Assessing the Performance of Large Language Models for Coreference Resolution

Pradnya Gotmare[1], Shivam Tiwari[2], Manish Potey[3]
[1,2,3]*Department of Computer Science, K. J. Somaiya College of Engineering, Somaiya Vidyavihar University, Mumbai, India.*

***Abstract*—Coreference resolution is a critical task in natural language processing, determining when multiple expressions in text refer to the same entity. This capability is particularly vital in question answering (QA) systems, that require pre-cise interpretation of textual context. This paper shows usage of SpanBERT and LLaMA model to enhance reference resolution accuracy and processing efficiency. A hybrid approach involving LLaMA 3.1 with Prompt Engineering achieves significant gains over traditional models like SpanBERT, reaching up to 80.48% accuracy on the GAP dataset. These improvements highlight the po-tential of integrating advanced LLMs in real-world information retrieval applications. The modular architecture of the proposed system allows easy integration into existing QA pipelines. Moreover, by combining zero-shot capabilities of LLMs with deterministic fallback mechanisms, the framework can offer adaptability to domain-specific scenarios such as biomedical QA, conversational data, and multilingual document understanding.**

***Index Terms*—Natural Language Processing, Coreference Resolution, Question Answering, Large Language Models, GAP Dataset.**

I. INTRODUCTION

1.1 Background

Coreference Resolution is one of the most essential tasks of Natural Language Processing that involves identifying and resolving pronouns in the text to the entity they refer to. This is needed in various NLP applications including reading comprehension, question answering, machine translation, text summarization, information extraction, and dialogue systems. However, this referencing is a challenging task as human language, being inherently complex, makes it non-intuitive for machines. It can be difficult to provide all the necessary background information, context, resolve ambiguities and determine anaphor city for sentences with different structures and meanings. To address and resolve these challenges, several techniques and tools have been developed.

In Question Answering (QA) systems, effective coreference resolution is especially vital. QA models must extract factual information and understand the relationships between entities in long passages of text. If a system does not resolve pronouns or noun phrases correctly, it may produce misleading or off-topic answers. For instance, in biomedical QA, confusing "the patient" with "she" could lead to completely inaccurate treatment suggestions. Therefore, strong coreference handling is essential for tasks like summarization, machine translation, dialogue systems, and information extraction.

Historically, traditional methods for coreference resolution are dependent on rule-based and feature-driven techniques [1]. Early deterministic systems, like the multi-pass sieve architecture, applied hand-crafted rules in a specific order (for example, gender agreement, number agreement, and syntactic constraints) to connect entities. While these methods were understandable and fairly efficient, they struggled with subtle or context-dependent references, especially in longer texts.

The introduction of deep learning and contextual embeddings [2], especially through models like BERT and SpanBERT [3], led to significant improvements. These transformer-based models grasp word meaning in relation to the surrounding context, allowing for more precise resolution of unclear pronouns and longer-range dependencies. Span BERT, in particular built on BERT consider whole spans of text instead of individual tokens, making it ideal for pointing out potential antecedent-pronoun pairs. Despite these developments, these models have limits when dealing with inputs longer than 512 tokens and may struggle

in cross-domain scenarios without specific fine-tuning.

Recently, Large Language Models (LLMs) like GPT and LLaMA have shown impressive zero-shot and few-shot abilities in handling natural language references. Unlike traditional models, LLMs can manage extended contexts (thousands of tokens) and adjust to various domains through prompt engineering rather than retraining. This creates new possibilities for building hybrid systems that combine deterministic precision, such as rule-based methods, with the adaptability of LLMs. However, challenges still exist, including high computational costs, occasional errors, and inconsistent output formatting.

From a tools and ecosystem standpoint, researchers and practitioners have access to many NLP frameworks to build and assess coreference systems. Popular libraries include:

- Ollama + LLaMA: Enables efficient local use of LLaMA-based models with custom prompt engineering, ideal for testing in hybrid frameworks.
- scikit-learn, PyTorch, TensorFlow: Core machine learning libraries for training classifiers, managing embeddings, and calculating accuracy metrics like F1 or precision/recall.

