Lost in Speech: Benchmarking, Evaluation, and Parsing of Spoken Code-Switching Beyond Standard UD Assumptions

Anonymous ACL submission

Abstract

Spoken code-switching (CSW) challenges syntactic parsing in ways not observed in written text. Disfluencies, repetition, ellipsis, and discourse-driven structure routinely violate standard Universal Dependencies (UD) assumptions, causing parsers and large language models (LLMs) to fail despite strong performance on written data. These failures are compounded by rigid evaluation metrics that conflate genuine structural errors with acceptable variation. In this work, we present a systems-oriented approach to spoken CSW parsing. We introduce a linguistically grounded taxonomy of spoken CSW phenomena and **SpokeBench**, an expert-annotated gold benchmark designed to test spoken-language structure beyond standard UD assumptions. We further propose **FLEX-UD**, an ambiguity-aware evaluation metric, which reveals that existing parsing techniques perform poorly on spoken CSW by penalizing linguistically plausible analyses as errors. We then propose **DECAP**, a decoupled agentic parsing framework that isolates spoken-phenomena handling from core syntactic analysis. Experiments show that **DECAP** produces more robust and interpretable parses without retraining and achieves up to **52.6%** improvements over existing parsing techniques. FLEX-UD evaluations further reveal qualitative improvements that are masked by standard metrics[1].

1 Introduction

Spoken code-switched (CSW) language poses a fundamental challenge for syntactic parsing. Unlike written text, spoken CSW exhibits disfluencies, repairs, ellipsis, fragmentary clauses, and discourse-driven structures that routinely violate the assumptions underlying canonical annotation frameworks such as Universal Dependencies (UD).

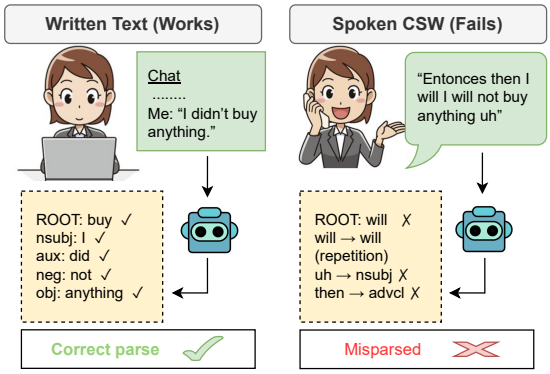

Figure 1: Illustration of the modality gap motivating this work. Parsers and LLMs typically produce well-formed dependency analyses for written text (left), but the same systems often misparse spoken code-switched utterances (right) due to disfluencies, repetition, and discourse phenomena that violate written-text assumptions.

These properties are intrinsic to conversational speech and are further amplified in bilingual settings, where syntactic structure must be negotiated across languages. As illustrated in Figure 1, models that perform well on written input often fail on spoken CSW, producing brittle or linguistically implausible analyses.

Recent work has shown that large language models (LLMs) can serve as effective tools for syntactic annotation and parsing in multilingual and low-resource contexts, particularly for written text (Lin et al., 2023; Tian et al., 2024; Kellert et al., 2025). However, their performance on spoken CSW remains inconsistent and often plateaus under standard parsing metrics. This degradation cannot be explained by bilingualism or data scarcity alone, but reflects a mismatch between written-language annotation conventions, rigid evaluation practices, and the inherently incremental and ambiguous nature of speech (Kahane et al., 2021; Dobrovoljc, 2022b). As a result, linguistically defensible analyses of spoken CSW are frequently penalized as er-

[1]Data and source code are available at https://anonymous.4open.science/r/sbench-0C0C

rors. While fine-tuning on large amounts of spoken CSW data might appear as a solution, this approach is largely infeasible due to the scarcity, cost, and uneven distribution of high-quality spoken corpora. Moreover, many challenges in spoken CSW, including non-canonical roots and discourse-level attachments, arise from structural assumptions in UD guidelines and evaluation metrics rather than data sparsity alone, making model capacity an unlikely path to scalable or language-agnostic solutions.

