Beyond Word Boundaries: A Hebrew Coreference Benchmark and an Evaluation Protocol for Morphologically Complex Text

Anonymous ACL submission

Abstract

Coreference Resolution (CR) is a fundamental NLP task, critical for long-form tasks as information extraction, summarization, and many business applications. However, CR methods originally designed for English struggle with Morphologically Rich Languages (MRLs), where mention boundaries often do not align with word boundaries, and a single token may consist of multiple anaphors. CR modeling and evaluation practices standardly assumes that, as in English, words and mentions mostly align, but this assumption breaks down for MRLs, creating a significant gap, particularly in the face of LLMs' raw text processing and end-to-end tasks. To assess and address this gap, we introduce *KibutzR*, the first comprehensive CR dataset for Modern Hebrew, an MRL rich with complex words and pronominal clitics. We deliver an annotated dataset that identifies references at all word, sub-word and multi-word levels, and propose an evaluation protocol that directly addresses word/morpheme boundary discrepancies. Our experiments show that contemporary LLMs perform significantly worse on Hebrew than on English, and performance degrades on raw unsegmented text, highlighting the unique challenges posed for CR by morphology. Critically, we show an inverse performance-trend in Hebrew relative to English, where smaller encoders perform far better than the much larger neural decoders, leaving ample space for investigation and improvement. This work lays the ground for advancing coreference resolution in MRLs, towards robust NLP models for such languages.

1 Introduction

The field of Natural Language Processing has achieved remarkable breakthroughs in recent years, driven by advanced large language models (LLMs) and large-scale resources.

Despite these advances, complex word structures continue to pose significant challenges for automatic text processing and discourse-level understanding. In particular, the task of Coreference Resolution (CR) involves identifying and clustering entities across a discourse, enabling deeper text comprehension. However, achieving accurate CR presents unique challenges in *morphologically rich languages* (MRLs), with phenomena as pronominal clitics and morpheme-stacking that obscure mention boundaries. These linguistic intricacies make mention detection and coreference resolution particularly difficult in MRLs.

Modern Hebrew, a Semitic morphologically rich language, exemplifies these challenges, where a single token in Hebrew may consist of multiple anaphors that designate different entities in one and the same token, via construct-state nouns and pronominal clitics (**?**). Similarly, Arabic dialects show pronominal clitic fusion (Maamouri et al., 2004), Turkish stacks multiple morphemes creating boundary detection problems (Schüller et al., 2017), and Slavic languages encode multiple grammatical relations within a single word. All these languages share the same critical challenge — that standard CR models and metrics assume that mentions essentially align with word boundaries — but this alignment either doesn't naturally apply, or requires error-prone preprocessing.

As NLP shifts towards end-to-end architectures and LLM processing of raw texts, evaluation practices designed for CR in English increasingly fail on MRL texts. The shift to generative LLMs has created an evaluation gap for morphologically rich languages: these models process raw texts directly, yet we lack frameworks to assess their performance when morphological segmentation and coreference errors are intertwined. In particular, many MRL

lower-resourced languages lack even the very basic research infrastructure, which no public benchmarks to assess CR and measure progress on. Without benchmarks that isolate such error sources, we cannot diagnose where and why CR models fail on morphologically-rich texts, and propose mitigations and improvements.

This work focuses on Hebrew, an MRL rich with clitics that break word-mention alignment. While significant progress has been made in Hebrew via pre-trained encoders such as AlephBERT (Seker et al., 2022) and DictaBERT (Shmidman et al., 2023), as well as Large Language Models (LLMs) like DictaLM (Shmidman et al., 2024), and while the Hebrew NLP community contributed valuable resources for shorter-text tasks such as sentiment analysis, named entity recognition (NER) (Bareket and Tsarfaty, 2021), and question answering (QA) (Cohen et al., 2025), there remains a notable lack of resources for *discourse-level* understanding, and in particular *coreference resolution* (CR), limiting the development of more complex applications and long-form Hebrew tasks.

In this work we address the multifaceted CR challenge in Hebrew, by developing the first comprehensive Modern Hebrew CR dataset, *KibutzR*, accompanied by annotation guidelines that account for morphologically complex phenomena and discrepancies in word-mention boundaries. Additionally we introduce an evaluation protocol that remains sensitive to word-mention boundary discrepancies, providing a robust framework for comparing and contrasting models, both generative and classification-based, on raw (unsegmented) texts.

The contribution of this paper is thus manifold. First, we deliver *KibutzR*, the first modern Hebrew CR dataset, alongside its detailed annotation guidelines and rule-based mention detector. Second, we introduce a language-agnostic, segmentation-aware evaluation protocol that handles mention boundary discrepancies without assuming gold morphological analysis — applicable to any MRL. Finally, we show a comprehensive empirical analysis of contemporary models tracing performance gaps back to their roots in detection and clustering. Together, these contributions provide an immediate resource for Hebrew NLP, and a methodological blueprint for targeting and achieving improved CR capabilities in Hebrew and other MRLs.

2 Morphological Challenges in Coreference Resolution

2.1 Referring Expressions in MRLs

In morphologically rich languages, referring expressions frequently occur as subtoken units. In Hebrew, for instance, the token דבריו ('his words') exemplifies this challenge: this single orthographic token requires segmentation into דבר_של_הוא (words_of_he).[1] It thus contains two separate mentions that can co refer to distinct entities: *words* and *he*.

This phenomenon extends across typologically diverse languages. Arabic *kitābu-hu* ('his book') similarly fuses nominal and pronominal elements that must be decomposed into *kitāb* + *hu* for coreference resolution (Maamouri et al., 2004). Turkish agglutination produces *evlerimizden* ('from our houses'), stacking morphemes—*ev+ler+imiz+den*—where possessive and plural markers create overlapping mention spans (Schüller et al., 2017). Basque *etxekoak* ('those of the house') demonstrates comparable fusion through *etxe+ko+ak*, interleaving genitive marking with determination (Soraluze et al., 2019). Georgian verbal morphology presents the most complex case: *mogvts'eren* ('they will write to us') encodes multiple argument references through preverb *mo-*, object marker *gv*, and subject agreement *-en*, all within a single form. These morphological patterns fundamentally challenge the word-as-unit assumption underlying most CR systems. Unlike English, where mention boundaries align with whitespaces, MRLs require models to simultaneously segment morphemes and resolve their referential relationships, transforming a primarily semantic task into one demanding morphosyntactic proficiency.

