Landcore: Coreference Resolution with Language-Specific LLM-Enhanced Prompts and XML-Inspired Annotation Scheme

Jan Pavelka
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics (ÚFAL)
Prague, Czechia
jan.pavelka.mm@gmail.com

Abstract

This paper presents *Landcore*,[1] our submission to the LLM Track of the CRAC 2026 Shared Task on Multilingual Coreference Resolution. We explore the capabilities of LLMs in coreference resolution across multiple languages and domains, using a few-shot prompting approach. We design a comprehensive prompt that includes detailed instructions and examples and further enhance it using an LLM to produce language-specific prompts. We present an XML-inspired annotation scheme that is more suitable for LLMs than the provided formats. Although our solution is not the best-performing, we show that our ideas improve performance across various settings.

1 Introduction

Coreference is the phenomenon that multiple phrases refer to the same real-world entity (e.g., *Bangladesh's capital city*, *Dhaka*, *the city*, *it*).

CRAC 2026[2] is the fifth edition of the shared task on multilingual coreference resolution (Novák et al., 2026). It features two tracks: the *unconstrained track* and the *LLM track*, which is limited to systems based on large language models. We participated in the LLM track. Our main goal was to test how far carefully designed prompts and few-shot examples can take us without fine-tuning.

The shared-task data (27 corpora in 19 languages) are taken from CorefUD 1.4. They span both contemporary and historical texts and domains such as news, literature, and web data. This year, several corpora with very long documents were added, posing a challenge for LLMs with limited context windows.

2 Main Ideas

We chose an in-context learning approach to assess how far carefully designed prompts with tailored instructions and few-shot examples can take us. We explore an XML-inspired annotation scheme and LLM-enhanced, language-specific prompts.

2.1 Annotation Schemes

The data for the LLM track of the CRAC 2026 are provided in two formats: CoNLL-U and plain-text.

In CoNLL-U format,[3] coreference annotation is stored in the 'Entity' attribute of the 'MISC' column. The plain-text format, designed specifically for the LLM track, includes inline markers denoting the start and end of mention spans. Linguistic tokenization is preserved, ensuring a 1:1 word correspondence between the two formats.

Beyond the provided formats, we also implemented the annotation scheme proposed by Seminck et al. (2025). We refer to this XML-inspired scheme as *EML* (*entity markup language*) format.

Figure 1 shows the comparison of the plain-text and EML formats. In contrast to the plain-text format, the EML scheme has the following properties:

1. Proper nesting: The last-opened span should be closed first. In the plain-text format, the tags are ordered by the entity ID number as they are in the CoNLL-U format.

2. The opening tags go before the first word of the mention. The whole mention is thus wrapped by the opening and the closing tag.

3. Each mention is annotated with two tags: opening <eX> and closing </eX>. In the plain-text format, single-word mentions use a single tag [eX], opening uses [eX and closing eX].

4. With little effort (such as adding a header), EML could be transformed into valid XML.

Both formats are fully equivalent and can be converted into each other without information loss. We

[1] LANguage Dependent COference REsolution
[2] https://ufal.mff.cuni.cz/corefud/crac26
[3] https://universaldependencies.org/format.html

Proceedings of the 2nd Joint Workshop on Computational Approaches to Discourse, Context and Document-Level Inferences and Computational Models of Reference, Anaphora and Coreference (CODI-CRAC 2026), pages 184–192

(a) the streets of Bangladesh 's capital city , Dhaka
(b) the|[e1 streets of Bangladesh|[e2,[e3 's|e2] capital city|e3] , Dhaka|[e3],e1]
(c) <e1>the streets of <e3><e2>Bangladesh 's</e2> capital city</e3> , <e3>Dhaka</e3></e1>

(d)

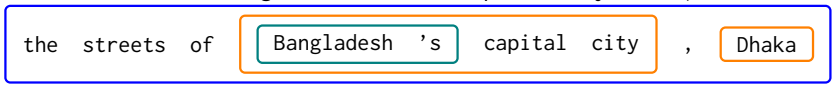

Figure 1: Comparison of the annotation schemes: (a) input blind text, (b) provided plain-text format, (c) our EML format, (d) visualization of entity mentions.

believe that EML is easier for LLMs to understand and generate, as much of their training data from the Internet is XML-based.

The organizers released the `text2text_coref`[4] script collection for converting CoNLL-U files to plain text and back. It also includes a cleaner that attempts to repair LLM outputs so they can be converted back to CoNLL-U. Without this cleaning, the conversion may fail. We implemented modified versions of these scripts for the EML format.