In this work, we argue that progress on spoken code-switched parsing requires rethinking both parsing architectures and evaluation practices, rather than scaling model capacity or data alone. We adopt a systems-oriented perspective that explicitly separates spoken-language phenomena from core syntactic analysis, and we ground this design in an empirical study of the structures that most frequently undermine canonical parsing assumptions. Building on this analysis, we introduce new benchmarks, metrics, and parsing abstractions tailored to the realities of conversational speech. Our contributions are:

- We introduce a linguistically grounded taxonomy of spoken CSW phenomena, capturing the sources of structural and cognitive complexity that undermine canonical parsing assumptions.
- We release **SpokeBench**, an expert-annotated, dispute-resolved, category-balanced gold UD benchmark for spoken CSW.
- We propose **FLEX-UD**, a weighted, ambiguity-aware evaluation metric that penalizes structurally catastrophic errors more than acceptable variation.
- We present **DECAP**, a decoupled agentic parsing framework that improves robustness and interpretability by isolating spoken-language handling from core syntactic analysis.

Together, these contributions provide a principled foundation for evaluating and parsing spoken code-switched language, highlighting the need for speech-aware language abstractions beyond written-text assumptions.

2 Related Work

UD Parsing for Non-Canonical and Code-Switched Text. UD has emerged as a widely adopted framework for cross-linguistic syntactic annotation, but its application to non-canonical data such as conversational speech and intra-sentential code-switching remains problematic. Spoken and code-switched utterances frequently violate core UD assumptions, including clause completeness and canonical head selection, leading to substantial annotation ambiguity (Çetinoğlu and Çöltekin, 2019; Kahane, 2019). To address these challenges, researchers have developed dedicated CSW treebanks and explored alternative dependency formalisms such as Surface-Syntactic Universal Dependencies (SUD), which modify head assignment to better reflect spoken structure (Kahane et al., 2021; Gerdes et al., 2019). Despite these efforts, prior work has largely focused on documenting annotation difficulties rather than enabling robust, scalable parsing or evaluation of spoken CSW.

LLMs for Annotation and Parsing. Large Language Models (LLMs) have recently been explored as parsers and annotation tools, showing that they can produce syntactic analyses for written text in zero-shot or few-shot settings (Zhang et al., 2025), and achieve competitive performance when fine-tuned (Bai et al., 2025). These capabilities motivate their use in bootstrapping annotations and generating synthetic treebanks, even for low-resource languages. However, consistent limitations remain. Beyond the well-documented challenges in written language, studies indicate that LLM performance degrades significantly with flexible, noisy, or conversational input, including spoken and CSW data (De Leon et al., 2024). Prior works have explored alternative parsing architectures, such as sequence-labeling approaches (Gómez-Rodríguez et al., 2024; Imran et al., 2024) or incorporating syntactic information into downstream tasks (Imran et al., 2025). While these models show promise for certain spoken-language tasks (e.g., detection or translation) (James et al., 2022; Huzaifah et al., 2024), ability to robustly parse fragmented or ambiguous syntax of real speech remains inconsistent.

Evaluation Limitations. Standard dependency parsing metrics such as Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS) enforce strict matching against a single gold tree, which is ill-suited to informal and non-standard language where multiple analyses may be equally plausible. Prior work has shown that such metrics conflate genuine structural errors with acceptable variation (Sterner and Teufel, 2025b,a). As a result, parser performance on spoken code-switched data remains difficult to interpret, motivating the need for ambiguity-aware evaluation.

3 Spoken Code-Switching: A Non-Canonical Parsing Domain

In this section, we examine spoken code-switching as a non-canonical parsing domain. We first develop a taxonomy of spoken CSW phenomena and then introduce **SpokeBench**, a gold benchmark constructed to systematically evaluate parsing performance under spoken-language conditions.

3.1 Taxonomy of Spoken Code-Switching Phenomena

Conversational speech exhibits non-canonical phenomena such as disfluencies, abandoned structures, repetitions, and discourse-driven insertions that are largely absent from written corpora and underrepresented in existing annotation guidelines. To identify the sources of systematic parsing failures in spoken CSW, we conducted a data-driven analysis of English-Spanish utterances from the Miami Corpus. Approximately 2,800 sentences were independently examined by trained linguists, who annotated spoken-language phenomena that violate standard dependency parsing assumptions. These observations were then consolidated through iterative discussion between linguists and computational researchers, resulting in a linguistically grounded taxonomy refined to remove infrequent or overlapping categories. The final taxonomy is directly motivated by observed parser failures and provides a structured account of the phenomena that complicate spoken CSW parsing.

Category Definitions and Corpus Distribution We now describe the final set of spoken code-switching phenomena included in our taxonomy. For each category, we provide (i) a linguistic characterization, (ii) an explanation of why the phenomenon violates or strains standard UD assumptions, and (iii) the resulting implications for human annotation and automatic parsing. Representative examples for each category, along with English glosses, are provided in Appendix B.

