Using Automatically Extracted Minimum Spans to Disentangle Coreference Evaluation from Boundary Detection

Nafise Sadat Moosavi[1] and **Leo Born**[2] and **Massimo Poesio**[3] and **Michael Strube**[4]
[1]Ubiquitous Knowledge Processing (UKP) Lab, Technische Universität Darmstadt
[2]Institute for Computational Linguistics, Heidelberg University
[3]School of Electronic Engineering and Computer Science, Queen Mary University of London
[4]Heidelberg Institute for Theoretical Studies gGmbH

Abstract

The common practice in coreference resolution is to identify and evaluate the maximum span of mentions. The use of maximum spans tangles coreference evaluation with the challenges of mention boundary detection like prepositional phrase attachment. To address this problem, minimum spans are manually annotated in smaller corpora. However, this additional annotation is costly and therefore, this solution does not scale to large corpora. In this paper, we propose the MINA algorithm for automatically extracting minimum spans to benefit from minimum span evaluation in all corpora. We show that the extracted minimum spans by MINA are consistent with those that are manually annotated by experts. Our experiments show that using minimum spans is in particular important in cross-dataset coreference evaluation, in which detected mention boundaries are noisier due to domain shift. We have integrated MINA into https://github.com/ns-moosavi/coval for reporting standard coreference scores based on both maximum and automatically detected minimum spans.

1 Introduction

Coreference resolution is the task of finding different expressions that refer to the same real-world entity. Each referring expression is called a mention. The common approach to annotate coreferring mentions is to specify the largest span of each mention. The problem with using maximum spans in coreference evaluation is that a single mention may have different maximum boundaries based on gold vs. automatically detected syntactic structures. For instance, variations in prepositional phrase attachment, which is a known challenge in syntactic parsing, will lead to different maximum boundaries for a single mention.

In order to decouple coreference evaluation from maximum boundary detection complexities, smaller corpora like MUC (Hirschman and Chinchor, 1997), ACE (Mitchell et al., 2002), and ARRAU (Uryupina et al., 2016) explicitly annotate the minimum span as well as the maximum logical span of each mention. The annotated minimum spans indicate the minimum strings that a coreference resolver must identify for the corresponding mentions. This solution comes with an additional annotation cost. As a result, the annotation of minimum spans has been discarded in larger corpora like CoNLL-2012 (Pradhan et al., 2012).

In this paper, we propose MINA, a MINimum span extraction Algorithm that automatically determines minimum spans from constituency-based parse trees. Based on our analyses, MINA spans are compatible with those that are manually annotated by experts. By using MINA, we can benefit from minimum span evaluation for all corpora without introducing additional annotation costs.

While the use of MINA spans already benefits in-domain evaluation, by reducing the gap between the performance on gold vs. system mentions, it has a more significant impact on cross-dataset evaluation, in which detected maximum mention boundaries are noisier due to domain shift.

Cross-dataset coreference evaluation is used to assess the generalization of coreference resolvers (Moosavi and Strube, 2017, 2018). Coreference resolution is a mid-step for text understanding in downstream tasks, e.g., question answering, text summarization, and information retrieval. Therefore, generalization is an important property for coreference resolvers because downstream datasets are not necessarily from the same domain as those of coreference-annotated corpora.

When coreference resolvers are applied to a new domain, detected maximum boundaries become noisier, e.g., gold and system mentions differ by

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4168–4178
Florence, Italy, July 28 - August 2, 2019.

the inclusion or exclusion of surrounding commas or quotation marks. Such noisy boundaries directly affect the coreference evaluation scores based on maximum spans. The use of minimum spans reduces the impact of such noises in coreference evaluation and results in more reliable comparisons between different coreference resolvers.

2 Boundary Mismatch Example

Example 1, and its corresponding gold and system parse trees in Figure 1 and Figure 2, respectively, show a sample boundary mismatch from the CoNLL-2012 development set. Based on the gold parse tree (Figure 1), "an extensive presence" is the maximum span of the first coreferring mention in Example 1. However, the corresponding maximum boundary for this same mention is "an extensive presence, of course in this country" based on the system parse tree (Figure 2).

Example 1 *This News Corp. has [an extensive presence]$_1$, of course in this country. [That presence]$_{(1)}$ may be expanding soon.*

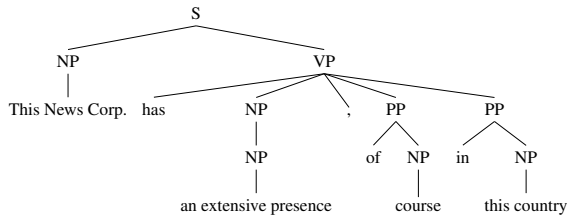

Figure 1: Gold parse tree of Example 1.