Our EML cleaner consists of three steps. First, it completes incomplete tags (<e1 → <e1>) and fixes spacing between tags and words:

<e1> mention </e1><e2>another</e2>
→ <e1>mention</e1> <e2>another</e2>

Second, it ensures that all mentions are properly opened and closed. Third, it aligns words between the input and annotated output using an edit-distance-like algorithm that preserves as many annotated words as possible. For the latter two steps, we adapt the scripts provided for plain text.[5]

2.2 Prompts

We introduce two prompt variants: hand-written language-universal prompts and LLM-enhanced, language-specific prompts. Figure 2 shows the structure of the universal prompt. It consists of instructions and examples. Empty word instructions are included only for corpora containing zero mentions. The task description is based on our linguistic understanding of coreference resolution and observations from development. The input format instructions describe the linguistic tokenization (UD style, where multi-word tokens are split, e.g., *don't* → *do n't*) and the structure of the input text. The output format instructions specify the target annotation scheme (EML or plain text). During development, recall in the metrics underlying the official score was consistently lower than precision, so we added instructions to favor recall. The

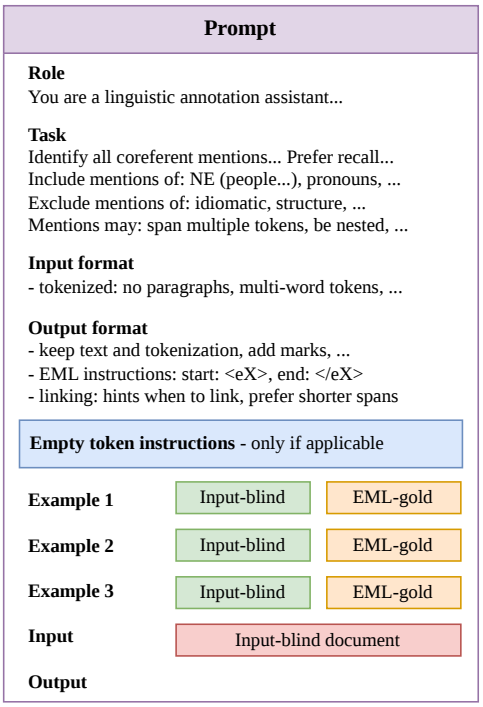

Figure 2: Universal prompt template structure.

prompt grammar and wording were refined with the help of GitHub Copilot (GPT-5.3-Codex). For the full universal, see Appendix A.1.

The universal prompt was given to GitHub Copilot chat (GPT-5.3-Codex), together with gold training data for each language, to adapt it to that language language (see Figure 3). The resulting LLM-enhanced prompts include a language-specific guidance section covering phenomena the annotation should focus on, such as common pronouns, morphological agreement, discourse relations, and domain-specific observations (e.g., dialogue in journalistic texts). Copilot also slightly revises other prompt sections that may appear language-independent. More details, including prompt examples, are provided in Appendices A.2 and A.3.

3 System Description

The final submission uses the EML annotation scheme and the LLM-enhanced language-specific

[4] https://github.com/ondfa/text2text-coref
[5] https://github.com/ondfa/text2text-coref/blob/master/src/text2text_coref/output_cleaner.py

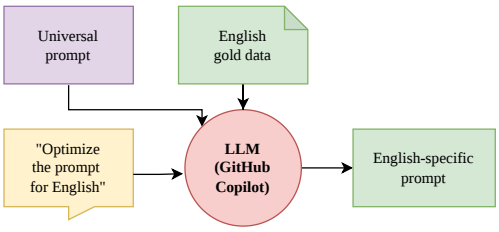

Figure 3: Generating language-specific prompts. This was done for each language.

prompts described in Section 2. This section describes other details of our final submission settings. The code is publicly available on GitHub.[6]

3.1 Few-Shot Examples

From the gold training data, we set aside some documents for development and validation. The remaining documents are used to select examples for the few-shot prompt. These documents are split into continuous chunks of at most 500 words while preserving sentence boundaries. We select the three longest chunks as examples, yielding a total example length of about 1500 words.[7] Examples are selected separately for each corpus to capture language- and corpus-specific style.

3.2 Input Documents

Since coreference chains do not cross document boundaries, each document can be processed independently. Each prompt contains a single input document, some of them very long, so we use LLMs with a large context window to process the entire document at once. Chunking documents into smaller parts of at most 1500 words, matching the total length of the examples, did not improve results (see Section 5).