Beyond such bounded clitics, three additional phenomena increase the difficulty of CR in MRLs. First, pro-drop significantly increases ambiguity by omitting arguments that can be recovered from morphological cues (Demir and Akdağ, 2024; Maamouri et al., 2009; Soraluze et al., 2019). Unlike languages with obligatory overt subjects, pro-drop languages force CR systems to infer referents from verbal inflections, creating additional decision points in the resolution process.

[1] By Hebrew UD v2 conventions (Sade et al., 2018).

Second, construct state formations create deeply nested nominal structures that challenge mention boundary detection. Unlike English, which uses prepositions and determiners to mark possession and modification, Hebrew concatenates nouns directly, creating complex multi-word expressions as דו"ח ישיבת ועדת מנויי דיני בית הדין הרבני הגדול ('report of the meeting of the committee of appointments of judges of the Great Rabbinical Court'). These constructions pose significant challenges for CR systems: they need to identify the entire construct chain as a single mention while also recognizing potential embedded mentions to (co)-refer to.

Third, many MRLs show flexible word order, that erodes positional heuristics for salience and proximity that systems traditionally exploit (Al-Thubaity and Al-Dossari, 2017; Soraluze et al., 2019). Classical approaches relied on surface positioning patterns from fixed word order languages, where syntactic roles correlate with linear position. However, Hebrew allows both דן קרא את הספר and את הספר קרא דן ('Dan read the book'), undermining positional features and surface-level cues.

These three factors strip away surface cues that English-centric systems exploit, forcing (any kind of) models to rely more on morphological and semantic understanding rather than mere surface patterns.

2.2 Word Segmentation: From Pipeline Artifacts to Modern Bottlenecks

A long-standing, implicit however persistent, assumption in evaluating CR systems, is that space-delimited tokens represent single mentions. This paradigm, while standard in the field, fundamentally misrepresents the challenge of MRL CR and creates a substantial gap between reported performance in research papers and real-world deployment scenarios. In English, where referential expressions often appear as standalone tokens, the distinction is inconsequential — models trained on space-delimited text face no additional challenge in deployment. In MRLs, the challenge to identify and segment referential material is inseparable from coreference resolution, as multiple referential material routinely appear within a complex worda that are input-streamed to the model.

In order to mitigate this intrinsic duality of mention detection and coreference clustering, MRL research standardized the use of gold-segmented text for coreference annotation and coreference evaluation as a pragmatic solution. Concretely, all major MRL coreference corpora adopt this approach: the OntoNotes–Arabic corpus reuses the Penn ATB segmentation; Marmara–Turkish inherits METU–Sabancı morphological analysis; and EUSKOREC pre-extracts mentions with finite-state rules (Pradhan et al., 2012; Schüller et al., 2017; Soraluze et al., 2019). While gold segmentation simplifies annotation and streamlines evaluation, it creates an artificial evaluation scenario that diverges from modern text processing.

In the earlier NLP-pipelines era, *mention detection* and coreference clustering formed two distinct phases, allowing researchers to isolate error sources by comparing performance with gold versus automatic mention-boundary detection. This diagnostic capability has been crucial: Marmara's baseline dropped by 31.4 CoNLL F_1 when switching from gold to automatic mentions, while Basque reported 19–21-point drops (Schüller et al., 2017; Soraluze et al., 2015). Such comparisons revealed precisely where systems failed: was the bottleneck in the *detection* or *clustering* phase? In today's neural end-to-end era, LLMs process raw text directly, fusing mention detection and coreference clustering into a single phase (Lee et al., 2017; Joshi et al., 2019). The relevant contrast is in what the model is provide as input: should these be pre-segmented tokens or raw texts? Subsequently, the question becomes: when a CR model fails on raw Hebrew text, can we distinguish whether errors stem from segmentation challenges (failing to identify clitics), linguistic nuances (mishandling construct states), or genuine coreference confusion (incorrect cluster)?

This shift fundamentally changes how we should evaluate MRL coreference systems, yet existing CR benchmarks for MRLs continue to report results on gold-segmented text, creating an evaluation-deployment gap: models achieve strong performance on pre-segmented benchmarks but struggle with raw text in deployment. This misalignment prevents us from understanding — let alone improving — actual CR performance on MRLs, as we lack diagnostic tools to isolate error sources and quantify their relative impact.

2.3 Towards Resolving Coreference Resolution in MRLs

To address the fundamental gap in MRL CR we propose a comprehensive solution that aligns research benchmarks with deployment realities while maintaining diagnostic capabilities. This proposal defines two interconnected objectives.

First, we need a comprehensive dataset that captures the full spectrum of mention spans, from subword to multiword levels, to reflect how referring expressions actually manifest in these languages. Unlike existing MRL corpora that rely on pre-existing segmentation schemas, we need annotations that explicitly mark referential material within complex word-forms — pronominal clitics, construct states, and other linguistic phenomena. We address this gap in Section 3 by constructing KibutzR, the first comprehensive Hebrew coreference dataset with morpheme-aware annotations that systematically handle morphological challenges in CR.

Second, we critically need evaluation scenarios that bridge the gap between research and deployment while preserving diagnostic capabilities. Rather than abandoning the insights from pipeline-era evaluations, we propose a diagnostic ladder of input conditions that systematically varies the level of preprocessing: (1) raw text, as encountered in real applications; (2) automatic segmentation, revealing the impact of segmentation errors; and (3) gold segmentation, isolating pure coreference challenges from morphological intricacies. This three-regime evaluation protocol allows us to quantify the relative contribution of different error sources — segmentation versus clustering — while maintaining comparability with existing benchmarks. By testing models across all three input conditions, we can answer critical questions about model capabilities: How substantial is the performance gap between raw and segmented text? What proportion of errors stem from segmentation versus genuine coreference confusion? Do modeling advantages observed in English — where LLMs now dominate — translate to MRLs when processing raw text? We define and implement this evaluation framework in Section 4.