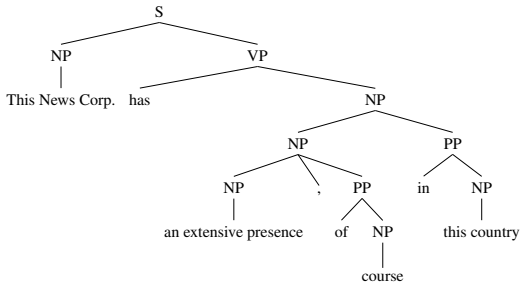

Figure 2: System parse tree of Example 1.

A system that uses the system parse tree for mention detection links "that presence" to "an extensive presence, of course in this country" and gets penalized based on recall and precision. This penalty is the same as that of a system that links "that presence" to "this News Corp.". Recall drops because of not recognizing "an extensive presence" and precision drops because of detecting a spurious mention.

3 Background

MINA is an attempt to decouple coreference evaluation from parsing errors to some extent. This motivation is the same as the one that resulted in the manual annotation of minimum spans in the MUC, ACE and ARRAU corpora. According to the MUC task definition,[1] the use of minimum spans in coreference evaluation is as follows:[2] Assume m_{max} and m_{min} are the annotated maximum and minimum spans for the mention m. The system mention $\hat{m}$ is equivalent to m if it includes m_{min} and it does not include any tokens beyond those that are included in m_{max}. This way of using minimum spans does not handle inconsistencies in gold vs. system mention boundaries in which system boundaries are larger than their corresponding gold boundaries, as it is the case for the mention "an extensive presence, of course in this country" in Example 1.

Compared to manually annotated minimum spans:

- MINA is applicable to any English coreference corpus.[3] In contrast, manually annotated minimum spans can be only used in their own corpora.
- For coreference evaluation, MINA extracts minimum spans for both gold and system mentions based on a single parse tree. Therefore, it can handle system-detected maximum spans that are either shorter or longer than their corresponding gold maximum span.

The coreference resolver of Peng et al. (2015) is developed around the idea that working with mention heads is more robust compared to working with maximum mention boundaries. In this regard, they develop a system that resolves coreference relations based on mention heads. The resolved mention heads are then expanded to full mention boundaries using a separate classifier that is trained to do so. Peng et al. (2015) also report the evaluation scores using both maximum mention boundaries and mention heads. Peng et al. (2015) extract mention heads using Collins' head finder rules (Collins, 1999). They use gold

[1] http://www-nlpir.nist.gov/related_projects/muc/proceedings/co_task.html

[2] The ARRAU dataset also follows this way of using minimum spans.

[3] We did not have the manually annotated minimum spans for coreference corpora of other languages in order to verify whether MINA is also applicable to them.

constituency-based parse trees and gold named entity information. The gold parse information is only used during training to train their mention head detection classifier. The gold named entity information is used to specify the whole span of named entities as their heads. The reason is that the head finding rules only specify one word as a head, and one-word heads can be troublesome for named entities, e.g., "Green" would be selected as the head of both "Mary Green" and "John Green".

In this paper, we also examine the use of head words as minimum spans. We show that compared to head words, `MINA` spans are more compatible with expert annotated minimum spans.

Since we evaluate minimum spans on various corpora, from which some do not include gold named entity information or even gold parse trees, we only use Collins' head finder rules, without the final adjustment for named entities, as the baseline for minimum span detection.

Collins' rules for finding the head of a noun phrase (NP) are as follows:

- If the tag of the last word is POS, return it as the head,
- else return the first child, from right to left, with an NN, NNP, NNPS, NNS, NX, POS, or JJR tag, if there is any,
- else return the first child, from left to right, with an NP tag, if there is any,
- else return the first child, from right to left, with one of the $, ADJP, or PRN tags, if there is any,
- else return the first child, from right to left, with a CD tag, if there is any,
- else return the first child, from right to left, with a JJ, JJS, RB, or QP tag, if there is any,
- else return the last word.

For the head finder rules for phrases other than NPs, please refer to Appendix A of Collins (1999).

4 How to Determine Minimum Spans?

We process the constituency-based parse trees of mentions, i.e., the parse sub-tree of their corresponding maximum span, in a breadth-first manner to determine minimum spans. Algorithm 1 outlines the minimum span extraction process. In this algorithm, `root` is the root of the mention's parse tree, `tags` is the set of acceptable syntactic tags for extracting minimum spans, `min-depth` is the depth of the minimum span nodes in the parse tree, and `min-spans` is the output of the algorithm that corresponds to the set of mention words that belong to the minimum span. `min-depth` is initially set to ∞, and `tags` and `min-spans` are empty.