3.3 Model and Inference

The final submission was generated with Claude Haiku 4.5 via the OpenRouter API[8], using temperature 0 and a 200,000-token input context window. The total cost of the LLM calls was about 20 USD.

Short documents were processed in tens of seconds, while the longest ones (nl_openboek, fr_litbank) took up to a few minutes. If the response did not contain valid output, i.e., it was empty, lacked EML tags, or was shorter than the input, we used the input text as a backup output and saved its index. All failed documents were then reprocessed with the same configuration. Processing all documents successfully required two iterations.

All test set documents were processed in parallel and finished in 10 minutes and 4 seconds. 22 of the 27 corpora were completed in under 2 minutes.

3.4 Postprocessing

The LLM outputs were cleaned and converted back to the CoNLL-U format using our modified `text2text_coref` scripts (see section 2.1).

After conversion, some fixes using Udapi blocks[9] (Popel et al., 2017) were applied. `corefud.MergeSameSpan` was used to merge mentions with the same span and the interleaved mentions were fixed with `corefud.FixInterleaved same_entity_only=False`. The entity IDs were prefixed with the document IDs using `corefud.FixEntityAcrossNewdoc`, ensuring that all entities in the corpus have unique IDs.

4 Results and Discussion

Among the four submissions to the LLM track, ours ranked fourth, with an official head-match CoNLL F1 score of 46.19%. Although this is substantially below the other submissions (74.32%, 73.83%, and 68.69%),[10] Table 1 shows that we outperformed the unconstrained-track baseline in five languages, especially historical ones (cu_proiel, grc_proiel, hbo_ptnk, la_coreflat). Results on nl_openboek, hu_korkor, and fr_litbank were comparable.

Compared with last year's in-context learning approaches, our solution outperformed the second-best NUST-FewShot (Sajid et al., 2025) on de_potsdamcc and the third-best PUXCRAC2025 (Phuc and Thin, 2025) on hbo_ptnk and ko_ecmt. On the other corpora, Landcore performs worse. A likely key weakness of our system is its example selection strategy. NUST-FewShot incorporates a large number of examples in the prompt, and PUXCRAC2025 applies a custom difficulty metric to select suitable examples, whereas Landcore simply uses three longest chunks.

On the development set, our system achieved 73.4% precision but only 49.7% recall in MOR (mention overlap ratio) which ignores mention *clustering* (Žabokrtský et al., 2022), suggesting that the

[6] https://github.com/pavlk-mm/landcore/releases/tag/crac2026

[7] Except for nl_openboek, where Claude Haiku struggles to generate output, likely because the documents in this dataset are too long; in those cases, we use only one example.

[8] https://openrouter.ai/anthropic/claude-haiku-4.5

[9] https://github.com/udapi/udapi-python

[10] https://www.codabench.org/competitions/13198/#/results-tab

Split	Model	cu_pro	grc_pro	hbo_ptnk	la_cor	hu_kor	fr_lit	nl_open	de_pot	ko_ecmt	...	Average
Dev	Landcore (LLM track)	**27.9**	**35.3**	**65.3**	**20.6**	36.3	**46.3**	**47.1**	50.5	59.0	...	47.0
	Baseline (Uncon. track)	24.3	30.2	24.8	8.6	**46.7**	43.1	40.8	**58.2**	**62.2**	...	**54.8**
Test	Landcore (LLM track)	29.4	43.4	61.4	**32.8**	41.6	**46.7**	39.3	49.3	59.5	...	46.2
	Baseline (Uncon. track)	24.7	30.6	31.7	6.8	42.2	46.1	**40.6**	52.4	65.0	...	54.5
	NUST-FewShot (LY)	**58.5**	**57.9**	**80.2**	—	43.5	—	—	48.7	**66.0**	...	**61.7***
	PUXCRAC2025 (LY)	43.7	47.9	45.3	—	**50.6**	—	—	**57.4**	50.3	...	60.1*

Table 1: Scores on corpora where our system outperforms or matches the official baseline for the Unconstrained track. The scores are taken from the Codabench leaderboards. The highest scores are shown in bold. For comparison, we also include last year's in-context learning systems, NUST-FewShot and PUXCRAC2025 (Novák et al., 2025). *) Last year's systems were evaluated on CorefUD 1.3 so the average is not comparable since new datasets were added to CorefUD 1.4. However, the plain texts of the six selected corpora are identical in CorefUD 1.3 and 1.4.