1. **Repetition** We define *repetition* as the recurrence of words or phrases that are not part of a fixed expression (e.g., excluding idiomatic forms such as "so so"). In spontaneous speech, it is commonly used to maintain the conversational floor while planning upcoming material (Clark and Wasow, 1998). From a dependency parsing perspective, repetition introduces multiple identical tokens that nonetheless require distinct syntactic attachments, undermining assumptions of stable head selection and one-to-one form–function mappings. Code-switched repetitions further complicate analysis when repeated material appears across languages.
2. **Discourse elements** *Discourse elements* include tokens or phrases whose primary function is pragmatic rather than structural, such as interjections and discourse markers (e.g., "well", "you know"). Distinguishing these elements from syntactically integrated material and identifying appropriate heads remains difficult even for expert annotators, particularly in bilingual contexts.
3. **Ellipsis** We define *ellipsis* as the presence of incomplete syntactic structures that are not subsequently repaired or replaced. These often arise when speakers abandon an utterance mid-production or are interrupted. Ellipsis poses challenges for UD parsing due to missing expected arguments, resulting in unconventional head assignments or dependency labels.
4. **Contractions** *Contractions* involve tokens that encode multiple syntactic units, such as English forms with apostrophes (e.g., "wasn't") or Spanish forms like "del" ("de el"). These constructions violate the assumption that each token corresponds to a single syntactic function unless explicitly split.
5. **Compound / MWE** *Compound words* consist of multi-word expressions that function as a single syntactic unit (proper names and conventionalized noun compounds). Treating these as independent tokens can obscure head-dependent relations.
6. **Break of thought** A *break of thought* occurs when a speaker initiates a phrase or clause but abandons it in favor of a revised continuation. Unlike ellipsis, the abandoned material is explicitly replaced. Parsing such structures requires inferring speaker intent and assigning syntactic roles to fragments that were never completed, a task that challenges both human annotators and automated parsers.
7. **Filler words** Among discourse-related phenomena, *filler words* are used specifically to hold the conversational floor during planning (e.g.,"umm"). We distinguish fillers from other discourse elements to assess whether they introduce distinct parsing challenges.

8. Slang / curses This category includes slang expressions and curse words whose interpretation depends heavily on pragmatic and cultural context. Such items have unclear syntactic roles and are often underrepresented / inconsistently treated in training data, posing challenges for rule-based and LLM-based parsers.

9. Enclisis *Enclisis* refers to verbs with attached clitic pronouns, a phenomenon common in Spanish. These forms combine verbal and argumental material into a single token, requiring morpheme-level reasoning for accurate dependency annotation.

3.2 SpokeBench: A Gold Benchmark for Spoken CSW Parsing

Subset Selection Strategy. Based on the identified spoken code-switching phenomena in the Miami Corpus, we construct **SpokeBench**, a gold benchmark for evaluating parsers on structurally complex spoken CSW. To our knowledge, no existing resource provides gold UD annotations for conversational CSW at this level of linguistic complexity. Annotating the full set of over 2,800 sentences was infeasible due to the cost and difficulty of spoken CSW annotation. We therefore selected a curated subset that balances coverage of spoken phenomena with overall structural complexity. Because utterances often exhibit multiple overlapping phenomena, sentences were stratified by both dominant phenomenon and degree of complexity rather than assigned to a single category. The resulting benchmark contains 126 sentences spanning ten categories; its composition is summarized in Table 3 (Appendix A).

Annotation Protocol and Adjudication. Annotations in **SpokeBench** follow a reformulated version of UD guidelines adapted for spoken, disfluent bilingual data. Standard UD conventions were applied where possible, with additional rules introduced for repetitions, contractions, discourse markers, fillers, and ellipsis. Each sentence was independently annotated by at least two linguistically trained annotators (seven in total), and disagreements were evaluated using a graded acceptability protocol that distinguishes acceptable variation from substantive errors. Only borderline or unacceptable cases were escalated and resolved through expert adjudication, yielding a consensus gold annotation for each sentence. Detailed annotation guidelines and review procedures are provided in Appendix C. **SpokeBench** will be released publicly to support future research on spoken CSW parsing.

4 Methodology

In this section, we describe our parsing and evaluation methodology for spoken code-switched language. We first introduce **DECAP**, a decoupled agentic framework designed to isolate spoken-language phenomena from core syntactic analysis. We then present our evaluation setup, including a critical examination of standard UD metrics and the introduction of **FLEX-UD**, an ambiguity-aware evaluation metric tailored to spoken CSW.