Algorithm `MINA` (*root*)
  min-depth = ∞
  if *tags*=$\emptyset$ **then**
    if *root is an NP* **then**
      tags= {NP acceptable tags}
    else if *root is a VP* **then**
      tags={VP acceptable tags}
  Process $root$ in a breadth-first manner
  for *each processed node* n **do**
    if $n.tag \notin tags$ **then**
      skip processing n*'s children*
    else if n *is an acceptable terminal node* & $n.depth \leq$*min-depth* **then**
      min-spans.add(n)
      min-depth = *n.depth*

Algorithm 1: Extraction of minimum spans.

The set of acceptable terminal nodes in a parse tree are those that include at least one word other than a determiner[4] or a conjunction[5]. We do not further split terminal nodes, e.g., an acceptable terminal node may contain both a determiner as well as a noun. For extracting the minimum span of a noun phrase, the set of acceptable syntactic tags is {"NP" (noun phrase), "NML" (nominal modifier), "QP" (quantifier phrase used within NP), "NX" (used within certain complex NPs)}. For verb phrases, "VP" is the only acceptable tag.

`MINA` processes the parse tree in a breadth-first manner. It skips processing sub-trees that are rooted by a node whose syntactic tag is not acceptable, e.g., "PP". For the rest of the nodes, it extracts all acceptable terminal nodes that have the shortest distance to `root` as minimum spans.

For instance, in Figure 3, the root node is an NP and `tag` would be set to NP's acceptable tags. Therefore, among the children of the root, `MINA` would only process the child with an NP tag (the left child) and skip the one with the PP tag.

If the final minimum span is empty, e.g., if due to parsing errors the syntactic tag of none of the tree nodes is among the acceptable tags, we fall back to using the maximum span.[6]

[4] A word with the "DT" POS tag.

[5] A word with the "CC" POS tag.

[6] If we use gold parse trees, this happens for 14 mentions in the CoNLL-2012 development set from which ten are one-word mentions, e.g., "ours" is detected as "ADJP".

MINA extraction examples. Figures 3-6 show the MINA minimum spans of various noun phrases with different internal structures. The corresponding MINA spans of the parse trees are boldfaced.

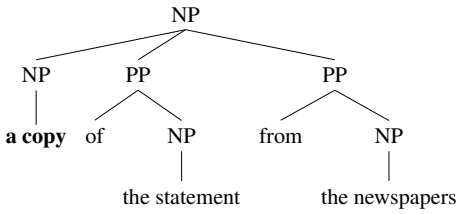

Figure 3: MINA span in an NP with the grammar form "NP $->$ NP PP PP". MINA span is boldfaced.

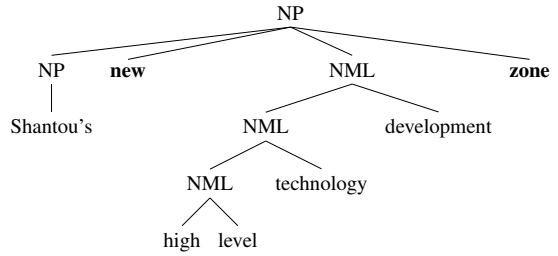

Figure 4: MINA spans in an NP with a nested structure.

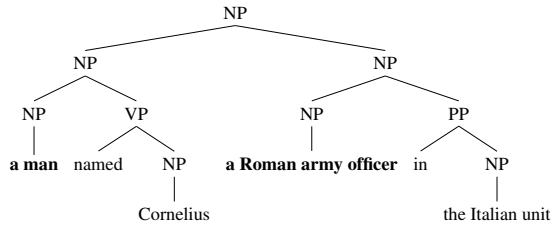

Figure 5: MINA spans in an appositive noun phrase.

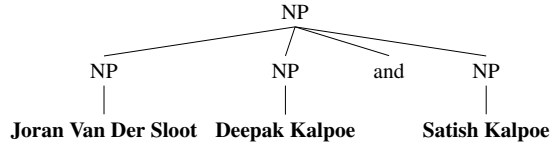

Figure 6: MINA spans in an NP with conjunction. Boldfaced minimum spans belong to a single mention.

Using MINA for coreference evaluation. For each coreference evaluation, we have a key file, including gold coreference annotations, and a system file, including predicted coreference outputs. For coreference evaluation using minimum spans, we use the provided parse trees in the key file.[7] Therefore, the minimum spans of both gold mentions and system mentions are determined based on the same parse tree. We then use minimum spans instead of maximum spans in all scoring metrics, i.e., a gold and a system mention are considered equivalent if they have the same minimum span.