Configuration		EML	Plain text
Final (3 ex., temp=0)		47.0	41.3
No lang.-spec. prompt		46.3 (−0.7)	38.3 (−3.0)
≤1500-word chunks		45.3 (−1.7)	37.8 (−3.5)
Examples	1	44.4 (−2.6)	39.8 (−1.5)
	0	28.8 (−18.2)	2.7 (−38.6)
Temperature	0.25	49.5 (+2.5)	42.7 (+1.4)
	0.5	48.0 (+1.0)	40.7 (−0.6)
	0.75	45.7 (−1.3)	41.9 (+0.6)
	1.0	39.9 (−7.1)	37.0 (−4.3)

Table 2: Ablation studies comparing changes to the final configuration. The final configuration uses LLM-enhanced, language-specific prompts, full input documents, three examples, and temperature 0. Each subsequent row shows the effect of changing a single hyperparameter while keeping all others fixed. Differences from the final score of each format are given in parentheses.

main bottleneck of our system may be mention *identification* recall.

5 Ablation Analysis

Due to the cost of Haiku, we conducted the ablation analysis with DeepSeek V3.2,[11] which is about five times cheaper while performing comparably. On the development set, DeepSeek was even slightly better (47.03% vs. 46.95%). Although results vary across corpora, the overall score is similar.

We performed ablation analyses along several dimensions: *prompt type* (language-specific vs. universal), *document segmentation* (full documents vs. 1500-word chunks), *number of examples* (3, 1, and 0), and *temperature* (0.0, 0.25, 0.5, 0.75, and 1.0). Each dimension was tested separately on both annotation schemes while keeping the remaining settings fixed. Results are shown in Table 2.

Language-specific prompts outperform the universal prompt, especially with the plain-text format. Chunking documents did not improve results, confirming that the context window is sufficient even for long documents. The number of examples has a significant effect: three examples perform much better than one. Zero-shot fails completely on plain text. Setting the temperature to 1.0, DeepSeek's default, sharply reduces performance. In post-submission experiments, temperatures of 0.25 and 0.5 slightly improved the score.

EML outperforms plain text by a large margin in all settings, especially in zero-shot, supporting our hypothesis that EML is easier for LLMs to understand and generate.

6 Conclusion

In this paper, we presented our solution to the LLM track of the CRAC 2026 Shared Task on Multilingual Coreference Resolution. Our approach is based on in-context learning with carefully designed LLM-enhanced language-specific prompts and a novel annotation scheme called EML. Ablation studies show that EML in particular improves performance over the plain-text format.

Our ideas can be combined with other in-context learning techniques to potentially drive further performance gains. One clear direction for future work is the integration of smart example selection strategies and employing a better system for mention identification. Furthermore, the EML annotation scheme is not limited to the few-shot setting and can be used in any LLM-based solution, including fine-tuning or agentic systems.

To illustrate a possible application of automatic coreference resolution, Appendix B presents several experiments on data from the CZDEMOS4AI project.

Acknowledgements

This work was supported by the project TQ12000040 (CZDEMOS4AI) financed by

[11] https://openrouter.ai/deepseek/deepseek-v3.2

the Technology Agency of the Czech Republic (www.tacr.cz) within the Sigma 3 Program. This work has been using data and tools provided by the LINDAT/CLARIAH-CZ Research Infrastructure (lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062).

References

Michal Novák, Miloslav Konopik, Anna Nedoluzhko, Martin Popel, Ondrej Prazak, Jakub Sido, Milan Straka, Zdeněk Žabokrtský, and Daniel Zeman. 2025. Findings of the fourth shared task on multilingual coreference resolution: Can LLMs dethrone traditional approaches? In *Proceedings of the Eighth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 95–118, Suzhou, China. Association for Computational Linguistics.

Michal Novák, Miloslav Konopik, Anna Nedoluzhko, Martin Popel, Ondrej Prazak, Jakub Sido, Milan Straka, Zdeněk Žabokrtský, and Daniel Zeman. 2026. Findings of the fifth shared task on multilingual coreference resolution: Expanding datasets for long-range entities. In *Proceedings of the 2nd Joint Workshop on Computational Approaches to Discourse, Context and Document-Level Inferences and Computational Models of Reference, Anaphora and Coreference (CODI-CRAC 2026)*, San Diego, California, USA. Association for Computational Linguistics.

Nguyen Xuan Phuc and Dang Van Thin. 2025. Few-shot coreference resolution with semantic difficulty metrics and in-context learning. In *Proceedings of the Eighth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 149–153, Suzhou, China. Association for Computational Linguistics.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101, Gothenburg, Sweden. Association for Computational Linguistics.