4.1 DECAP: A Decoupled Agentic Framework for Spoken CSW Parsing

We propose **DECAP** (DECoupled Agentic Parser), a modular, instruction-grounded framework for parsing spoken code-switched language. DECAP replaces monolithic parsing with specialized agents that handle spoken-language phenomena, language-specific normalization, core UD structure assignment, and global validation. The framework is guided by three principles: (i) *instruction grounding*, leveraging LLMs' ability to follow explicit linguistic guidelines; (ii) *modularity*, isolating distinct sources of complexity; and (iii) *extensibility*, enabling adaptation to new languages and spoken phenomena without retraining. Figure 2 illustrates the framework and agent interactions.

Spoken-Phenomena Handler (SPH). The SPH agent identifies and localizes non-canonical phenomena in spoken code-switched utterances prior to syntactic parsing. Given a tokenized sentence, SPH detects disfluencies such as repetitions, abandoned constructions, ellipsis, discourse markers, fillers, contractions, enclisis, and multiword expressions, and proposes minimal tokenization edits where required. Rather than assigning syntactic relations, SPH provides lightweight structural guidance, e.g., marking reparanda or stranded fragments and suggesting intended attachment points, which constrains downstream parsing. This separation prevents spoken-language irregularities from propagating as structural errors in later stages.

Language-Specific Resolver (LSR). This agent applies conservative, language-aware normalization to the output of SPH. It confirms or completes language-specific transformations necessary for UD parsing, such as expanding English and

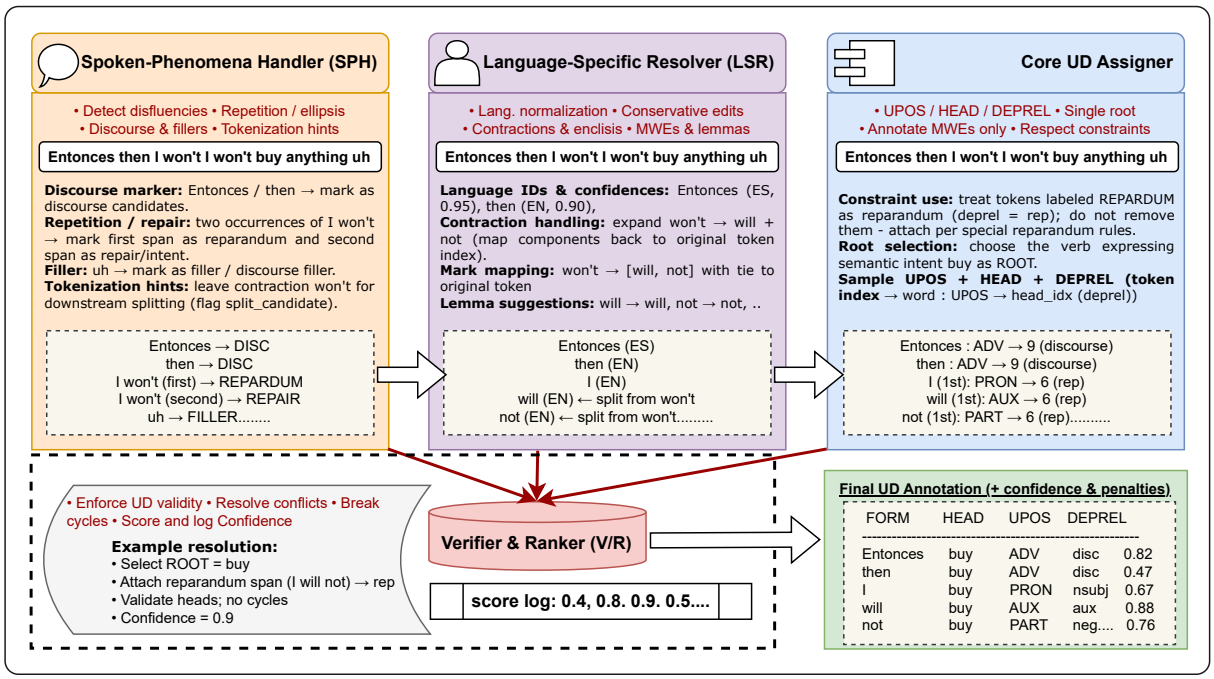

Figure 2: Overview of the **DECAP** framework for spoken code-switching parsing, illustrated with a running example. The input utterance ("*Entonces then I won't I won't buy anything uh*") is processed. The **Spoken-Phenomena Handler (SPH)** detects disfluencies (e.g., repetition, discourse markers, fillers) and provides tokenization hints. The **Language-Specific Resolver (LSR)** applies conservative, language-aware normalization, and contraction splitting. The **Core UD Assigner** constructs a dependency parse under these constraints, preserving reparanda and enforcing a single root. Finally, the **Verifier and Ranker (V/R)** enforces global UD validity and outputs a final annotation with confidence and penalty scores, demonstrating incremental resolution of ambiguity.