With these foundations in place — comprehensive data and diagnostic evaluation — in Section 5 we establish baseline performance across architectures, both state-of-the-art LLMs and neural encoder models, providing the first comprehensive evaluation leaderboard for Hebrew coreference resolution that reflects the gap between research scenarios and deployment realities.

Split	Docs	Sents	Tokens	Mentions	%Docs
Train	301	5,236	137,333	17,500	85.8
Dev	26	428	10,474	1,243	7.4
Test	24	487	12,168	1,451	6.8
Total	351	6,151	159,975	20,194†	100

Table 1: Corpus statistics for KibutzR. The marking † excludes singletons. With singletons, the corpus contains 47,879 mentions.

3 Building KibutzR: Construction and Annotation of the Modern Hebrew Coreference Corpus

3.1 Scope and Document Selection

The Hebrew KibutzR dataset is based on The Hebrew Universal Dependencies Treebank (Sade et al., 2018), containing 6,151 sentences, without document boundaries — a critical limitation for coreference resolution research. To solve this we reconstructed the original document structure. Through metadata analysis and discourse pattern recognition, we successfully identified document boundaries and segmented the continuous sentence stream into 351 complete documents. This transformation — from isolated sentences to coherent documents averaging 17.3 sentences (453.6 tokens) As shown in Figure 1 — provides Hebrew NLP with its first document-aware corpus derived from the richly annotated UD treebank.

The reconstruction process required rethinking train/dev/test partitioning. The original Hebrew UD splits scattered sentences from the same document across different partitions, creating evaluation contamination. We therefore relocated any document appearing in multiple partitions exclusively to training. This principled partitioning preserves compatibility with existing Hebrew NLP tools trained on UD while ensuring clean evaluation: dependency parsers and morphological analyzers can operate on the same distribution without compromising coreference evaluation integrity.

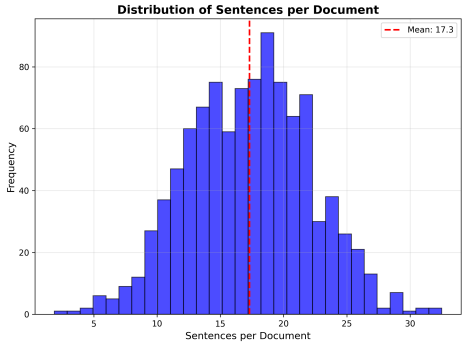

Figure 1: Distribution of sentences per document.

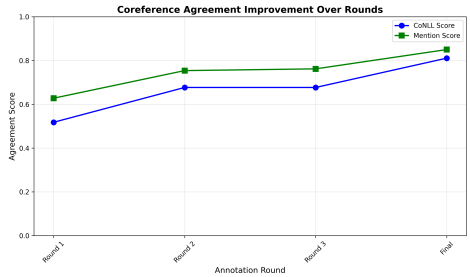

Figure 2: Agreement improvement across annotation rounds (CoNLL and Mention F_1).

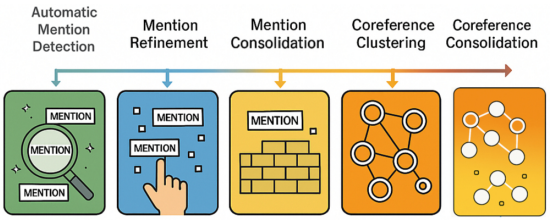

Figure 3: The five-stage annotation pipeline

3.2 Annotation Guidelines

Our annotation guidelines adopt OntoNotes 5.0 as their foundation while making two substantive departures to accommodate concrete morphosyntactic phenomena (Sec. 2).

First, OntoNotes restricts mention boundaries to space-delimited tokens and does not annotate sub-token morphemes. This is problematic for Hebrew where possessive suffixes and pronominal clitics carry independent referents. We therefore permit *morpheme-level* mentions within orthographic tokens whenever fused morphemes carry reference. To illustrate, in דבריו ('his words'), we annotate the possessive suffix as a separate pronominal mention alongside the full token. This modification enables annotation of possessive and pronominal clitics, and proclitic prepositions/conjunctions — all central to Hebrew reference. We do not split lexical roots from templatic patterns.

Second, OntoNotes generally annotates only single maximal NPs (no nested mentions/i-within-i), with limited exceptions (e.g., proper-noun premodifiers and appositives) — insufficient for recursive construct state nouns where sub-constituents maintain independent referents. We therefore treat recursive *smixut* (construct state) as nested mention hierarchies. For ממשלת אנגליה ('the government of England'), we allow coreference links to both the full compound and the embedded constituent אנגליה ('England'), capturing the dual referential nature of these constructions.[2]

The complete annotation manual, with detailed examples, is available in our repository.[3]

[2]Other adjustments, e.g. head selection for quantificational/partitive NPs and Hebrew-specific tests for non-referentials, are refinements rather than conceptual.

[3]Data, guidelines and metadata: anonymous.org.

3.3 Annotation Pipeline and Quality Control

Our annotation methodology follows a systematic five-stage pipeline, illustrated in Figure 3. Given the challenges outlined in Section 2, we separate mention boundary decisions from coreference clustering. This separation allows annotators to first resolve complex morphological boundaries before making referential judgments, reducing cognitive load and improving consistency. All stages are carried out independently by at least three annotators to secure reliable coverage in high agreement.

For the annotation platform, we extended the TNE annotation platform (Elazar et al., 2022) with support for morpheme-level span editing and robust right-to-left Hebrew display, enabling efficient and accurate annotation of complex Hebrew morphological structures. Annotators were compensated at 50 NIS/hour (60% more than the state's minimum wage).

The process begins with **automatic mention detection**. We designed a custom rule-based mention detector that leverages Universal Dependencies parses with rules for clitics, construct states, and other morphological and morphosyntactic phenomena. This recall-oriented system processes raw Hebrew text from UD-annotated documents and produces text with candidate mention-spans pre-marked, including nested mentions. The high-

recall design ensures comprehensive candidate coverage, allowing annotators to focus on refinement rather than mention discovery.