Moiz Sajid, Seemab Latif, Zuhair Zafar, and Muhammad Moazam Fraz. 2025. Few-shot multilingual coreference resolution using long-context large language models. In *Proceedings of the Eighth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 154–162, Suzhou, China. Association for Computational Linguistics.

Olga Seminck, Antoine Bourgois, Yoann Dupont, Mathieu Dehouck, and Marine Delaborde. 2025. GLaRef@CRAC2025: Should we transform coreference resolution into a text generation task? In *Proceedings of the Eighth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 119–129, Suzhou, China. Association for Computational Linguistics.

Zdeněk Žabokrtský, Miloslav Konopík, Anna Nedoluzhko, Michal Novák, Maciej Ogrodniczuk, Martin Popel, Ondřej Pražák, Jakub Sido, Daniel Zeman, and Yilun Zhu. 2022. Findings of the shared task on multilingual coreference resolution. In *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*, pages 1–17, Gyeongju, Republic of Korea. Association for Computational Linguistics.

A Prompts

In this appendix, we present the prompts used in our solutions.

A.1 The Language-Universal Prompt

The language-universal prompt is based on our linguistic knowledge. It was written by hand and then revised by GitHub Copilot chat (GPT-5.3-Codex) for grammar and formulation issues. The empty token instructions are inserted only for the corpora where zero mentions are present.

```
# Role
You are a linguistic annotation assistant. Your task is to
perform coreference resolution for tokenized text in
{$LANGUAGE}.

# Primary Objective (High Recall)
Maximize recall of valid coreferent mentions. Prefer
including a plausible mention over omitting it.
If uncertain between "annotate" and "skip", annotate.

# Task
Identify all token spans that refer to entities and link
coreferent mentions with the same entity ID.

Include mentions of:
– named entities (persons, locations, organizations, works,
events, dates, times),
– nominals and definite descriptions,
– pronouns and demonstratives when referential,
– possessive and genitive referring expressions,
– appositions and aliases,
– nested mentions,
– repeated lexical mentions,
– implicit subjects (pro–drop) via empty tokens when
supported by morphology/context.

Mentions may:
– span multiple tokens,
– be nested,
– cross clause boundaries through coreference chains.

## Input Format
You receive one chunk of linguistically tokenized text in
Universal Dependencies style.
– Tokens are separated by single spaces.
– The chunk is a single line (no paragraph breaks).
– Some fused forms are split (for example: don't –> do n't;
Spanish del –> de el).

## Output Format
Return only one line: the annotated token sequence.
– Keep all original tokens exactly as given.
– Do not change tokenization, spacing, punctuation, or
token order.
```

– Add annotation marks directly on tokens.
– If no mentions are present, return the input unchanged.

Annotate mention boundaries by xml–like opening and closing tags where a boundary occurs:
– mention start: <eX>
– mention end: </eX>

Entity ID Policy
– Use IDs e1, e2, e3, ... in order of first appearance.
– Reuse the same ID for all mentions of the same entity.
– Different entities must have different IDs.

High–Recall Decision Rules
– Prefer broader mention coverage: include pronouns, shortened names, titles, and descriptive nominals when they likely refer to an entity.
– In borderline cases, annotate the candidate mention unless there is strong evidence it is non–referential.
– If two spans could be linked and both are plausible, link them.
– Keep both outer and inner spans when nesting is plausible.
– Avoid only clearly non–referential items (purely expletive/pleonastic forms, idiomatic non–entity uses, and strictly generic non–referential nouns).

{$EMPTY_TOKENS_INSTRUCTIONS}

Annotation Example
<e1>the streets of <e3><e2>Bangladesh 's</e2> capital city </e3> , <e3>Dhaka</e3></e1>

{$FEW_SHOT_EXAMPLES}

Input:
{$DATA}

Output:

Empty token instructions for zero mentions.

Empty Tokens
Some entities are implicit (for example, dropped subjects in pro–drop languages). In such cases, insert an empty token ## and annotate it.
– Place ## immediately after the syntactic head (usually the verb).
– Optionally include the omitted form if needed.
– Add empty tokens only when strongly supported by context and morphology.

Few-shot examples (1 or 3).