Spanish contractions or validating high-precision MWE, while respecting all upstream tokenization decisions. LSR additionally provides normalized lemmas and confidence scores for language identification. By isolating language-dependent decisions, LSR ensures that subsequent parsing operates over a representation that is both UD-compatible and robust to cross-linguistic variation.

Core UD Structure Assigner. The Core UD Structure Assigner produces the dependency parse over the normalized representation provided by SPH and LSR. Operating under their constraints, it assigns UPOS tags, head indices, and dependency relations to all annotatable nodes, ensuring exactly one root per sentence and annotating only combined multiword-expression nodes where applicable. Spoken-language annotations are treated as hard constraints, preserving disfluencies without forcing ill-formed reconstructions. This stage focuses exclusively on syntactic structure, deferring validation and correction to the final agent.

Verifier and Ranker (V/R). The V/R agent integrates the outputs of all upstream agents and enforces global structural well-formedness. It ensures UD validity by resolving issues such as multiple roots, invalid head references, or dependency cycles, consulting confidence signals and agent-specific priorities when repairs are necessary. In addition to producing sheet-ready annotations, V/R assigns per-token confidence and penalty scores, yielding an interpretable record of uncertainty and intervention. This final stage guarantees that DECAP outputs are structurally valid, auditable, and suitable for downstream evaluation.

Interaction protocol. DECAP applies a fixed, deterministic sequence of agents to each sentence, with each stage consuming the structured output of the previous one and enforcing increasingly global constraints, as detailed in the Appendix. D.

4.2 Evaluation Metrics

Limitations of Standard UD Metrics. Dependency parsing is commonly evaluated using attachment-based metrics such as Unlabeled Attachment Score (UAS), Labeled Attachment Score (LAS), and Content-Labeled Attachment Score (CLAS), which measure head and label agreement against a single gold tree. These metrics are effec-

tive for written text with largely unambiguous structure but are poorly suited to spoken CSW, where disfluencies, fragments, and discourse elements permit multiple UD-consistent analyses. By enforcing strict gold matching, standard metrics penalize linguistically defensible alternatives, such as acceptable root choices in fragments or variable attachment of discourse markers, thereby conflating genuine parsing failures with structural variation.

FLEX-UD Metric. FLEX-UD (Flexible Evaluation for UD) is a composite, severity-aware metric designed to evaluate dependency parses under the ambiguity endemic to spoken code-switching. Rather than enforcing a single gold tree, FLEX-UD compares system and gold tokenizations, IDs, POS tags, heads, and dependency labels using graded tolerances, producing both component scores and a single aggregated score. The metric is designed to (i) tolerate linguistically defensible variation, (ii) distinguish structurally catastrophic errors from benign alternatives, and (iii) provide interpretable diagnostics for error analysis and model ranking. Illustrative examples of error severity and penalty assignment are provided in Appendix E.

Formal definition. Let the five component scorers in **FLEX-UD** produce integer scores $S = \{s_{\text{Split}}, s_{\text{ID}}, s_{\text{UPOS}}, s_{\text{HEAD}}, s_{\text{DEPREL}}\}$, each in $[1, 100]$. Let the component weights be $w = \{w_{\text{Split}}, w_{\text{ID}}, w_{\text{UPOS}}, w_{\text{HEAD}}, w_{\text{DEPREL}}\}$ with $\sum w_i = 1$. The raw aggregate score is

$$\text{raw} = \sum_i w_i \cdot s_i.$$

To account for severity, define a severity multiplier $M \in (0, 1]$ computed from detected catastrophic errors (e.g., missing dotted MWE when gold has it, gross head invalidity, cycle-causing structural mismatches). Let $P \in [0, 1]$ be the severity penalty (higher = worse). Then

$$\text{final} = \text{round}\big(\text{raw} \times (1 - P)\big),$$

where P is computed as a bounded function of flagged catastrophic errors (see App. B). Output also the component vector S, weights w, the alignment, and a short diagnostics log.