Next is the **mention refinement stage**, where human annotators receive text with pre-marked mention candidates and systematically accept or reject each candidate while adding any mentions missed by the rule-based detector. Using morpheme-level span editing capabilities, annotators can precisely define mention boundaries that cross traditional word boundaries. Each annotator produces their individual mention decisions for subsequent consolidation.

Later, the **mention consolidation stage** resolves inter-annotator variation by merging mention decisions into a unified inventory. Boundary disagreements are settled by majority vote or expert adjudication, producing a fixed set of mentions before clustering.

During **coreference clustering**, annotators process the consolidated mentions sequentially using a single-link strategy, deciding for each mention whether it opens a new cluster or links to an existing one. Singletons are retained to ensure complete mention coverage.

Finally, the **quality control and adjudication stage** handles systematic disagreement resolution through expert consolidation. An expert annotator reviews all disagreements and consolidates them cluster by cluster in sequential order, following an approach similar to Bornstein et al. (2020). This expert makes final decisions on conflicting annotations based on linguistic criteria and annotation guidelines, ensuring consistency across the dataset.

We monitored **inter-annotator agreement** after each batch and conducted a targeted revision following the first three batches. This iterative process improved macro average pairwise scores from 0.63 (mention) / 0.52 (CoNLL) initially to a final agreement of 0.87 for mentions and 0.81 for CoNLL F_1 — relative gains of 38% and 56% respectively (Figure 2). The final agreement scores for the dataset are: Mention $F_1 = 0.87$ (P: 0.85, R: 0.90), MUC = 0.84, $B^3 = 0.81$, $CEAF_\phi = 0.79$.

Notably, these final agreement scores **match or surpass** all of those reported for the most known large-scale CR corpora: OntoNotes (85.8% MUC F_1; Pradhan et al. 2012), LitBank (78% B^3 F_1; Bamman et al. 2019), and PreCo (77% agreement; Chen et al. 2018).

4 Experimental Setup

Goal and Evaluation Scenarios We set out to evaluate coreference resolution models and isolate the impact of morphological complexity on downstream performance. To do so, we propose to assess CR in 3 input conditions:

- **Raw Text:** The CR models receive unsegmented text and must discover mention boundaries and resolve coreference chains
- **Automatic Segmentation:** The CR models operate on text segmented by a state-of-the-art linguistic segmentation model.[4]
- **Gold Segmentation:** The CR models receive gold-standard segments boundaries.

For LLMs, we additionally assess CR performance in a **Gold Mentions** condition, where models receive pre-identified mentions, to isolate performance on the *clustering* subtask.[5]

Metrics. We follow the standard CoNLL-2012 shared task evaluation setup (Pradhan et al., 2012), which established three metrics as the canonical framework for cross-linguistic coreference evaluation.[6] We report standard CR metrics: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and $CEAF_\phi$ (Luo, 2005), along with their arithmetic mean (CoNLL F_1). MUC measures link-based precision and recall, B^3 evaluates mention-based clustering quality, and $CEAF_\phi$ computes the optimal alignment between predicted and gold clusters.

[4]For the automatic segmentation condition, we use the joint Hebrew segmentation model of Yshaayahu Levi and Tsarfaty (2024), which achieves state-of-the-art performance on Hebrew UD benchmarks and produces Universal Dependencies-conformant boundaries – 98.52 F1 on Hebrew UD benchmarks (Yshaayahu Levi and Tsarfaty, 2024).

[5]Because neural encoders operate on token sequences *(i.e., as token/span classifiers rather than segmenters)*, they require pre-segmented input by design and cannot process raw text directly, so neural encoder based systems are evaluated in automatic and gold segmentation conditions only. while LLMs are evaluated across all three input conditions plus gold mentions. This asymmetry reflects fundamental architectural differences: current neural encoder based systems inherit pipeline assumptions where segmentation precedes coreference, while LLMs are capable of handling both tasks simultaneously from raw text.

[6]Although we annotated singleton mentions, all reported results exclude them, following the conventions of prior work (Cattan et al., 2021).

Neural Encoder Models. We fine-tune two state-of-the-art neural CR architectures on KibutzR's training set. The *wl-coref* system (Dobrovolskii, 2021) uses word-level span representations with a coarse-to-fine antecedent scoring mechanism. The *LingMess* system (Otmazgin et al., 2023) incorporates multiple expert scorers that capture different linguistic signals. We adapt LingMess to Hebrew (LingMess-He) by replacing the English stopword and pronoun lists with their Hebrew equivalents in the scorer modules. We experiment with two Hebrew neural encoders, AlephBERT (Seker et al., 2022) and DictaBERT (Shmidman et al., 2023), which constitute the strongest pretrained models available for Modern Hebrew to date.[7]

Generative Language Models. We evaluate eight state-of-the-art LLMs: a Hebrew-monolingual model, DictaLM 2.0 (Shmidman et al., 2024) , and major multilingual LLMs: GPT-4.1/4o/o1/o3, and Gemini 2.0-Flash/-Lite/2.5-Pro. Following prior work (Le and Ritter, 2023), we use zero-shot prompting for the gold mentions condition to enable direct comparison with their English setup. For the end-to-end task (i.e., raw and gold segmentation conditions), we employ 2-shot prompting with examples from the training data, as zero-shot evaluation proved ineffective for this more complex task. Our end-to-end prompt implements a three-step pipeline that explicitly handles Hebrew morphological challenges. We evaluated each scenario with temperature 0 and report averages across five runs. See complete prompts and examples in appendix B.

5 Results and Discussion

Neural Encoder-Based Systems Table 2 presents our neural encoder baseline results under gold segmentation, the standard de facto evaluation standard in coreference resolution. Our strongest system, *LingMess-He* with the *DictaBERT* encoder, achieves 50.4 CoNLL F_1. The choice of the Hebrew encoder also proves critical; DictaBERT consistently outperforms AlephBERT by 5–6 F_1 points, likely due to its larger pre-training corpus and larger vocabulary, optimized for Modern Hebrew.