Example 1 input:
{$EXAMPLE_1_INPUT}

Example 1 output:
{$EXAMPLE_1_OUTPUT}

Example 2 input:
{$EXAMPLE_2_INPUT}

Example 2 output:
{$EXAMPLE_2_OUTPUT}

Example 3 input:
{$EXAMPLE_3_INPUT}

Example 3 output:
{$EXAMPLE_3_OUTPUT}

A.2 Language-Specific Prompts

LLM-enhanced language-specific prompts were used for the final submission. GitHub Copilot chat (GPT-5.3-Codex) was given the universal prompt and gold training data in the specified language. It was asked to optimize the universal prompt for the specified language:

How could I improve this prompt to achieve higher performance on this English data? Given data are the expected output.

This was done for all languages of the data. Copilot modified the prompt to fit the specific phenomena of the given language. The instructions remain in English for all languages, but examples of the target language were added. The generated prompt template is longer and more detailed.

The English-specific prompt follows. Black represents the text unchanged from the language-universal prompt, paraphrases are brown, additions are teal, and added sections that differ considerably across languages are purple (especially the language-specific guidelines section).

Role
You are a linguistic annotation assistant. Your task is to perform coreference resolution for tokenized text in {$LANGUAGE}.

Primary Objective
Maximize coreference quality with high recall and schema–valid output.
Prefer recall only when referentiality is plausible and link consistency is preserved.

Task
Identify all discourse–referring spans and link coreferent mentions with the same entity ID.
In this corpus, referents include people, groups, places, objects, institutions, roles, and abstract discourse entities (events, propositions, statements, situations) when they are referred to again.
Prioritize complete, internally consistent chains while avoiding clearly non–referential spans.

Include mentions of:
– named entities (persons, places, organizations, works, events, dates, times),
– common noun phrases and definite descriptions,
– pronouns and demonstratives when referential,
– possessive referring expressions when referential,
– titles, roles, aliases, kinship terms, and appositions,
– nested mentions,
– repeated lexical mentions,
– abstract referents such as events, actions, claims, facts, and prior discourse segments when they are referred to again .

Exclude mentions that are clearly non–referential:
– expletive or pleonastic uses,
– idiomatic fillers and pure discourse particles,
– purely generic/class–level noun uses with no discourse referent,

```
– predicative–only descriptions that do not denote a discourse entity,
– connective uses that only structure syntax and do not refer.

Mentions may:
– span multiple tokens,
– be nested,
– cross clause boundaries through coreference chains.

## Input Format
You receive one chunk of linguistically tokenized text in Universal Dependencies style.
– Tokens are separated by single spaces.
– The chunk is a single line (no paragraph breaks).
– Some fused forms are split.

## Output Format
Return only one line: the annotated token sequence.
– Keep all original tokens exactly as given.
– Do not change tokenization, spacing, punctuation, or token order.
– Add annotation marks directly on tokens.
– If no mentions are present, return the input unchanged.
– Do not output explanations, comments, or extra lines.

Annotate mention boundaries by xml–like opening and closing tags where a boundary occurs:
– mention start: <eX>
– mention end: </eX>

### Entity ID Policy
– Use IDs e1, e2, e3, ... in order of first appearance.
– Reuse the same ID for all mentions of the same entity.
– Different entities must have different IDs.

### Coreference Linking Policy
– Link only when coreference is supported by syntax, morphology, discourse structure, lexical identity, or clear semantic equivalence.
– If multiple antecedents are plausible, choose the nearest discourse–compatible antecedent unless broader context clearly favors another.
– Keep nested mentions only when both inner and outer spans are independently referential.
– Prefer a shorter, head–centered span unless a longer span is clearly required by corpus style.
– Preserve abstract–anaphora chains when pronouns or demonstratives refer back to clauses, events, claims, or previous discourse segments.

### English–Specific Guidance
– Resolve personal pronouns (`I`, `you`, `he`, `she`, `it`, `we`, `they`) and object forms when referential.
– Resolve possessives (`my`, `your`, `his`, `her`, `its`, `our`, `their`) when they denote discourse referents.
– Treat demonstratives (`this`, `that`, `these`, `those`) as mentions only when they point to concrete or abstract discourse referents.
– Relative forms (`who`, `whom`, `whose`, `which`, `that`, `where`) may be annotated when they genuinely refer to an antecedent in corpus style; do not annotate them when they are only structural linkers.
– Distinguish referential `it/this/that` from pleonastic or weather/time/cleft uses.
– Literary English frequently alternates narration and dialogue; track speaker/addressee shifts carefully before linking first– and second–person mentions.
– Titles and role nouns (e.g., `Mr.`, `Mrs.`, `doctor`, `captain`, `mother`, `father`) can be referential and should be linked when reused for the same entity.
– Abstract anaphora (`this`, `that`, `it`, `which`, `this fact`, etc.) may refer to prior propositions, speech acts, or events; annotate when clearly discourse–referential.
– Empty–token mentions are generally not expected in English unless explicitly required by the injected empty–token instructions.

### High–Recall Decision Rules
– Prefer broader mention coverage for referential pronouns, titles, aliases, and abstract anaphora.
– In borderline cases, annotate only when referentiality is plausible and chain consistency is maintained.
– Do not create singleton or fragmented IDs if a plausible link to an existing entity exists.
– Avoid over–linking across incompatible number, gender, person, semantic type, discourse perspective, or clear topic shifts.

### Minimal Internal Procedure
1) Detect discourse–referring mentions, including abstract anaphora when justified.
2) Assign or reuse entity IDs in first–appearance order.
3) Verify tag well–formedness, nesting, and chain consistency.
4) Output one final annotated line only.

### Final Validation Checklist (before output)
– All tags are balanced and properly nested.
– Entity IDs are sequential by first introduction.
– No token text, order, or spacing has been changed.
– Relative forms are linked only when genuinely referential.
– Pleonastic or purely structural `it/this/that` uses are not annotated.
– Demonstratives and abstract anaphors are linked only when discourse–referential.
– Empty–token mentions are not invented unless explicitly required.
– No clearly non–referential mentions were annotated.

{$EMPTY_TOKENS_INSTRUCTIONS}

### Annotation Example
<e1>the streets of <e3><e2>Bangladesh 's</e2> capital city </e3> , <e3>Dhaka</e3></e1>

{$FEW_SHOT_EXAMPLES}

# Input:
{$DATA}

# Output:
```