Overall, the progress from rule-based methods to transformer-based embeddings and now to LLM-powered hybrid systems indicates successive improvements. It reflects the larger direction of NLP research, moving towards systems that balance context depth, scalability, and domain flexibility. This paper shows comparative performance of LLaMA 3.1 over SpanBERT. This approach enhances accuracy on the GAP dataset that showcases the potential of LLMs over NeuralCoref and SpanBERT. It can act as a co-reference resolution tool for real-world applications in legal, biomedical, and multilingual QA systems.

1.2 Motivation

QA systems must understand the implicit relations in natural text. Ambiguous pronouns and vague expressions remain a key challenge. A reliable coreference resolution frame-work bridges the gap between human-like comprehension and machine interpretation. Applications span are

- Medical texts – linking patient names, conditions, and treatments
- Legal documents – resolving pronouns across large, formal documents
- Search engines – improving context-driven result accuracy

1.3 Motivation

- Design and implement models that give high probabilities to correct references so that the output they create are relevant and accurate.
- Develop scalable solutions that are able to handle comprehensive dataset with minimal overhead in computations.
- Improving the performance metrics have better coreference for handling very complex sentences and further improving the performance metrics.

II. LITERATURE SURVEY

2.1. Related work

The several foundational and modern techniques have shaped the field of coreference resolution. Each method offers unique insights and tackles different challenges involved in identifying references in natural language text.

In the work on deterministic coreference resolution [4], author has described an entity-centric sieve architecture, applying coreference rules in a specific order from most to least precise. It focuses on precision and recalls by sharing features across mentions of the same entity and works well across different languages like English, Chinese, and Arabic. This approach brings together the insights of supervised and unsupervised mod-els along with deterministic rule-based systems. In another case of multi-pass sieve sys-tem [5] based on the deterministic model, different features like gender and number agreements are considered while passing the text. In this technique text passes through multiple times using a sequence of sieves, each focusing on different features. In this, document-level information is shared and achieved high accuracy in tasks like CoNLL-2011, demonstrating steady improvements through each layer. It is an extension of [6][7] multi sieve architecture that considers semantic similarity between mentions.

In the work [8] author has introduced higher order inference that uses the span ranking architecture from Lee et al [9] in an iterative manner. An antecedent

distribution is used as an attention mechanism to update the existing span representations. Instead of relying only on first-order relationships, this model uses span-ranking and attention mechanisms to make globally consistent decisions.

In the work [10] coreference resolution is mentioned for ambiguous pronouns using pre-trained BERT model [11][3]. In this, BERT is applied on Gendered Ambiguous Pronoun (GAP) Dataset [12] released by Google AI Language. It is a large and diverse dataset that is good enough for solving the issues in ambiguous pronoun resolution. In this work BERT model is trained to obtain contextual embeddings that creates a supervised dataset. It is followed by SVM classifier for classification and obtaining the correct mention for the target pronouns. In this work with GAP dataset the system had to consider whether the target pronoun is coreferent with the first, the second or neither of the given antecedent candidates [12].

In the work on scientific papers [13] multiple domain data from science, technology and medicines is considered to address coreference resolution. In the initial stage the corpus is annotated for coreference resolution by considering abstracts from different domain papers. In this work annotated mentions of scientific concepts namely process, methods, material and data are considered. Later on, these concepts are linked to entities in Wikipedia and Wikidata. Span BERT [14][15] and SciBERT based systems are incorporated for training purpose on the corpus data. This approach proposed sequential trans-fer learning for coreference resolution on research papers.

In an alternative approach with sequence-to-sequence models, coreference resolution is performed by finetuning a pretrained model. In this approach raw document is treated as a source sequence and the coreference annotation as a target sequence [16]. Later on, a pretrained encoder-decoder model like T5 is finetuned without any modification to the architecture. The model represents the coreference annotation as a sequence of actions that copy a token from the source sequence, start a mention span, or tag a predicted mention span with an integer [16]. It tags and regenerates input documents with embed-ded coreference information on English dataset with minimal task-specific tuning. The recent work describes Large Language models (LLMs) for coreference resolution with their strength and weaknesses [17]. Experimental results with LLMs indicate deficiencies in their fine-grained content analysis and comprehension.