Model	Gold Segmentation			
	MUC	B^3	$CEAF_\phi$	CoNLL F_1
wl-coref (+AlephBERT)	47.1	41.6	44.4	44.4
wl-coref (+DictaBERT-base)	47.7	41.9	44.1	44.5
lingmess-he (+AlephBERT)	45.7	41.8	45.2	44.3
lingmess-he (+DictaBERT)	**52.6**	**47.7**	**51.0**	**50.4**

Table 2: neural encoder baseline performance with gold segmentation. Results averaged across 5 seeds; standard deviations range from ±0.7 to ±3.7 F_1 points. This evaluation regime matches all prior coreference work on other languages.

To isolate the effect of linguistic segmentation on downstream performance, we evaluated our best models on gold-segmented input vs. predicted segmentation. Table 3 reveals a consistent performance drop of 1.4–2.6 F_1 points when replacing gold with automatic segmentation. Although modest in absolute terms, this degradation isolates segmentation as an independent bottleneck in Hebrew NLP pipelines. This finding reveals that reporting scores only on gold-segmentation scenarios, hides a substantial portion of the error budget. For MRLs where the traditional assumption of reliable segmentation breaks down, results should be reported under both gold and automatic segmentation conditions, as well as on raw text, to provide realistic performance estimates.

Generative LLMs Table 4 evaluates state-of-the-art LLMs on Hebrew CR using in-context learning across three input conditions: raw text, gold segmentation, and gold mentions. Surprizingly, even when provided with perfect mention boundaries (the gold-mention condition), the best-performing LLM (GPT-4o at 45.4 F_1) falls 5.0 points behind the much smaller neural encoder-based baseline (50.4 F_1). This finding is particularly striking given the vast parameter difference—hundreds of billions compared with hundreds of millions. The underperformance persists across all evaluated models; prominent LLMs like o3, Gemini 2.5-Pro, and GPT-4.1 all fail to exceed 45 F_1. This pattern directly contradicts the English pattern where the same LLMs consistently outperform neural encoder-based systems (Table 5). The inverse performance trend, from a +7.0

[7] Training: 150 epochs, AdamW optimizer (encoder lr=1×10^{-5}, task lr=3×10^{-4}), dropout=0.3. `wl-coref`: max span=64, top-k=50. LingMess-He: max span=30, top-λ=0.4. Results averaged over 5 seeds. All models use early stopping on development CoNLL F_1 with hyperparameters following the original papers. Full hyperparameter details are supplied in the Appendix D. Training scripts will be released upon publication.

Model	Gold Segmentation				SOTA Automatic Segmentation				ΔF_1
	MUC	B^3	$CEAF_\phi$	F_1	MUC	B^3	$CEAF_\phi$	F_1	
lingmess-he (+AlephBERT)	45.7	41.8	45.2	44.3	44.2	40.4	43.9	42.8	−1.4
lingmess-he (+DictaBERT)	52.6	47.7	51.0	50.4	50.0	45.1	48.5	47.9	−2.6

Table 3: Neural model performance under gold versus automatic segmentation. The ΔF_1 column shows absolute performance drop when using SOTA segmentation instead of gold tokens. Both conditions use identical text; only token boundaries differ.

Model	Raw Text	Gold Seg.	Gold Mentions
Dicta 2.0	1.0	0.3	13.8
GPT-4.1	15.1	17.7	44.8
GPT-4o	14.2	14.5	**45.4**
o1	13.4	17.9	37.9
o3	15.7	19.4	42.1
Gemini 2.0-Flash	13.2	15.2	41.0
Gemini 2.0-Flash-Lite	12.1	15.2	38.4
Gemini 2.5-Pro	**22.2**	**26.8**	44.7
Neural encoder baseline (gold seg.)	—	50.4	—

Table 4: LLM performance (CoNLL F_1) under three supervision regimes. Results averaged across 5 runs; closed-source LLMs exhibit σ=0.4–3.2 despite temperature 0, reflecting known non-determinism in production systems.

Language	Neural-Encoder (Gold Seg.)	LLM (Gold Mentions)	Δ (LLM−NE)
English	81.4	88.4	**+7.0**
Hebrew	50.4	45.4	**−5.0**

Table 5: Cross-linguistic performance inversion. English results from Le and Ritter (2023) using the same LingMess architecture. The 12-point swing between languages reveals fundamental limitations of current LLMs on morphologically rich languages.

F_1 advantage in English to a -5.0 F_1 deficit in Hebrew, suggests that current LLMs, including the Hebrew-specialized DictaLM 2.0 (13.8 F_1), struggle fundamentally with coreference resolution in morphologically rich languages.

This underperformance stems from two types of weaknesses. Providing the LLM with gold mentions substantially improves its score from 26.8 F to 45.4 F , which confirms that mention detection is a significant hurdle. However, the fact that the clustering-only task plateaus at a modest 45.4 F indicates an additional bottleneck in discourse-level clustering.

Our detailed error analysis shows that in fact the mention clustering challenge is more significant. While gold segmentation inflates pronouns share, their resolution is rather trivial (via proximity and agreement features) and yield modest gains. However, neural encoders produce five times more correct clusters than LLMs (9.2/8.9 vs. 1.6–1.9 per document) and miss seven fewer gold clusters (9.7–10.0 vs. 17.0–17.3). So, when both architetures receive identical segmented input, LLMs struggle with discourse-level clustering. See further details on our detailed error analysis in Appendix H.

6 Conclusion

This paper introduces *KibutzR*, the first coreference corpus for Modern Hebrew, and uses it to re-examine modeling and evaluation practices of coreference resolution in morphologically rich languages. By evaluating both supervised encoders and frontier LLMs, across scenarios with raw text, automatic segmentation, and gold conditions, we can isolate the segmentation and clustering challenges.

Our key findings are that, first, state-of-the-art segmentation reduces performance of neural encoders modestly, by 1.4–2.6 F_1 points relative to gold. Next, contemporary LLM decoders underperform neural encoders by 5 points in Hebrew, even with gold mentions, reversing the English pattern where LLMs clearly dominate. Finally, we show that the primary bottleneck for LLMs is clustering rather than boundary detection, with gold segmentation yielding minimal gains ($\sim$2–4 F_1) versus dramatic improvements from gold mentions ($\sim$20–30 F_1). These findings reveal that current CR modeling and prompting practices struggle with morphologically rich languages when facing the dual challenge of segmentation and coreference.