5 Experiments

5.1 Experimental Setup

Dataset. All experiments are conducted on **SpokeBench**, the gold benchmark introduced in Sec . 3, which is derived from the Miami Corpus (Deuchar et al., 2014), a widely used English-Spanish CSW research dataset. We evaluate systems on a curated subset of 126 sentences, selected to balance spoken-language phenomena and structural complexity. Gold Universal Dependency annotations are used for evaluation across all systems to ensure control and directly comparable results.

Parsers. We compare three classes of parsing systems. *First*, we evaluate traditional dependency parsers based on Stanza, including models trained on English-only data, Spanish-only data, multilingual data, and a bilingual Stanza-based parser trained on a mixture of English and Spanish sentences for baselines. *Second*, we include BiLingua, an LLM-based Spanish-English parsing pipeline Kellert et al. (2025). It uses LLMs for UD annotation of code-switches, but remains a largely monolithic pipeline. *Finally*, we evaluate the proposed **DECAP** framework, which decouples spoken-phenomena handling, language-specific resolution, core UD structure assignment, and global verification into specialized agents. In all experiments, DECAP agents are instantiated using GPT-4.1 (version 2025-04-14) with deterministic decoding (temperature = 0), and no system is fine-tuned on **SpokeBench** or Miami data.

5.2 Overall Results

Results under Standard Metrics Table 1 reports performance under traditional attachment-based metrics (LAS, UAS, CLAS, UPOS-LAS) for three representative systems: a bilingual traditional parser, the LLM-based BiLingua pipeline, and the proposed DECAP framework. Across categories, all systems perform best on canonical sentences ("none") and show substantial degradation on spoken-language phenomena such as repetition, ellipsis, and discourse-heavy constructions. While DECAP consistently outperforms both baselines under LAS and UPOS-LAS, for example, improving overall LAS from 0.31 (BiLingua) to 0.48 and UPOS-LAS from 0.70 to 0.87, standard metrics compress these gains and fail to distinguish linguistically principled analyses from benign structural variation. As a result, improvements in handling ambiguity and non-canonical structure are only partially reflected by attachment-based scores.

Results under FLEX-UD Table 2 presents results under FLEX-UD for the same 3 systems. In contrast to standard metrics, FLEX-UD yields

Category	Bilingual Traditional Parser				Bilingua Parser				DECAP			
	LAS	UAS	CLAS	U-LAS	LAS	UAS	CLAS	U-LAS	LAS	UAS	CLAS	U-LAS
Repetition	0.15	0.25	0.08	0.62	0.32	0.55	0.20	0.82	0.36	0.58	0.21	0.79
Repetition+	0.16	0.30	0.10	0.58	0.22	0.38	0.17	0.71	0.27	0.41	0.22	0.77
Contr. (EN)	0.03	0.10	0.02	0.33	0.09	0.29	0.07	0.29	0.26	0.37	0.19	0.76
Contr. (ES)	0.02	0.13	0.01	0.40	0.31	0.37	0.33	0.78	0.35	0.39	0.26	0.86
Ellipsis	0.16	0.26	0.09	0.53	0.31	0.41	0.19	0.64	0.24	0.30	0.24	0.74
Ellipsis+	0.19	0.43	0.13	0.57	0.39	0.56	0.25	0.79	0.39	0.54	0.22	0.77
Discourse	0.17	0.23	0.12	0.59	0.33	0.45	0.27	0.75	0.19	0.27	0.16	0.83
Discourse+	0.19	0.28	0.16	0.61	0.35	0.45	0.29	0.69	0.36	0.44	0.32	0.74
Complex	0.14	0.23	0.12	0.51	0.37	0.48	0.29	0.71	0.27	0.35	0.21	0.70
None	0.20	0.30	0.15	0.64	0.38	0.45	0.26	0.77	0.14	0.17	0.11	0.46
Overall	**0.13**	**0.23**	**0.09**	**0.52**	**0.32**	**0.45**	**0.25**	**0.69**	**0.26**	**0.35**	**0.19**	**0.70**

Table 1: Performance of traditional parsers across spoken code-switching categories under standard UD metrics.

