation quality.

Cluster-based tools, including Cromer (16), Model based annotation tool (17), CorefAnnotator (11), and SACR (12), ask annotators to first detect mention spans and then cluster them, thus complicating the overall task without allowing the delegation of mention detection to a preprocessing phase. Such a method does not guarantee exhaustive annotation, since annotators may miss some mentions. With respect to operation efficiency, mentions are often linked to clusters via somewhat slow operations, such as drag-and-drop or selection from a drop-down list, in comparison to the fast keyboard operations in COREFI.

Notably, to the best of our knowledge, COREFI is the first cluster-based crowdsourcing tool that provides an end-to-end annotation suite, including automated onboarding tasks and exhaustive reviewing, the latter enabled by our novel reviewing algorithm. Furthermore, it is the first tool that was developed using the WebComponent standard, embeddable in any website.

5 Conclusion

In this paper, we aim to facilitate crowdsourced creation of needed large-scale coreference datasets, in both the within- and the cross-document setting. Our comprehensive end-to-end tool suite, COREFI, enables high quality and fairly cheap crowdsourcing of exhaustive coreference annotation in various domains and languages. Our experiments demonstrate that COREFI's automatic onboarding is effective at augmenting Roit et al.(10)'s controlled crowdsourcing. COREFI provides the first reviewing algorithm and implementation for cluster-based coreference annotation. Overall, we demonstrated that non-expert annotators can be trained to effectively perform and review coreference annotations, allowing for cost-effective annotation efforts.

Acknowledgments

The work described herein was supported in part by grants from Intel Labs, Facebook, the Israel Science Foundation grant 1951/17, the Israeli Ministry of Science and Technology and the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

In addition to the support above, we would like to thank Uri Fried, Ayal Klein, Paul Roit, Amir Cohen, Sharon Oren, Chris Noring, Asaf Amrami, Ori Shapira, Daniela Stepanov, Ori Ernst, Yehudit Meged, Valentina Pyatkin, Moshe Uzan, and Ofer Sabo for their support with architecture, design and crowdsourcing.

References

[1] A. Cybulska and P. Vossen, "Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. Reykjavik, Iceland: European Languages Resources Association (ELRA), May 2014, pp. 4545–4552. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/pdf/840_Paper.pdf

[2] T. Falke, C. M. Meyer, and I. Gurevych, "Concept-map-based multi-document summarization using concept coreference resolution and global importance optimization," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 801–811. [Online]. Available: https://www.aclweb.org/anthology/I17-1081

[3] K. Liao, L. Lebanoff, and F. Liu, "Abstract meaning representation for multi-document summarization," in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1178–1190. [Online]. Available: https://www.aclweb.org/anthology/C18-1101

[4] B. Dhingra, Q. Jin, Z. Yang, W. Cohen, and R. Salakhutdinov, "Neural models for reasoning over multiple mentions using coreference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 42–48. [Online]. Available: https://www.aclweb.org/anthology/N18-2007

[5] H. Wang, M. Yu, X. Guo, R. Das, W. Xiong, and T. Gao, "Do multi-hop readers dream of reasoning chains?" in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 91–97. [Online]. Available: https://www.aclweb.org/anthology/D19-5813

[6] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, "CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes," in *Joint Conference on EMNLP and CoNLL - Shared Task*. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 1–40. [Online]. Available: https://www.aclweb.org/anthology/W12-4501

[7] A.-L. Minard, M. Speranza, R. Urizar, B. Altuna, M. van Erp, A. Schoen, and C. van Son, "MEANTIME, the NewsReader multilingual event and time corpus," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 4417–4422. [Online]. Available: https://www.aclweb.org/anthology/L16-1699

[8] P. Vossen, F. Ilievski, M. Postma, and R. Segers, "Don't annotate, but validate: a data-to-text method for capturing event data," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. Miyazaki, Japan: European Languages Resources Association (ELRA), May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1480

[9] N. Nguyen, J.-D. Kim, and J. Tsujii, "Overview of BioNLP 2011 protein coreference shared task," in *Proceedings of BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 74–82. [Online]. Available: https://www.aclweb.org/anthology/W11-1811

[10] P. Roit, A. Klein, D. Stepanov, J. Mamou, J. Michael, G. Stanovsky, L. Zettlemoyer, and I. Dagan, "Controlled crowdsourcing for high-quality QA-SRL annotation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7008–7013. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.626

[11] N. Reiter, "CorefAnnotator - A New Annotation Tool for Entity References," in *Abstracts of EADH: Data in the Digital Humanities*, December 2018.