The corresponding sub-trees of the discussed gold and system mentions of Example 1, based on the gold parse tree in Figure 1, are shown in Figure 7.[8] The MINA span of both of these two trees is "an extensive presence". Therefore, the gold coreference chain {"an extensive presence", "that presence"} and the system coreference chain {"an extensive presence, of course in this country", "that presence"} are equivalent if they are evaluated based on minimum spans.

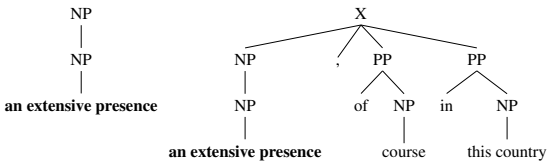

Figure 7: Mention trees of Example 1. The left and right sub-trees represent the boundaries of gold and system mentions, respectively.

5 Evaluating MINA Spans

In order to analyze the detected MINA spans, we evaluate the following two properties:

- Length of MINA spans: since we retrieve MINA spans from the corresponding parse tree of the mentions, MINA spans are always a subset of their corresponding maximum span words. However, on average, the length of minimum spans (number of containing words) should be smaller than that of maximum spans.
- Compatibility of automatically extracted minimum spans with those that are manually annotated by experts: we evaluate MINA spans against manually annotated minimum spans, called MIN, in the MUC and ARRAU corpora to examine whether the reduced spans still contain words of the mention that were deemed important by experts.

For the experiments of this section, we use the MUC-6, MUC-7, ARRAU, and CoNLL-2012

[7] If the key file does not include parse information, we parse it with the Stanford parser.

[8] If the boundary of a mention is not recognized as a single phrase in the parse tree, as it is the case for the system mention, we add a dummy root ("X" in the right subtree of Figure 7) to include the whole span into a single phrase.

corpora, from which MUC and ARRAU contain manually annotated minimum spans. We use the Stanford neural constituency parser (Socher et al., 2013) for getting system parse trees, unless otherwise stated. For the ARRAU corpus, we use mentions of the training split of the RST Discourse Treebank subpart.

As a baseline, we also evaluate the syntactic head of mentions, based on Collins' rules, as the minimum span.[9]

How does the length of evaluated spans change by using MINA? Table 1 shows the average length of maximum spans vs. that of MINA spans on the training splits of the MUC-6, MUC-7 and ARRAU corpora as well as the development set of the CoNLL-2012 dataset. For the CoNLL-2012 dataset, we use the provided gold parse information. We parse the MUC and ARRAU datasets, since the gold parse information is not available for these datasets.

Based on Collins' head finder rules, the detected head always includes one word.

	MUC-6	MUC-7	ARRAU	CoNLL
maximum span	5.2	5.3	3.8	2.4
MINA span	2.6	2.7	2.0	1.6

Table 1: The average length of MINA spans compared to that of maximum spans in the MUC, ARRAU, and CoNLL-2012 datasets.

Figure 8 shows the number of mentions with the length of one, two, three, and $\geq$4 based on both maximum and MINA spans on the CoNLL-2012 development set. The length of the maximum span of around 14% of mentions is longer than three, while this ratio is only 4% for MINA minimum spans. Mentions with long MINA spans include appositions or conjunctions, e.g., the MINA span in Figure 6.

Does MINA correlate with MIN? We evaluate MINA minimum spans against manually annotated minimum spans in the MUC and ARRAU corpora. The manually annotated minimum span in these corpora is referred to as MIN.

Table 2 shows the ratio of minimum spans that contain the corresponding MIN when the minimum span is extracted by MINA and the head finding rules. As we can see, MINA contains MIN in the majority of the mentions, and therefore, it is compatible with what experts would consider as the most important part of the mentions.

[9]We use the implementation of the head-finding rules that is available at https://github.com/smartschat/cort/.

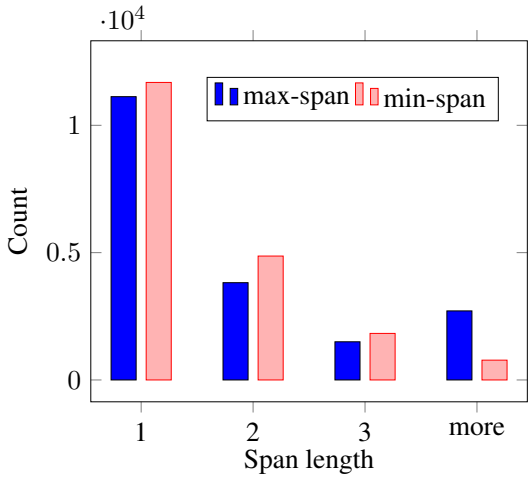

Figure 8: Span length based on maximum vs. MINA's minimum spans on the CoNLL-2012 development set.