2.2. Comparative Analysis of SpanBERT, Neurocore, and LLaMA 3.1+ Prompt Engineering

Table 1 discusses strength and limitations of SpanBERT, NeuralCoref and LLAMA 3.1. Neural-Coref is the coreference resolution library available with Spacy. It is the tradi-tional approach having rule based and statistical techniques. SpanBERT is derived from BERT model that is based on contextual embeddings. SpanBERT's ability to extract contextualized representations of words within and across sentences provides a power-ful mechanism for resolving coreferential expressions. Unlike BERT, SpanBERT masks contiguous spans of text (group of words), forcing it to learn broader contextual relationships. The SpanBERT architecture consists of Transformer Encoder blocks along with span masking module as shown in Figure 1.

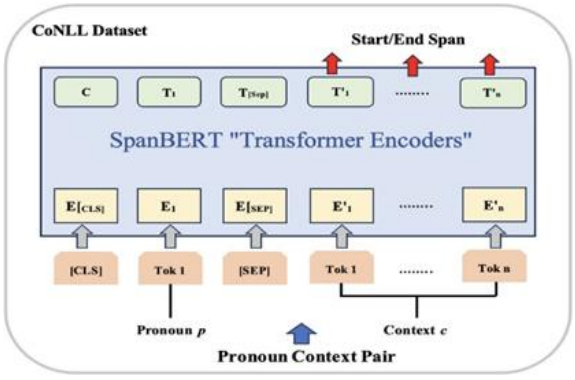

Figure 1. SpanBERT Architecture diagram [14]

Large Language model, LLaMA 3.1 is one of the largest LLMs available, following the transformer architecture. The model was designed with a standard decoder only trans-former model architecture and an iterative post training procedure [18]. This work em-phasizes role of Large Language models like LLaMA 3.1 for coreference resolution. This model has capability to learn statistical and semantic patterns associated with pro-noun, entities, and references in natural language. The generalized architecture of LLaMA 3.1 is shown in Figure 2.

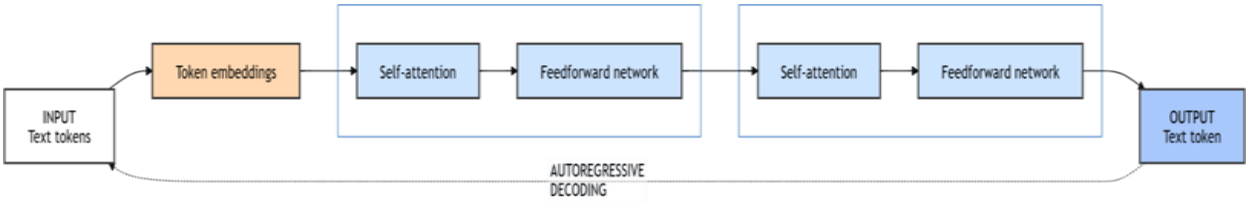

Figure 2. Overall architecture LLaMA 3.1 [18]

Table 1. Comparison of different approaches.

Model	Approach	Strengths	Limitations
SpanBERT	Uses a transformer-based encoder trained with span- level masking. It captures contextual embeddings of entire text spans, making it effective in structured datasets.	Strong on span prediction; effective for co-reference within short documents.	Limited to fixed-length inputs (512 tokens); less effective with ambiguous or long-range dependencies.
Neural-Coref	A hybrid of rule-based and statistical techniques implemented on top of spaCy. It links mentions based on syntactic heuristics and distance-based scoring.	Easy to integrate with NLP pipelines; fast and explainable rule behavior.	Struggles with complex nested references; poor generalization across diverse domains.
LLAMA 3.1 + Prompt	Uses custom natural language prompts to instruct the LLaMA 3.1 model to return structured JSON output identifying antecedents for ambiguous pronouns.	Processes long contexts effectively; adapts to domain language; no fine-tuning required.	May generate malformed or wrong outputs; computationally heavy for large-scale use.