[12] B. Oberle, "SACR: A drag-and-drop based tool for coreference annotation," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. Miyazaki, Japan: European Languages Resources Association (ELRA), May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1059

[13] J. Chamberlain, M. Poesio, and U. Kruschwitz, "Phrase detectives corpus 1.0 crowdsourced anaphoric coreference." in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 2039–2046. [Online]. Available: https://www.aclweb.org/anthology/L16-1323

[14] B. Z. Li, G. Stanovsky, and L. Zettlemoyer, "Active learning for coreference resolution using discrete annotation," in *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics. Online: Association for Computational Linguistics, Jul. 2020, pp. 8320–8331. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.738

[15] T. O'Gorman, K. Wright-Bettner, and M. Palmer, "Richer event description: Integrating event coreference with temporal, causal and bridging annotation," in *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 47–56. [Online]. Available: https://www.aclweb.org/anthology/W16-5706

[16] C. Girardi, M. Speranza, R. Sprugnoli, and S. Tonelli, "CROMER: a tool for cross-document event and entity coreference," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. Reykjavik, Iceland: European Languages Resources Association (ELRA), May 2014, pp. 3204–3208. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/pdf/726_Paper.pdf

[17] R. Aralikatte and A. Søgaard, "Model-based annotation of coreference," in *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 74–79. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.9

[18] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy, "Learning whom to trust with MACE," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 1120–1130. [Online]. Available: https://www.aclweb.org/anthology/N13-1132

[19] S. Pradhan, X. Luo, M. Recasens, E. Hovy, V. Ng, and M. Strube, "Scoring coreference partitions of predicted mentions: A reference implementation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 30–35. [Online]. Available: https://www.aclweb.org/anthology/P14-2006

[20] A. Cybulska and P. Vossen, "Translating granularity of event slots into features for event coreference resolution." in *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*. Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 1–10. [Online]. Available: https://www.aclweb.org/anthology/W15-0801

[21] B. Yang, C. Cardie, and P. Frazier, "A hierarchical distance-dependent Bayesian model for event coreference resolution," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 517–528, 2015. [Online]. Available: https://www.aclweb.org/anthology/Q15-1037

[22] P. K. Choubey and R. Huang, "Event coreference resolution by iteratively unfolding inter-dependencies among events," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2124–2133. [Online]. Available: https://www.aclweb.org/anthology/D17-1226

[23] K. Kenyon-Dean, J. C. K. Cheung, and D. Precup, "Resolving event coreference with supervised representation learning and clustering-oriented regularization," in *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1–10. [Online]. Available: https://www.aclweb.org/anthology/S18-2001

[24] 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 *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4179–4189. [Online]. Available: https://www.aclweb.org/anthology/P19-1409

[25] A. Cattan, A. Eirew, G. Stanovsky, M. Joshi, and I. Dagan, "Streamlining cross-document coreference resolution: Evaluation and modeling," *ArXiv*, vol. abs/2009.11032, 2020.

[26] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman, "A model-theoretic coreference scoring scheme," in *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, 1995. [Online]. Available: https://www.aclweb.org/anthology/M95-1005

[27] A. Bagga and B. Baldwin, "Entity-based cross-document coreferencing using the vector space model," in *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998. [Online]. Available: https://www.aclweb.org/anthology/C98-1012

[28] X. Luo, "On coreference resolution performance metrics," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, Oct. 2005, pp. 25–32. [Online]. Available: https://www.aclweb.org/anthology/H05-1004

[29] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "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*. Avignon, France: Association for Computational Linguistics, Apr. 2012, pp. 102–107. [Online]. Available: https://www.aclweb.org/anthology/E12-2021

[30] A. Widlöcher and Y. Mathet, "The glozz platform: A corpus annotation and mining tool," in *Proceedings of the 2012 ACM Symposium on Document Engineering*, ser. DocEng '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 171–180. [Online]. Available: https://doi.org/10.1145/2361354.2361394

[31] F. Landragin, T. Poibeau, and B. Victorri, "ANALEC: a new tool for the dynamic annotation of textual data," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. Istanbul, Turkey: European Languages Resources Association (ELRA), May 2012, pp. 357–362. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/638_Paper.pdf

[32] M. Kopeć, "MMAX2 for coreference annotation," in *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 93–96. [Online]. Available: https://www.aclweb.org/anthology/E14-2024

A Reviewing Algorithm

Algorithm 1 implements the mechanism to find potential clusters given the initial annotation and previous reviewing modifications.