We release KibutzR, its guidelines, its annotation UI and evaluation code, to enable research toward architectures that handle morphological segmentation and coreference resolution as interconnected components.

Limitations

Our corpus is limited to newswire text; broader genres remain a subject of future work. The requirement for UD-annotated text during annotation restricted us to news. The use of automatic parses may mitigate this, however at risk of increasing error propagation sources. Recently released Hebrew UD resources will enable expansion to other domains.

The corpus consists of publicly available news articles. As newswire may encode topical and gender biases, results may not generalize beyond news domains, or across news domains in vastly different (temporally spread) eras.

LLM results reflect zero-shot prompting for gold mentions (following prior work) and few-shot prompting for end-to-end evaluation; alternative instruction curricula may improve performance, but do not eliminate the need to model segmentation uncertainty. Finally, while our automatic segmenter is strong, improved segmentation tools could narrow (even if not entirely erase) the observed gaps.

Finally, while all of our code, guidelines, modeling, and experimental design is done in a language-agnostic manner, it is executed and evaluated only on Modern Hebrew texts. Parallel stream of research on additional MRLs are needed to strengthen the cross-lingual manifestation of this challenge. We hope that are code, guidelines, and actual tools (UI, evals) will greatly enhance and expand the availability of such resources and analyses for multiple languages.

References

Aseel Al-Thubaity and Sarah Al-Dossari. 2017. A coreference resolution approach using morphological features in Arabic. In *2017 2nd International Conference on Anti-Cybercrime (ICACC)*, pages 123–128. IEEE.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.

David Bamman, Ted Underwood, and Noah A. Smith. 2019. A large-scale corpus of coreference in literary text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5746–5756, Hong Kong, China. Association for Computational Linguistics.

Dan Bareket and Reut Tsarfaty. 2021. Neural modeling for named entities and morphology (NEMO2). *Transactions of the Association for Computational Linguistics*, 9:909–928.

Ari Bornstein, Arie Cattan, and Ido Dagan. 2020. CoRefi: A crowd sourcing suite for coreference annotation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 205–215, Online. Association for Computational Linguistics.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021. Realistic evaluation principles for cross-document coreference resolution. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 143–151, Online. Association for Computational Linguistics.

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. 2018. PreCo: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181, Brussels, Belgium. Association for Computational Linguistics.

Amir DN Cohen, Hilla Merhav, Yoav Goldberg, and Reut Tsarfaty. 2025. Heq: a large and diverse hebrew reading comprehension benchmark. *Preprint*, arXiv:2508.01812.

Şeniz Demir and Hanifi İbrahim Akdağ. 2024. Mention detection in Turkish coreference resolution. *Turkish Journal of Electrical Engineering and Computer Sciences*, 32(5):682–697.

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanai Elazar, Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2022. Text-based NP enrichment. *Transactions of the Association for Computational Linguistics*, 10:764–784.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Nghia T. Le and Alan Ritter. 2023. Are large language models robust coreference resolvers? *Preprint*, arXiv:2305.14489.

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.

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.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR conference on Arabic language resources and tools*.

Mohamed Maamouri, Ann Bies, Seth Kulick, Nizar Habash, and Owen Obeid. 2009. A conventional orthography for dialectal Arabic. In *Proceedings of the LREC 2009 Workshop on Semitic Language Processing*.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2023. LingMess: Linguistically informed multi expert scorers for coreference resolution. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2752–2760, Dubrovnik, Croatia. 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.

Shoval Sade, Amit Seker, and Reut Tsarfaty. 2018. The Hebrew Universal Dependency treebank: Past present and future. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Peter Schüller, Kübra Cıngıllı, Ferit Tunçer, Barış Gün Sürmeli, Ayşegül Pekel, Ayşe Hande Karatay, and Hacer Ezgi Karakaş. 2017. Marmara Turkish Coreference Corpus and Coreference Resolution Baseline. *arXiv preprint arXiv:1706.01863*.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Greenfeld, and Reut Tsarfaty. 2022. AlephBERT: Language model pre-training and evaluation from sub-word to sentence level. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–56, Dublin, Ireland. Association for Computational Linguistics.

Avi Shmidman, Shaltiel Tzion, and Moshe Shmidman. 2023. DictaBERT: A state-of-the-art BERT suite for modern Hebrew. *arXiv preprint arXiv:2308.16687*.

Shaltiel Shmidman, Avi Shmidman, Amir DN Cohen, and Moshe Koppel. 2024. Adapting llms to hebrew: Unveiling dictalm 2.0 with enhanced vocabulary and instruction capabilities. *Preprint*, arXiv:2407.07080.

Ander Soraluze, Xabier Arregi, Olatz Arregi, and Xabier Artola. 2015. Adapting the Stanford Coreference Resolution System to Basque. In *Proceedings of the 20th International Conference on Applications of Natural Language to Information Systems (NLDB 2015)*, pages 289–296.

Ander Soraluze, Xabier Arregi, and Xabier Artola. 2019. EUSKOR: An end-to-end coreference resolution system for Basque. *PloS one*, 14(9):e0221801.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. 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*.

Danit Yshaayahu Levi and Reut Tsarfaty. 2024. A truly joint neural architecture for segmentation and parsing. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, St. Julian's, Malta. Association for Computational Linguistics.

A Mention-Detection Implementation

A.1 Inputs and Dependencies

The system requires Universal Dependencies parses of Hebrew text, including tokenization, POS tags, and dependency labels. Root candidates comprise `NOUN`, `PROPN`, `PRON`, `NUM`, and verbs when occurring in *smixut* constructions. Span expansion utilizes the following permitted dependency labels: `appos`, `compound:smixut`, `nmod:poss`, and `conj`.

A.2 Span Construction Rules (Deterministic)

Spans expand bidirectionally from a root node along the permitted dependency edges, optionally including `amod` and `det` modifiers according to Hebrew-specific rules. The system handles Hebrew-specific phenomena including pronominal clitics (possessive, object, and prepositional), nested *smixut* constructions, determiners, numerals, and conjunctions. When overlaps occur, the system resolves them by selecting the maximal span, while compatible overlaps are merged. The traversal process is fully deterministic, employing fixed token order and single-threaded graph walks to ensure reproducibility.