III. METHODOLOGY

3.1. Dataset

GAP dataset is considered as it has different entries of ambiguous pronouns and candidates A, B [12]. It is especially designed to highlight gender ambiguity needed in coreference resolution. A and B indicate pronouns corresponding to two different nouns/noun phrases in antecedent part in a statement. The dataset is cleaned and stored in a normalized Unicode format.

3.2. Prompt Design

Prompts were refined to ensure JSON compliance and reduce wrong output. You are a coreference resolution expert.

3.2.1. Identify Ambiguous Pronouns

The first step is to carefully examine the input text and find all the ambiguous pronouns. Ambiguous pronouns can refer to more than one entity in the text, such as he, she, they, him, her, it, his, their, this, or that. At this stage, the goal is not to resolve them but simply to mark them as candidates for resolution. This ensures that the system focuses only on pronouns that need clarification, avoiding unnecessary work on clear references.

3.2.2. Resolve Their Antecedents

Once the ambiguous pronouns are identified, the next step is to resolve them by determining which noun phrase, or antecedent, each pronoun refers to. In datasets like the GAP dataset, these antecedents usually include Candidate A, Candidate B, or neither. If the pronoun clearly refers to one of the candidates, it should be linked to that candidate. However, if the pronoun does not refer to either candidate, the system must assign "Neither" as the antecedent. This step helps the model make a clear decision and prevents incorrect mappings.

3.2.3. Return Results in JSON Format

After resolving each ambiguous pronoun to its correct antecedent, the results must be returned in a strict JSON array format. Each pronoun should be represented as a JSON object containing two fields: "pronoun" for the detected pronoun itself and "antecedent" for the resolved reference entity. If multiple pronouns are found, they should all be listed as separate objects within the array. For example, the output should follow this structure:

{"pronoun": "he", "antecedent": "John"},
{"pronoun": "her", "antecedent": "Mary"}

It is important that no extra commentary, explanations,

or formatting appears outside of the JSON array. This ensures that the output is always machine-readable and con-sistent for downstream evaluation.

3.3. Model Setup of LLaMA 3.1

The overall system architecture combines LLaMA 3.1 with preprocessing and evalua-tion modules, run locally through the Ollama server. The flow includes several steps that ensure model predictions match ground truth labels for accurate assessment.

1. Ollama Setup and Model Initialization
- The Ollama server is installed and set up locally.
- The LLaMA 3.1 model, like the 8B variant, is downloaded and used for inference.
- This setup allows for low-latency querying of the model without needing external API calls, making offline experiments reproducible.

2. Input Processing
- Each row from the GAP dataset, which contains a passage, a pronoun, and candidate antecedents A and B, is retrieved.
- Text normalization, which includes handling unicode, cleaning whitespace, and validating candidates, is done to ensure consistency.
- The text segment is then placed into the carefully designed prompt de-scribed in Section III.B.

3. Model Querying
- The processed prompt is sent to the LLaMA model through the Ollama API.
- The model returns a JSON list of mappings, with each mapping linking an ambiguous pronoun to its predicted antecedent.
- If malformed JSON is received, a simple post-processing script tries to fix brackets or formatting before parsing.

4. Prediction Alignment
- From the JSON output, the resolved antecedent is extracted.
- This prediction is checked against the gold-standard label in the GAP da-taset.
- Candidate mapping is standardized, such as changing “A” to the name of Candidate A, to align with dataset conventions.

5. Evaluation Metrics
- Accuracy is calculated using scikit-learn’s accuracy score, comparing the predicted antecedent with the true antecedent label for each sample.