To support span modification, we build two main data structures at the *token* level (lines 1 and 2). Given the original annotation M, we build a static mapping Ant (line 1), where each single token is associated with all tokens from previous mentions that belong to the same coreference cluster. $T2C$ is a growing mapping that will keep track of the reviewer decisions.

After the initialization phase, the reviewer is shown all the annotator mentions in a sequential order. For each presented mention, the reviewer first decides whether to agree or to modify the mention span boundaries (line 5). Future mentions in the stack M that are fully covered by the reviewed span need to be removed (lines 6-8). The reviewer may also split the current mention or partially cover next mentions (line 9-11).

In order to find the potential coreference clusters (line 13), we first use Ant to retrieve the antecedent tokens in the original annotation, for each single token in the reviewed span Sp'. Then, we use the reviewer mapping $T2C$ for each antecedent tokens to identify the possible cluster(s) that will be displayed to the reviewer (Figure 3). Given the coreference decision (assigning to an existing cluster or to a new one) of the reviewer (line 14), we update the reviewer mapping $T2C$ (line 15) and coreference assignments (line 16).

Table 3 illustrates the reviewing decision step by step, given an initial annotation that incorrectly assigned the following gold clustered mentions {{Bank of America, bank, BoA} {American}} into one coreference cluster {Bank of America, American bank, BoA}.

Algorithm 1: Reviewing Algorithm

Input: M: Stack of mentions with their initial clustering assignment
Output: R: Reviewed Assignment

$Ant \leftarrow CreateAntecedentMapping(M)$
$T2C$: Map of token to cluster ID
while M *not empty* **do**
    // Set reviewer span
    $Sp' \leftarrow$ ReviewSpan($M.top()$)
    while $M.top().end \leq Sp'.end$ **do**
        $M.pop()$
    end
    if $M.top().start \leq Sp'.end$ **then**
        popSplitPush(M, Sp')
    end
    // Set reviewer cluster
    $C \leftarrow getCandidates(Sp', Ant, T2C)$
    $cluster \leftarrow selectCluster(C)$
    $T2C.update(Sp', cluster)$
    $R.push(Sp', cluster)$
end

	Mention Stack	Annotator Candidates	Reviewer Decision	Explanation
1	[**Bank Of America**, American bank, BoA]	{Bank of America}	✓	The reviewer agrees that Bank of America is the start of a new cluster
2	[**American bank**, BoA]	{Bank of America}	Split **American bank** into two mentions	Two mentions are created, **American** and **bank**. The reviewer will next determine the cluster assignment of American.
3	[**American** bank, BoA]	{Bank of America}	Assign **American** to a new cluster	The reviewer is shown {Bank of America} as the candidate cluster since the *token* **American** was assigned with Bank of America by the annotator.
4	[**bank**, BoA]	{Bank of America}, {American}	Assign **bank** to the {Bank of America}	The reviewer is shown both {Bank of America} and {American} as candidate clusters for bank. Now, the reviewer decides to assign bank to the {Bank of America} cluster.
5	[**BoA**]	{{Bank of America, bank}, {American}}	Assign **BoA** to cluster {Bank of America, bank}	The reviewer is shown two candidate clusters {Bank of America, bank} and {American} which correspond to the clusters that include the antecedent tokens of {BoA} initially assigned by the annotator (Bank of America, American bank, BoA).

Table 3: Examples of reviewing assignment, the initial clustering assignment is [(Bank of America, American bank, BoA] and the reviewer modifies into [(Bank of America, bank, BoA), (American)]

B Tutorial

Figure 4 and 5 demonstrate notifications that explains conceptual aspect of CoRefi. Figure 4 explains what the current mention to assign is where as Figure 5 explains what clusters are and how to manage them in the cluster bank. Figure 6 demonstrates a more interactive tutorial prompt. It shows how to make an active coreference decision with the keyboard and encourages the trainee to experiment with in the confines of the tutorial environment to familiarize themselves with the feature.

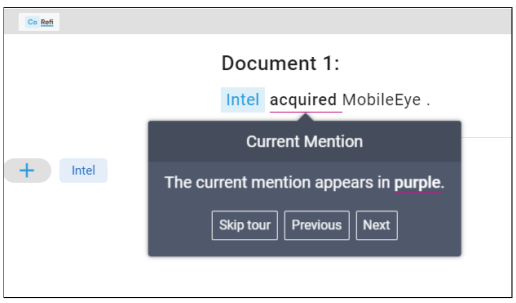

Figure 4: Example of the tutorial explaining the current mention.