A.3 Mention-Detection Pseudocode

Preconditions. UD-parsed Hebrew sentences; root candidates: `NOUN/PROPN/PRON/NUM` and verbs-in-*smixut*; permitted edges: `appos`, `compound:smixut`, `nmod:poss`, `conj`. Deterministic traversal; overlaps resolved by maximal span.

Algorithm 1 High-level Mention Detection for Hebrew

```
for each sentence in a UD-parsed corpus do
    for each token t in the sentence do
        if t is a root candidate (NOUN/PROPN/PRON/NUM/verb-in-smixut) then
            Initialize span S ← [t]
            Expand S left/right via permitted dependency relations
            Apply Hebrew-specific rules (clitics, determiners, numerals, conjunctions)
            Add S to mention candidates
        end if
    end for
end for
Filter/merge overlaps by maximal span
Output all candidate mentions
```

The complete Mention Detection system will be released upon publication.

B Prompt Templates

Gold-Mention Clustering (`doc_template`).

Task: Annotate all entity mentions in the following text with coreference clusters.
Use Markdown tags to indicate clusters, e.g. `[mention](#cluster_name)`.
Do not add extra information or clusters beyond those marked.
Input: [Tom](#) and [Mary](#) go to [the park](#). [It](#) was full of trees.
Output: [Tom](#cluster_0) and [Mary](#cluster_1) go to [the park](#cluster_2). [It](#cluster_2) was full of trees.

End-to-End Coreference (`e2e_template`).

Cluster all entity mentions in the following Hebrew text to coreference clusters. Use Markdown tags to indicate the coreference group in the output, with the format `[mention](#)`, e.g.
`)#(])#(]  [  [  )#(] [`

- `First, tokenize words to expose clitics`
- `Second, mark the mentions`
- `Finally, cluster all mentions together`

```
Tokenization example
# Input word:
# Output:    _ _
# Input word:
# Output:  _ _

Mention-marking example
# Input tokenized:   _ _
# Output: )#(])#(]  [_ _  [      )#(]  [

Clustering example
# Input: )#(])#(]  [_ _  [      )#(]  [
# Output:      )1_  #(]  [
)2_  #(])1_  #(]  [_ _  [
```

Examples

```
# Input:
לידס עלתה למקום החמישי אחרי שניצחה אתמול בחוץ במשחק
השלמה את מנצסטר סיטי 3 2. השערים ללידס: לי צפמאן (41),
קארל שאט (24), גורדון סטראקאן (26). לסיטי: אשלי וורד (94
מ-11 מ), דייויד ווייט (56).
# Output:
[לידס](#אגד_0) עלתה ל ה_ מקום ה חמישי אחרי ש ניצחה
אתמול ב ה_ חוץ ב משחק השלמה את [מנצסטר סיטי](#אגד_1)
3 2. [[ה שערים ל[לידס](#אגד_0)](#):[לי צפמאן (41), קארל
שאט (24), גורדון סטראקאן (26)]](#). ל[סיטי](#אגד_1): אשלי
וורד (94 מ - 11 מ), דייויד ווייט (56).

# Input:
הרבה החמצות ממצבים נוחים של יבנה, בגלל משחק הגנתי של
טבריה שהזמינה התקפות. בין חלוצי יבנה, שהרבו להחמיץ,
ניצל אנריקה ורון הזדמנות אחת בלבד, כדי להעניק לקבוצתו
פרס של 3 נקודות בעד נצחון שהיתה ראויה לו. שפט אריה
וולף, 1,000 צופים, ביבנה.
# Output:
הרבה החמצות מ מצבים נוחים של [יבנה](#אגד_0),
בגלל משחק הגנתי של טבריה ש הזמינה התקפות. בין
חלוצי [יבנה](#אגד_0), ש הרבו להחמיץ, ניצל [אנריקה
ורון](#אגד_1) הזדמנות אחת בלבד, כדי להעניק
ל[קבוצה_של_[הוא](#אגד_1)](#אגד_0) פרס של 3 נקודות
בעד [נצחון ש היתה ראויה ל[הוא](#אגד_0)](#אגד_1). שפט
אריה וולף, 1,000 צופים, ב יבנה.
```

```
Nested mentions are allowed; mark every
nested span and noun-phrase candidate.
Keep the text exactly as it was, except
for Markdown.
Do not output singletons in the final
cluster document.
```

C LLM Inference Pipeline

C.1 Deterministic Settings

- Temperature = 0.0; default top-p; max output tokens = min(context_limit − input, 4096).
- Up to 3 retries on format-validation failure; fixed random seeds where applicable.

Algorithm 2 Inference and Output Validation

```
for each document d do
    p ← render_template(d)
    r ← model(p, temperature = 0)
    Validate bracket balance, sentence alignment,
    cluster-ID schema
    if validation fails then
        apply non-semantic formatting fix
        retry (max 3)
    end if
    Parse clusters → C_d
end for
return ⋃_d C_d
```

C.2 Models and Artifacts

- Models used (replication list): GPT-4o, GPT-4o-mini, GPT-4.1, o1, o3, Gemini 2.5 Pro, Gemini 2.0 Flash, Gemini 2.0 Flash Lite, DictaLM 2.0-Instruct.
- The experimental framework is built on OpenAI API v0.28.1 and vLLM for local model serving.

D Neural Baseline Configurations

D.1 Training Regime

- Optimizer: AdamW; dropout: 0.3; weight decay: 0.01; epochs: 150.
- Seeds for averaging: {42, 123, 2021, 27182, 31415}.
- Max segment length: 512; max span width: 30 (LingMess-He) / 64 (wl-coref); rough scoring $k = 50$ (wl-coref).

Hyperparameter	LingMess-He	wl-coref
LR (encoder)	1×10^{-5}	1×10^{-5}
LR (task)	3×10^{-4}	3×10^{-4}
Dropout	0.3	0.3
Max span width	30	64
Max segment length	512	512
Rough scoring k	–	50
Hidden size (FFNN/Scorer)	2048	1024
Weight decay	0.01	0.01

Table 6: Key hyperparameters for neural baselines (replication).