	MUC-6	MUC-7	ARRAU
MINA	96.2	93.1	98.3
head	94.0	91.1	93.9

Table 2: Ratio of detected MINA and head words which contain the corresponding MIN annotations in the MUC and ARRAU corpora. The same parse information is used for detecting both MINA and head words. Datasets are parsed using the Stanford neural parser.

Figure 9 shows an example from ARRAU in which MINA contains MIN but the head does not.

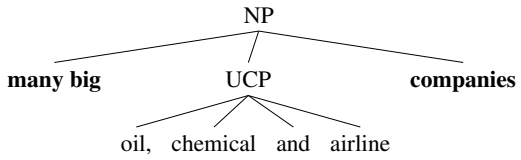

Figure 9: System parse tree of a mention from ARRAU. MINA spans are boldfaced. "many" and "companies" are the corresponding MIN and head, respectively.

MINA and MIN inconsistencies, i.e., cases in which MINA does not contain MIN, are mainly due to parsing errors. Figure 10 and Figure 11 show two examples from the MUC and ARRAU datasets in which MINA selects an incorrect minimum span because of an incorrect parse tree.

Figure 12 shows two sample mismatch examples between MINA and MIN from the ARRAU

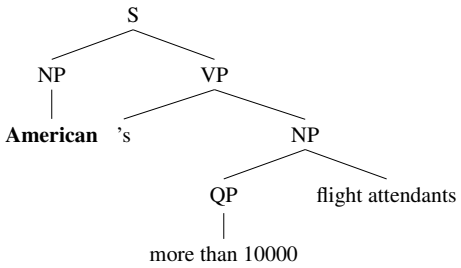

Figure 10: The system parse tree of a mention from MUC-6. MINA spans are boldfaced ("American"). "attendants" is the annotated MIN.

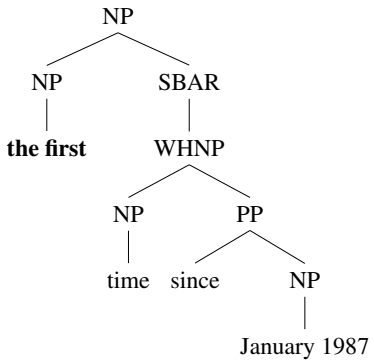

Figure 11: The system parse tree of a mention from ARRAU. MINA spans are boldfaced ("the first"). "time" is the annotated MIN.

dataset, in which the mismatch is not due to parsing errors.

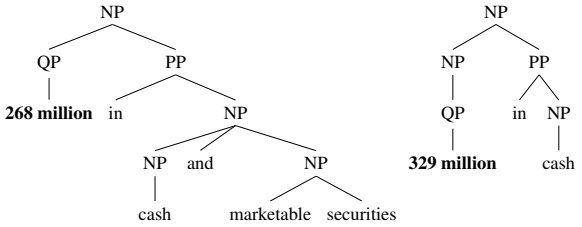

Figure 12: The system parse trees of two mentions from ARRAU. MINA spans are boldfaced. "securities" and "cash" are annotated as MIN for the left and right mentions, respectively.

In order to investigate the effect of using a different parser, we perform the experiment of Table 2 using the Stanford English PCFG parser (Klein and Manning, 2003). The results are reported in Table 3. As we see, the use of a better parser, i.e., the Stanford neural parser, makes MINA spans, as well as detected heads, more consistent compared to MIN spans.

In addition to the above two properties, i.e. the length of minimum spans and their consistency with MIN annotations, we also check that MINA returns *different minimum spans for distinct overlapping mentions*.

	MUC-6	MUC-7	ARRAU
MINA	95.6	92.4	98.1
head	92.9	90.0	93.4

Table 3: Ratio of the detected MINA and head words that contain their corresponding MIN annotations in MUC and ARRAU. MINA and head words are detected using the parse trees of the Stanford PCFG parser.

As an example, the minimum span of the mention "John and Mary" should be different from those of "John" and "Mary", because they all refer to different entities. In this regard, we examine all overlapping coreferent mentions in the CoNLL-2012 English development set, from which none of the overlapping mentions has the same MINA span. However, this is not the case for heads.

6 Effect on Coreference Evaluation

6.1 Experimental Setup

In this section, we investigate how the use of minimum spans instead of maximum spans in coreference evaluation affects the results in in-domain as well as cross-dataset evaluations. For comparisons, we use the *CoNLL* score (Pradhan et al., 2014), i.e. the average F_1 value of *MUC* (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and $CEAF_e$ (Luo, 2005), and the *LEA* F_1 (Moosavi and Strube, 2016) score.[10] Minimum spans are detected using both MINA and Collins' head finding rules. All examined coreference resolvers are trained on the CoNLL-2012 training data. For in-domain evaluations, models are evaluated on the CoNLL-2012 test data and minimum spans are extracted using gold parse trees, which are provided in CoNLL-2012.[11]

For cross-dataset evaluations, models are tested on the WikiCoref dataset (Ghaddar and Langlais, 2016). For extracting minimum spans, we parse WikiCoref by the Stanford neural parser. This dataset is annotated using the same annotation guidelines as that of CoNLL-2012, however, it contains documents from a different domain.