IV. IMPLEMENTATION

4.1. Code architecture

The overall code architecture uses a pipeline-based design to ensure modularity and transparency in evaluation. The GAP dataset is first loaded and pre-processed into a structured table format. This preserves each sample's pronoun, candidate antecedents, and gold-standard labels. The ambiguous pronoun and its surrounding sentence context are then input to the LLaMA 3.1 model through Ollama, which handles prompt execu-tion locally. The model produces a JSON-formatted output containing the resolved an-tecedent. This output is parsed and mapped back to the candidate entities (A, B, or Neither) defined in the dataset. Next, the prediction is compared with the ground truth labels to check for correctness. To ensure reliability, results are logged into an Ex-cel/CSV file that includes the input snippet, the model's resolved antecedent, the pre-dicted label, and the actual label. This design supports reproducibility, easy debugging, and clear error analysis.

4.2. Output storage
- To ensure transparency, reproducibility, and systematic evaluation, all out-puts are stored in a structured tabular format, such as an Excel sheet or CSV file.
- Each row corresponds to a single example from the GAP dataset and con-tains multiple descriptive columns. The first column, input pronoun and text snippet, records the ambiguous pronoun (e.g., “he,” “she,” “it”) along with the surrounding passage, ensuring context is preserved for human in-spection and interpretation.
- The second column, candidate antecedents A and B, lists the two possible entities specified by the GAP dataset, usually proper nouns or noun phrases like “John,” “Mary,” or “the doctor.” These candidates represent the only valid resolution options and serve as the comparison point for model pre-dictions.
- The third column, models resolved antecedent, captures the entity se-lected by the LLaMA 3.1

model after applying prompt-based coreference resolution. For example, if the model links "he" to "John," this column explicitly records Candidate A (John) as the resolved antecedent.

- Finally, the predicted vs. actual label column checks correctness by com-paring the model's choice with the gold-standard label from the GAP da-taset. It contains both the predicted label (A, B, or Neither) generated by the model and the actual label annotated in the dataset, making it easy to calculate accuracy and identify cases of agreement or error.

4.3. Usage of Models

Two different models are used to evaluate the effectiveness of coreference resolution approaches. At first, the SpanBERT model from Spark NLP served as a baseline. SpanBERT is a transformer-based architecture fine-tuned for span-level coreference resolution. It operates in a more traditional supervised setting, identifying mentions and clustering them into coreference chains.

Second, the LLaMA 3.1 model is used with carefully designed prompt engineering. LLaMA is asked for context and instructions to extract ambiguous pronouns, resolve their antecedents, and return outputs in a structured JSON format. This makes it suitable for handling nuanced natural language understanding. SpanBERT serves as the base-line, while LLaMA 3.1 with prompt engineering shows a modern generative approach.

4.4. Performance Evaluation

The evaluation was done on 1500 sampled rows from the GAP dataset to ensure statis-tically meaningful results. SpanBERT achieved an accuracy of 51.79%, representing the baseline performance. The proposed LLaMA 3.1 with prompt engineering approach achieved a much higher accuracy of 80.48%. The detailed comparison between ap-proaches is summarized in the table below:

Table 2 Performance of different models

Approach	Model Used	Accuracy (%)	Sampling Size
Approach 1	SpanBERT	51.79	1500 rows
Approach 2	LLaMA 3.1 8B + Prompt Engineering	80.48	1500rows

V. RESULTS AND DISCUSSION

With sampled data from GAP dataset the system accuracy is 80.48%. This data is in-tensely considered as ambiguous data for all the noun pronoun relationships. LLaMA being a generative model, sometimes create information that does not exist in the text, mislabel entities, or format responses incorrectly. This requires extra work to build re-gex checks, fallback parsing, or repair methods to ensure results are consistent and ma-chine-readable. Since GAP focuses on Wikipedia, it limits cross-domain evaluation and may not work well for legal, biomedical, or conversational text. Running LLaMA lo-cally uses a lot of resources. It requires high GPU memory and time, which slows down evaluation on large datasets.

VI. CONCLUSION AND FUTURE WORK

6.1. Conclusions

This work demonstrates a system that uses LLaMA 3.1 with prompt engineering to resolve coreferences more effectively than traditional SpanBERT-based systems on the GAP dataset. It has been observed that prompt refinement greatly reduced invalid JSON outputs, and structured output enabled consistent evaluation against ground truth. LLaMA showed strong generalization compared to rule-based SpanBERT. The system did face challenges with long-distance dependencies and with pronouns having multi-ple equally plausible antecedents.