Category	Bilingual Traditional Parser					Bilingua Parser					DECAP				
	ID	UPOS	HEAD	DEPREL	Final	ID	UPOS	HEAD	DEPREL	Final	ID	UPOS	HEAD	DEPREL	Final
Repetition	9.7	9.7	9.7	9.7	24.8	78	85	65	63.5	74.7	72	85.3	68	71	77.7
Repetition+	17.4	17.4	17.4	17.4	32.3	76.3	77.5	56.3	57.3	70.1	70	81.1	63	66.6	73.2
Contr. (EN)	29.7	29.7	29.7	29.7	36	49.5	51	44	46.5	48.6	72.5	77.5	66	68.5	73.2
Contr. (ES)	30.4	30.4	30.4	30.4	36.4	56.6	78.8	61.1	65	64.1	80	88.5	66.3	74.1	79.3
Ellipses	12.4	12.4	12.4	12.4	27.8	79.0	78.8	57.4	70.7	74.0	51.8	79.5	54.5	64.0	67.6
Ellipses+	5.53	5.53	5.53	5.53	23.3	91.1	85.7	77.3	71.5	83.7	60	80	64.6	67.3	72.8
Discourse	10.7	10.7	10.7	10.7	25.9	79	79.5	55.8	60	70.6	63.5	83	52.5	57	66.7
Discourse+	29.4	29.4	29.4	29.4	39.5	71.6	77.4	62.6	65.6	69.8	60.4	81.3	59	68.3	71
Complex	13.3	13.3	13	13	28	69.2	77.3	60.9	65.7	69.8	55.8	70.9	53.4	55.7	63.8
None	12.7	13.7	13.7	13.7	29.9	89.5	91	66.7	69.9	81.4	47.2	55	34.8	39.5	52.4
Overall	15.6	15.7	15.7	15.7	29.5	76.2	78.5	62.3	65.1	72.2	58.2	73.7	54.6	59.1	66.6

Table 2: Component-wise evaluation (ID, UPOS, HEAD, DEPREL and aggregated Final) across spoken-CSW categories for three parser classes.

clearer separation across systems and sentence categories by explicitly accounting for ambiguity and error severity. DECAP achieves the highest overall FLEX-UD score (76.2), compared to 70.7 for BiLingua and 30.4 for the traditional parser, with especially large gains on repetition, ellipsis+, and discourse-heavy sentences. These improvements are driven not only by higher head and label accuracy, but by substantial reductions in catastrophic structural errors, which FLEX-UD penalizes more heavily than minor deviations. Additional results for monolingual and multilingual parsers are reported in Appendix F. Together, these findings demonstrate that FLEX-UD more faithfully captures qualitative improvements in spoken CSW parsing and that DECAP provides more robust and interpretable analyses than prior approaches.

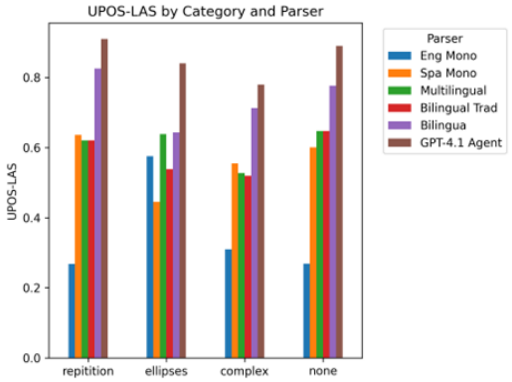

Figure 3: UPOS-LAS by Category and Parser; DECAP is the GPT-4.1 agent and performs the best across all categories.

6 Discussion and Linguistic Analysis

UPOS-LAS and LAS reflect complementary sources of difficulty. As shown in Figure 3, UPOS-LAS exhibits a highly consistent hierarchy across bilingual and multilingual parsers, with repetition achieving the highest scores, followed by canonical sentences ("none"), ellipsis, and complex constructions. Averaged across systems, repetition reaches approximately 0.80-0.90 UPOS-LAS, compared to 0.75-0.85 for none, 0.60-0.70 for ellipsis,

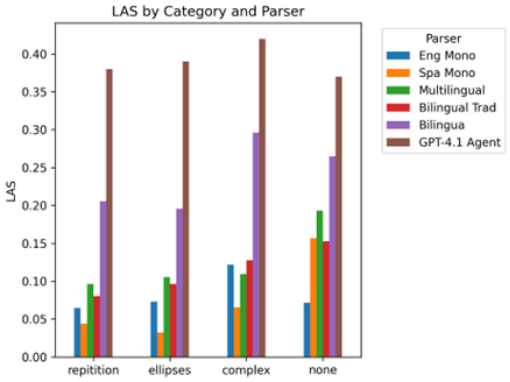

Figure 4: LAS by Category and Parser. DECAP is the GPT-4.1 agent and performs the best across all categories.