[10] We use the python implementation that is available at https://github.com/ns-moosavi/coval.

[11] We also examined the in-domain results of Table 4 based on the system parse trees of CoNLL-2012 instead of gold parse trees. The differences between scores based on MINA spans that are extracted from gold vs. those that are extracted from system parse trees were only about 0.2 points.

	CoNLL			LEA		
	max	MINA	head	max	MINA	head
			CoNLL-2012 test set			
Stanford rule-based	55.60 (8)	57.55 (8)	57.38 (8)	47.31 (8)	49.65 (8)	49.44 (8)
cort	63.03 (7)	64.60 (6)	64.51 (6)	56.10 (6)	58.05 (6)	57.93 (6)
Peng et al.	63.05 (6)	63.50 (7)	63.54 (7)	55.22 (7)	55.76 (7)	55.80 (7)
deep-coref ranking	65.59 (5)	67.29 (5)	67.09 (5)	59.58 (5)	61.70 (5)	61.43 (5)
deep-coref RL	65.81 (4)	67.50 (4)	67.36 (4)	59.76 (4)	61.84 (4)	61.64 (4)
Lee et al. 2017 single	67.23 (3)	68.55 (3)	68.53 (3)	61.24 (3)	62.87 (3)	62.82 (3)
Lee et al. 2017 ensemble	68.87 (2)	70.12 (2)	70.05 (2)	63.19 (2)	64.76 (2)	64.64 (2)
Lee et al. 2018	72.96 (1)	74.26 (1)	75.29 (1)	67.73 (1)	69.32 (1)	70.40 (1)
			WikiCoref			
Stanford rule-based	51.78 (4)	53.79 (5)	57.10 (4)	43.28 (5)	45.48 (6)	49.28 (4)
deep-coref ranking	52.90 (3)	55.16 (2)	57.13 (3)	44.40 (3)	46.98 (3)	49.05 (5)
deep-coref RL	50.73 (5)	54.26 (4)	57.16 (2)	41.98 (6)	46.02 (4)	49.29 (3)
Lee et al. 2017 single	50.38 (6)	52.16 (6)	54.02 (6)	43.86 (4)	45.75 (5)	47.69 (6)
Lee et al. 2017 ensemble	53.63 (2)	55.03 (3)	56.80 (5)	47.50 (2)	48.98 (2)	50.87 (2)
Lee et al. 2018	57.89 (1)	59.90 (1)	61.33 (1)	52.42 (1)	54.63 (1)	56.19 (1)

Table 4: Evaluations based on maximum span, MINA, and head spans on the CoNLL-2012 test set and WikiCoref. The ranking of corresponding scores is specified in parentheses. Rankings which are different based on maximum vs. MINA spans are highlighted.

CoNLL-2012 contains the newswire, broadcast news, broadcast conversation, telephone conversation, magazine, weblogs, and Bible genres while the annotated documents in WikiCoref are selected from Wikipedia.

6.2 Results

Table 4 shows the maximum vs. minimum span evaluations of several recent coreference resolvers on the CoNLL-2012 test set and the WikiCoref dataset. The examined coreference resolvers are as follows: the Stanford rule-based system (Lee et al., 2013), the coreference resolver of Peng et al. (2015), the ranking model of `cort` (Martschat and Strube, 2015), the ranking and reinforcement learning models of `deep-coref` (Clark and Manning, 2016a,b), the single and ensemble models of Lee et al. (2017), and the current state-of-the-art system by Lee et al. (2018).

We make the following observations based on the results of Table 4:

Using minimum spans in coreference evaluation strongly affects the comparisons in the cross-dataset setting. The results on the WikiCoref dataset show that mention boundary detection errors specifically affect coreference scores in cross-dataset evaluations. The ranking of systems is very different by using maximum vs. minimum spans. The reinforcement learning model of `deep-coref`, i.e., `deep-coref RL`, has the most significant difference when it is evaluated based on maximum vs. minimum spans (about 4 points). The ensemble model of `e2e-coref`, on the other hand, has the least difference between maximum and minimum span scores (1.4 points), which indicates it better recognizes maximum span boundaries in out-of-domain data.