A.3 Language-Specific Guidance

For comparison, we also include the language-specific guidance section for three other selected languages (Czech, Spanish, German). The prompts for the remaining 15 languages are analogous. See the GitHub repository for all the prompts.[12]

From the examples, we can see that, especially, lists of personal, possessive, demonstrative, and reflexive pronouns are included. Besides that, the instructions focus on omitted subjects in pro-drop

[12] https://github.com/pavlk-mm/landcore/tree/main/prompt_templates/language_specific_prompts

languages, morphological agreement, distinguishing purely grammatical uses of words that are otherwise referential (e.g. 'se' clitic in Spanish), abstract anaphora, and other common phenomena, especially discourse-related.

Czech–Specific Guidance
– Czech frequently omits subjects and other arguments. Use empty–token mentions when person, number, gender, and discourse context make the omitted referent recoverable.
– Link inflected forms of the same noun phrase or proper name when they denote the same discourse referent.
– Resolve personal pronouns (`já`, `ty`, `on`, `ona`, `ono`, `my`, `vy`, `oni`) and their clitic/object forms when referential.
– Resolve reflexive forms (`se`, `si`, `sebe`, `sobě`) and reflexive possessives (`svůj`, `svoje`, `svého`, etc.) only when they denote a discourse referent; do not annotate purely grammatical reflexive uses.
– Treat demonstratives such as `ten`, `ta`, `to`, `tenhle`, `tamten`, and related forms as mentions only when they point to a concrete or abstract discourse referent.
– In Czech, `to` often refers to a prior event, statement, or situation; annotate such abstract anaphora when clearly discourse–referential.
– Relative pronouns such as `který`, `která`, `které`, `co`, `kdo`, `kam`, `kde`, `odkud`, and inflected forms may be annotated when they clearly refer back to an antecedent in corpus style.
– Vocatives, kinship terms, occupations, and role nouns can be referential and should be linked when reused for the same entity.
– Spoken or interview–style discourse may contain resumptions, repairs, repetitions, and discourse markers; annotate only the genuinely referential parts.
– Dialogue participants and reported speech often shift perspective; track speaker/addressee carefully before linking first– and second–person forms.