E Annotation Guidelines (Operational)

E.1 Mention Types

Named entities; nominals; pronominals (including clitics); zero pronouns when recoverable from morphology.

E.2 Hebrew-Specific Handling

- Pronominal clitics: possessive on nouns (ספרו → [[ספר_של_[הוא]]), object on verbs (ראיתיו → [ראיתי_[אותו]), prepositional clitics (אליהם → [אל_[הם]).
- Construct state (*smixut*): allow nested constructs; annotate construct and internal heads when referential.

E.3 Exclusions

We exclude pleonastic or dummy uses, idioms with non-referential components (e.g., יצא מכליו), and negated non-existent entities (e.g., אין לי מכונית).

E.4 Inter-Annotator Agreement Protocol

The annotation process began with initial calibration on 10 documents followed by group discussion. We then conducted three annotation batches with continuous agreement monitoring and targeted rule clarifications. Final expert adjudication resolved disagreements, with IAA computed on pre-adjudication annotations.

F Scoring Definitions

F.1 Weighted Similarity

For brevity let $C_p = C_{\text{pred}}$ and $C_g = C_{\text{gold}}$.

$$\text{Sim}(C_p, C_g) = \frac{\sum_{m \in C_p \cap C_g} w(m)}{\sum_{m \in C_p \cup C_g} w(m)}. \quad (1)$$

We assign weights based on mention type: pronouns (including clitics) receive $w(m) = 0.2$,

while content mentions (nouns and named entities) receive $w(m) = 1.0$.

F.2 Outcome Categories

- **Correct**: similarity ≥ 0.5
- **Extra**: predicted cluster without a sufficient gold match
- **Missed**: gold cluster without a sufficient predicted match

G Environment and Artifact Bundle

Our experiments support single-GPU training with CPU-only inference capability. The artifact bundle includes all tool and model versions, encoder checkpoints, random seeds, run scripts, prompt JSON files, validator regexes, and parsing scripts. The bundle will be distributed upon publication.

H Error Analysis

We analyze where coreference models fail in Hebrew by comparing error patterns across segmentation conditions and model architectures. We examine (1) how mention type distributions — particularly pronoun ratios — shift with segmentation methods, and (2) clustering success rates. We categorize model outputs into three outcomes: *Correct* clusters (predicted clusters that match gold clusters), *Extra* clusters (spurious predictions), and *Missed* clusters (unresolved gold clusters). To determine matches, we compute weighted overlap between predicted and gold clusters, down-weighting pronouns (0.2) versus content mentions (1.0).[8]

Experimental Setup. We compare five configurations: **Neural–Gold** and **Neural–SOTA** (our best-performing neural encoder based system on gold/automatic segmentation), **LLM–Gold**, **LLM–SOTA**, and **LLM–Raw** (Gemini 2.5 Pro on gold/automatic/raw text). All analyses use the dev split.

Gold segmentation inflates pronoun detection. By examining which mentions each segmentation method discovers, we observe that switching from automatic to gold segmentation increases pronoun mention share by 9 percentage points in both architectures (Figure 4). This increase occurs because gold segmentation correctly parses grammatical morphemes, extracting bound pronouns (possessive suffixes, object clitics) that automatic segmenters either miss or incorrectly attach to stems. These newly discovered pronouns are trivial to chain using local syntactic agreement, directly accounting for the reported performance gains. Thus, improved segmentation primarily recovers easy pronouns rather than challenging content mentions.

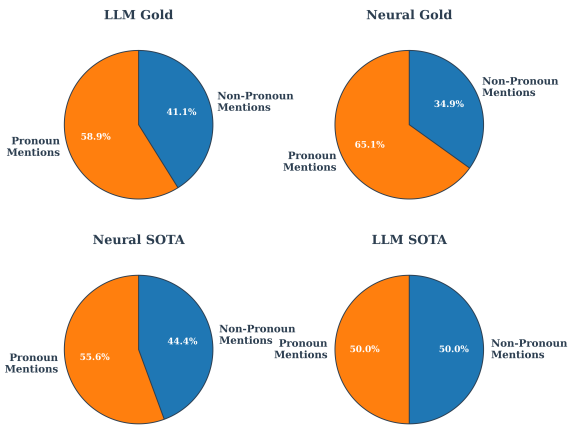

Figure 4: **Pronoun share rises under gold segmentation. LLM**: 50.0% (*LLM–SOTA*) → 58.9% (*LLM–Gold*) (+8.9pp). **Neural**: 55.6% (*Neural–SOTA*) → 65.1% (*Neural–Gold*) (+9.5pp).

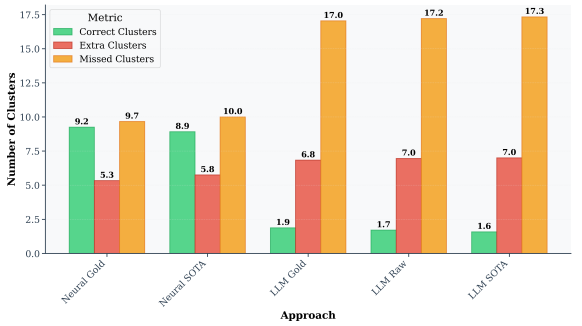

Figure 5: **Cluster outcomes by approach (per-document means).** Neural models produce 5× more correct clusters than LLMs regardless of segmentation quality.

Clustering, not segmentation, limits LLMs. Figure 5 reveals a striking architecture gap. Neural encoders produce five times more correct clusters than LLMs (9.2/8.9 vs. 1.6–1.9 per document) and miss seven fewer gold clusters (9.7–10.0 vs. 17.0–17.3). Critically, this gap persists even when both architec-

[8] Multiple entities can share the same pronoun form (e.g., multiple male entities all referred to as "he"), making pronoun overlap weak evidence for cluster matching.

tures receive identical segmented input (Neural–SOTA vs. LLM–SOTA), indicating that LLMs struggle with discourse-level reasoning rather than mention detection. The three LLM conditions (Raw/SOTA/Gold) reinforce this diagnosis: improved segmentation yields modest gains, but correct clusters remain sparse across all conditions. While neural models' performance drops primarily from segmentation errors (gold→SOTA gap), LLMs fail at the clustering stage regardless of input quality.