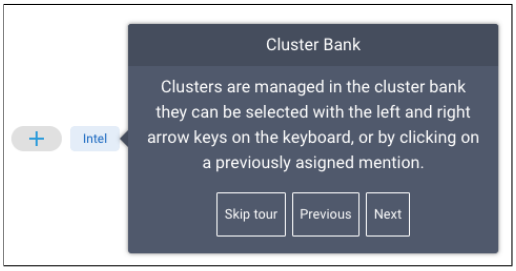

Figure 5: Example of the tutorial explaining the cluster bank.

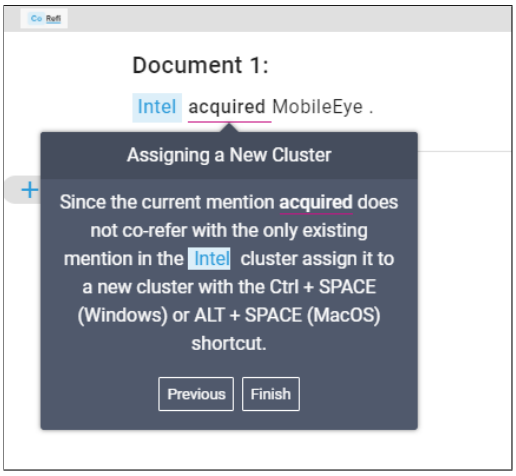

Figure 6: Example of the tutorial explaining the cluster assignment operation.

C Guided Annotation

Figure 7 demonstrates the guided experience of the on-boarding flow of CoREFI. In Figure 7, the trainee is learning the nuances of coreference and makes the mistake of attempting to assign the mention name to the same cluster as another mention with the exact same expression. However, in context the name event mention expressed by the current mention does not refer to the selected cluster.

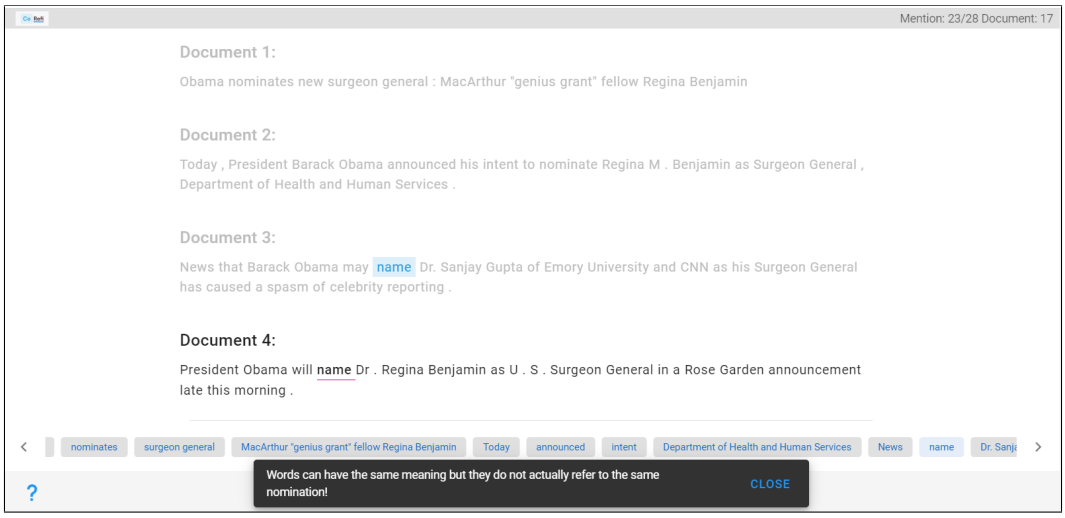

Figure 7: Example of an automatic feedback during the guided annotation.

The current mention refers to the naming of the Dr. Regina Benjamin as U.S Surgeon General where as the selected cluster refers to the event of naming Dr. Sanjay Gupta to Surgeon General. Since, the correct decision is subtle the user receives a toast informing them that Words can have the same meaning but not corefer. This toast helps to guide the annotator to the correct decision and reinforces the coreference guidelines. As the trainee is familiarized with the subtleties of coreference they are less likely to make similar mistakes during annotation of the real dataset.