Spanish–Specific Guidance
– Resolve personal pronouns (`yo`, `tú`, `él`, `ella`, `nosotros`, `vosotros`, `ellos`, etc.) and clitic/object forms (`me`, `te`, `se`, `lo`, `la`, `le`, `los`, `las`, `les`, `nos`, `os`) when referential.
– Distinguish referential clitics from purely grammatical uses; annotate only when they denote a discourse referent.
– Resolve possessives (`mi`, `tu`, `su`, `nuestro`, `vuestro`) when they denote discourse referents.
– Treat demonstratives (`este`, `ese`, `aquel`, `esto`, `eso`, `aquello`) and neutral `lo` as mentions only when they refer to concrete or abstract discourse entities.
– Relative forms (`que`, `quien`, `el cual/la cual/los cuales/las cuales`, `donde`, `cuyo`) may be annotated when they genuinely refer to an antecedent; do not annotate them when they are only structural.
– Spanish allows frequent omitted subjects and arguments; when empty–token mentions (`##`) are explicitly enabled by task instructions and context supports them, annotate and link them consistently.
– Distinguish discourse–referential `lo/eso/esto/ello` from non–referential or fixed–expression uses.
– News and formal Spanish often rementions entities via role nouns, institutional labels, or aliases (e.g., person name vs office; club name vs descriptor); link these when they denote the same referent.
– Abstract anaphora are frequent (`esto`, `eso`, `lo`, `este hecho`, `esta situación`, etc.) and may refer to prior propositions/events; annotate when clearly discourse–referential.

German–Specific Guidance
– Resolve personal pronouns (`ich`, `du`, `er`, `sie`, `es`, `wir`, `ihr`, `Sie`) and object/dative forms (`mich`, `dich`, `ihn`, `ihm`, `ihr`, `uns`, etc.) when referential.
– Resolve possessives (`mein`, `dein`, `sein`, `ihr`, `unser`, `euer`, `Ihr`) when they denote discourse referents.
– Treat demonstratives and pronominal adverbs (`dieser`, `jener`, `der/die/das` as pronouns, `diese/dies/das`, `daran`, `damit`, `darauf`, `dadurch`, `davon`, etc.) as mentions only when they refer to concrete or abstract discourse entities.
– Relative forms (`der/die/das`, `welcher`, `wer`, `wo`, `woran`, `womit`, etc.) may be annotated when they genuinely refer to an antecedent; do not annotate them when they are purely structural.
– Distinguish referential `es/das/dies` from pleonastic or purely formal uses (e.g., weather/extraposition/fixed constructions).
– German case, number, and gender inflection is strong evidence for (or against) links; avoid links that violate clear agreement constraints.
– News/editorial German often rementions entities via role nouns, institutional labels, and paraphrases (e.g., person name vs office/title; institution name vs descriptor); link these when they denote the same discourse referent.
– Abstract anaphora are common (`das`, `dies`, `daran`, `damit`, `dazu`, etc.) and may refer to prior propositions, events, or arguments; annotate when clearly discourse–referential.
– Empty–token mentions are generally not expected in this corpus unless explicitly required by injected empty–token instructions.

B CZDEMOS4AI

The CZDEMOS4AI project aims to create a retrieval augmented generation (RAG) system to answer Czech questions from various domains. We illustrate how an automatic coreference resolution system can be helpful for analyzing the generated answers. We analyze answers for 19 questions (from the area of Contemporary History Education) using our system. We compare answers produced by humans (gold), answers generated by Llama 3.3, and answers generated by the RAG system.

Table 3 shows that the gold answers, which are significantly longer than the generated ones, have a similar density of entities. From Table 4, we can see that the generated answers contain more mentions per 1000 words than the gold answers, suggesting that the generated answers are denser in terms of information included. The distribution of mention lengths is similar for both automatic methods, while the gold answers contain more long mention spans.

In the future, we plan to investigate how the coreference is linked to the reported flaws in the structure of argumentation in the generated answers.

dataset	entities				distribution of entity lengths				
	total	per 1k	length		1	2	3	4	5+
	count	words	max	avg.	[%]	[%]	[%]	[%]	[%]
Gold	714	104	14	1.4	75.5	16.5	4.3	2.1	1.5
Llama	505	109	18	1.8	62.2	24.0	5.9	2.4	5.5
RAG	324	96	21	1.7	68.2	15.7	8.0	3.4	4.6

Table 3: Statistics of the entities in the CZDEMOS4AI dataset.

dataset	mentions				distribution of mention lengths					
	total	per 1k	length		0	1	2	3	4	5+
	count	words	max	avg.	[%]	[%]	[%]	[%]	[%]	[%]
Gold	1,012	148	42	3.8	2.0	27.9	21.6	14.4	9.1	25.0
Llama	923	199	32	3.3	0.8	36.2	25.2	11.7	6.5	19.6
RAG	548	163	31	3.2	2.2	35.2	25.0	9.3	6.9	21.4

Table 4: Statistics of the mentions in the CZDEMOS4AI dataset.