Using minimum spans in coreference evaluation reduces the gap between the performance on gold vs. system mentions. It is shown that there is a large gap between the performance of a coreference resolver on gold vs. system mentions, see e.g., Peng et al. (2015). The use of minimum spans in coreference evaluation reduces this gap by about two points. The comparison of the results of different systems on gold and system mentions using both maximum and minimum spans are included in Appendix A.

Evaluation based on minimum spans reduces the differences that are merely due to better maximum boundary detection. The coreference resolver of Peng et al. (2015) has the smallest difference between its maximum and minimum span evaluation scores. This result indicates the superiority of Peng et al. (2015)'s mention

boundary detection method compared to other approaches.[12] Based on maximum spans, Peng et al. (2015) performs on-par with `cort` while `cort` outperforms it by about one percent when they are evaluated based on minimum spans. Therefore, the use of minimum spans in coreference evaluation decreases the effect of mention boundary detection errors in coreference evaluation and results in fairer comparisons.

7 Analysis

In order to better understand the impact of using minimum spans in cross-dataset evaluations, we analyze the output of `deep-coref RL`, on which minimum span evaluation has the largest impact, for the cases in which a system mention and its corresponding gold mention have the same minimum span while they have different maximum boundaries.

We have included some examples from these mismatches in Example 2–Example 6. The boundaries of gold and system mentions are determined by g and s indices, respectively. Mismatching spans are boldfaced in all examples.

We observe that the majority of the mismatches are due to (1) incorrect detection of appositive relation (Example 2), (2) mismatch as a result of not including a surrounding quotation (Example 5), and (3) inclusion of an additional comma at the end of the mention (Example 3).

Example 2 *Canada is noted for having a positive relationship with [[the Netherlands]$_g$* ***owing,***$]_s$ *in part, to its contribution to the Dutch liberation during World War II.*

Example 3 *.[[Le Courrier du Sud]$_g$***,**$]_s$ *published by Quebecor Media, is the oldest, and contains inserts tailored to specific boroughs*

Example 4 *in 2007, [[Pierce College]$_g$* ***sheltered***$]_s$ *and fed more than 150 horses under the direction of the L.A. County Volunteer Equine Response team.*

Example 5 *Prime Minister Brian Mulroney's Progressive Conservatives abolished the NEP and changed the name of FIRA to [**"**[Investment Canada]$_s$**"**]$_g$, to encourage foreign investment.*

Example 6 *In 2011, [$_s$***nearly 6.8 million** *[$_g$Canadians]] listed a non-official language as their mother tongue.*

[12] It has a separate classifier for detecting maximum boundaries based on mention heads.

8 Conclusions

Coreference evaluation based on maximum spans directly penalizes coreference resolvers because of parsing complexities and also small noises in mention boundary detection. This is a known problem that is addressed by manually annotating minimum spans as well as maximum spans in several corpora. Minimum span annotation is expensive, and therefore it is not a scalable solution for large coreference corpora. In this paper, we propose the `MINA` algorithm to automatically extract minimum spans without introducing additional annotation costs. `MINA` automatically extracts corresponding minimum spans for both gold and system mentions and uses the resulting minimum spans in the standard evaluation metrics. Based on our analysis on the MUC and ARRAU datasets, extracted minimum spans are compatible with those that are manually annotated by experts. The incorporation of automatically extracted minimum spans reduces the effect of maximum boundary detection errors in coreference evaluation and results in a fairer comparison. Our results show that the use of minimum spans in coreference evaluation is of particular importance for cross-dataset settings, in which the detected maximum boundaries are noisier.

In addition to coreference evaluation, automatically extracted minimum spans can benefit the annotation process of new corpora. If we provide automatically extracted minimum spans alongside maximum spans to the annotators, the annotation of coreference relations may get easier. For instance, detecting the coreference relation of the two nested mentions in "[a deutsche mark based currency board where we have a foreign governor on [the board]$_{(1)}$]$_{(1)}$"[13] would be more straightforward knowing that the minimum span of the first mention is "a currency board".

A future direction is to investigate the effect of using `MINA` spans not only in evaluation but also for training existing coreference resolvers. Maximum spans are recoverable given the `MINA` spans and their corresponding parse trees. Therefore, we can use `MINA` spans for training and testing coreference models and then retrieve their corresponding maximum spans for evaluation. Investigating the use of `MINA` in other NLP areas, e.g., evaluating spans in named entity recognition or reading comprehension, is another future line of work.

[13] Taken from the CoNLL-2012 development set.

Acknowledgments

The authors would like to thank Mark-Christoph Müller, Ilia Kuznetsov and the anonymous reviewers for their valuable comments and feedbacks. This work has been supported by the Klaus Tschira Foundation, Heidelberg, Germany, the German Research Foundation (DFG) as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1), and the DFG-funded research training group Adaptive Preparation of Information form Heterogeneous Sources (AIPHES, GRK 1994/1).