We chose LLaMA 3.1 because of its generative reasoning ability and flexibility for handling ambiguous pronouns compared to fixed-span models. In comparison, SpanBERT achieved higher precision on straightforward cases but struggled with am-biguity, while LLaMA excelled in recall and resolving complex references. This re-sulted in overall higher accuracy.

6.2. Future work

Fine-tuning of LLaMA on specific datasets can be explored. The current method uses zero-shot prompting, but focused fine-tuning on legal, biomedical, or conversational data could improve accuracy and reduce errors. Finetuning would also help the model to recognize specific reference patterns that general pre-training might overlook.

Another challenge is to explore multilingual coreference performance. Expanding this method to non-English texts would enable evaluation in various languages and help assess performance of LLaMA for cross-lingual coreference resolution.

REFERENCES

[1] E. Bengtson and D. Roth, "Understanding the value of features for coreference resolution," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2008, pp. 294–303, doi: 10.3115/1613715.1613756.

[2] M. E. Peters *et al*., "Deep contextualized word representations (ELMo)," 2018.

[3] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, vol. 1, pp. 4171–4186, 2019.

[4] Deterministic coreference resolution based on entity-centric, precision-ranked rules," vol. 70, no. 8, p. 4943, 2010, doi: 10.1162/COLI.

[5] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky, "Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task," in *Proc. Conf. Computational Natural Language Learning (CoNLL)*, pp. 28–34, 2011.

[6] K. Raghunathan *et al*., "A multi-pass sieve for coreference resolution," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, pp. 492–501, 2010.

[7] Haghighi and D. Klein, "Coreference resolution in a modular, entity-centered model," in *Proc. North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 385–393, 2010.

[8] K. Lee, L. He, and L. Zettlemoyer, "Higher-order coreference resolution with coarse-to-fine inference," in *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, vol. 2, pp. 687–692, 2018, doi: 10.18653/v1/n18-2108.

[9] K. Lee *et al*., "End-to-end neural coreference resolution," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[10] M. Mohan and J. J. Nair, "Coreference resolution in ambiguous pronouns using BERT and SVM," in *Proc. Int. Symp. Embedded Computing and System Design (ISED)*, pp. 68–72, 2019, doi: 10.1109/ISED48680.2019.9096245.

[11] H. Dong, "Introduction to BERT and transformer: Pre-trained self-attention models to leverage unlabeled corpus data," 2019.

[12] K. Webster, M. R. Costa-jussà, C. Hardmeier, and W. Radford, "Gendered ambiguous pronoun (GAP) shared task at the Gender Bias in NLP Workshop 2019," pp. 1–7, 2019, doi: 10.18653/v1/w19-3801.

[13] Brack, D. U. Müller, A. Hoppe, and R. Ewerth, "Coreference resolution in research papers from multiple domains," in *Lecture Notes in Computer Science*, vol. 12656, pp. 79–97, 2021, doi: 10.1007/978-3-030-72113-8_6.

[14] M. Joshi, D. Chen, Y. Liu, D. Weld, and L. Zettlemoyer, "SpanBERT: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020, doi: 10.1162/tacl_a_00300.

[15] M. Joshi, L. O. Levy, and L. Zettlemoyer, "BERT for coreference resolution: Baselines and analysis," pp. 5802–5807, 2019.

[16] W. Zhang, S. Wiseman, and K. Stratos, "Seq2seq is all you need for coreference resolution," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, pp. 11493–11504, 2023, doi: 10.18653/v1/2023.emnlp-main.704.

[17] Y. Gan, J. Yu, and M. Poesio, "Assessing the capabilities of large language models in coreference: An evaluation," in *Proc. Joint Int. Conf. Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pp. 1645–1665, 2024.

[18] K. S. Raja Vavekanand, "Llama 3.1: An in-depth analysis of the next-generation large language model," *ResearchGate*, 2024, doi: 10.13140/RG.2.2.10628.74882.