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation,* Granada, Spain, 28–30 May 1998, pages 563–566.

Kevin Clark and Christopher D. Manning. 2016a. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016b. Deep reinforcement learning for mention-ranking coreference models. pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, Penn.

Abbas Ghaddar and Philippe Langlais. 2016. WikiCoref: An English coreference-annotated corpus of Wikipedia articles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation,* Portorož, Slovenia, 23–28 May 2016.

Lynette Hirschman and Nancy Chinchor. 1997. MUC-7 coreference task definition, `http://www.muc.saic.com/proceedings/`.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418.

Alexis Mitchell, Stephanie Strassel, Mark Przybocki, JK Davis, George Doddington, Ralph Grishman, Adam Meyers, Ada Brunstain, Lisa Ferro, and Beth Sundheim. 2002. ACE-2 Version 1.0. LDC2003T11, Philadelphia, Penn.: Linguistic Data Consortium.

Nafise Sadat Moosavi and Michael Strube. 2016. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Berlin, Germany. Association for Computational Linguistics.

Nafise Sadat Moosavi and Michael Strube. 2017. Lexical features in coreference resolution: To be used with caution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Vancouver, Canada. Association for Computational Linguistics.

Nafise Sadat Moosavi and Michael Strube. 2018. Using linguistic features to improve the generalization capability of neural coreference resolvers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Brussels, Belgium. Association for Computational Linguistics.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A joint framework for coreference resolution and mention head detection. In *Proceedings of the 19th Conference on Computational Natural Language Learning,* Beijing, China, 30–31 July 2015, pages 12–21.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. 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)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria. Association for Computational Linguistics.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Kepa Rodriguez, and Massimo Poesio. 2016. ARRAU: Linguistically-motivated annotation of anaphoric descriptions. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pages 45–52, San Mateo, Cal. Morgan Kaufmann.

A Appendix

Table 5 shows *CoNLL* scores and the *LEA* F_1 values of the participating systems in the CoNLL-2012 shared task (closed task with predicted syntax and mentions) based on both maximum and minimum span evaluations. Minimum spans are detected using both `MINA` and Collins' head finding rules using gold parse trees.

The corresponding results using gold mentions (system used gold mentions to resolve coreference relation), are given in Table 6.

Based on the results of Tables 5 and 6: (1) the use of minimum spans reduces the gap between the performance on gold vs. system mentions by about two percent, (2) the use of minimum instead of maximum spans results in a different ordering for some of the coreference resolvers, and (3) when gold mentions are used, there are no boundary detection errors, and consequently the results using `MINA` are the same as those of using maximum spans. Due to recognizing the same head for distinct overlapping mentions, the scores using the head of gold mentions are not the same as using their maximum span, which in turn indicates `MINA` is suited better for detecting minimum spans compared to head words.

	CoNLL			LEA		
	max	MINA	head	max	MINA	head
fernandes	60.6 (1)	62.2 (1)	63.9	53.3	55.1	57.0
martschat	57.7 (2)	59.7 (2)	61.0	50.0	52.4	53.9
bjorkelund	57.4 (3)	58.9 (3)	60.7	50.0	51.6	53.6
chang	56.1 (4)	58.0 (4)	59.6	48.5	50.7	52.5
chen	54.5 (5)	56.5 (5)	58.2	46.2	48.6	50.4
chunyuang	54.2 (6)	56.1 (6)	57.9	45.8	48.1	50.2
shou	53.0 (7)	54.8 (8)	56.5	44.0	46.1	48.1
yuan	52.9 (8)	54.9 (7)	56.7	44.8	47.0	48.9
xu	52.6 (9)	53.9 (9)	55.2	46.8	48.4	50.0
uryupina	50.0 (10)	51.0 (11)	52.4	41.2	42.3	43.7
songyang	49.4 (11)	51.3 (10)	52.9	41.3	43.5	45.3

Table 5: CoNLL-2012 shared task systems evaluations based on maximum spans, MINA spans, and head words. The rankings based on the CoNLL scores are included in parentheses for maximum and MINA spans. Rankings which are different based on maximum vs. MINA spans are highlighted.

	CoNLL			LEA		
	max	MINA	head	max	MINA	head
fernandes	69.4	69.4	69.8	56.1	56.1	56.1
bjorkelund	68.0	68.0	68.1	61.1	61.1	61.1
chang	77.2	77.2	77.2	67.9	67.9	67.6
chen	71.3	71.3	71.4	63.9	63.9	63.9
yuan	70.4	70.4	70.4	63.4	63.4	63.4
xu	61.0	61.0	61.2	56.9	56.9	57.1

Table 6: CoNLL-2012 shared task systems evaluations using gold mentions.