and 0.50-0.65 for complex sentences. This ordering reflects the sensitivity of UPOS-LAS to surface availability and local morphosyntactic cues: repetition maximizes overt lexical material, while ellipsis removes cues and complex constructions increase structural depth. Crucially, this hierarchy is stable across bilingual and multilingual parsers, whereas monolingual parsers score substantially lower, indicating that UPOS-LAS performance on CSW primarily reflects cross-lingual representational capacity rather than construction-specific heuristics. In contrast, LAS reveals a different hierarchy driven by structural ambiguity and recoverability.

LAS: structural ambiguity and recoverability. Insights from figure 4 show that LAS yields a reversed but equally systematic hierarchy, with canonical and complex sentences achieving the highest accuracy, and ellipsis and repetition consistently performing worst (none $\approx$ complex $>$ ellipsis $\approx$ repetition). Averaged across parsers, none and complex constructions reach approximately 0.75-0.85 LAS, while ellipsis drops to 0.55-0.70 and repetition to 0.50-0.65. This pattern aligns with established findings that ellipsis requires recovery of unexpressed structure and discourse-level inference (Merchant, 2001; Dobrovoljc, 2022a; Cavar et al., 2024), a difficulty corroborated by annotation effort: sentences with ellipsis required roughly two times as much annotation time as sentences without ellipsis. Repetition also reduces LAS by introducing competing attachment sites despite surface redundancy. LLM-based parsers follow this hierarchy but exhibit sharper declines for ellipsis and repetition, whereas the traditional bilingual parser degrades more uniformly, suggesting differences in sensitivity to structural ambiguity rather than some random errors.

These construction-specific difficulties are systematically amplified in bilingual production, where repetition, reformulation, and ellipsis arise more; a detailed analysis of bilingualism as a complexity multiplier is provided in Appendix G.

Implications for UD Annotation and Evaluation. The consistent difficulty of ellipsis and repetition across parsers has direct implications for dependency-based annotation and evaluation in spoken and bilingual data. Our results expose a tension between surface-oriented approaches, such as Surface-Syntactic Universal Dependencies, which avoid forced reconstruction (Gerdes and Kahane, 2016), and inference-based approaches that posit unpronounced but structurally present material (Merchant, 2001; Nielsen, 2004; Hardt and Romero, 2004; Liu et al., 2016). In spontaneous speech, ellipsis is often discourse-dependent and structurally indeterminate, leading to high annotation cost and lower LAS even for strong models. The persistence of these effects suggests that the challenge lies not in model capacity alone, but in the representational assumptions of fully specified inferred structures when applied to spoken language, where structural completeness is frequently underspecified rather than implicit.

7 Conclusion

Spoken code-switching exposes a fundamental mismatch between written-language assumptions and the structural realities of conversational speech. Our analysis shows that phenomena such as repetition and ellipsis introduce systematic ambiguity that challenges both human annotators and automated parsers, and that existing parsing techniques perform poorly under standard evaluation metrics, which often penalize linguistically defensible analyses as errors. To address this gap, we introduced a linguistically grounded taxonomy of spoken CSW phenomena, **SpokeBench**, a gold benchmark for spoken CSW parsing, **FLEX-UD**, an ambiguity-aware evaluation metric, and **DECAP**, a decoupled agentic parsing framework that isolates spoken-language handling from core syntactic analysis. Together, these contributions demonstrate that progress on spoken CSW parsing depends not only on improved models, but also on annotation and evaluation practices that explicitly recognize structural uncertainty as an inherent property of the data.

Limitations

This work focuses on depth and linguistic complexity rather than scale. **SpokeBench** is intentionally small and curated, prioritizing expert annotation and challenging spoken CSW phenomena over broad coverage across speakers, dialects, or interactional contexts. While this design enables detailed analysis, extending the benchmark to larger and more diverse spoken corpora would strengthen generalizability. In addition, our experiments focus on English-Spanish code-switching; although the identified phenomena are common in many bilingual settings, their distribution and interaction may vary across language pairs and sociolinguistic environments. Finally, we proposed an instruction-grounded framework rather than a trained parser, which enables extensibility without retraining, but depends on the quality of prompts and the underlying language models. Exploring hybrid approaches that combine spoken-aware system design with learned representations remains an important avenue for future research.

Ethical Considerations

This work uses data from the Miami Corpus of English-Spanish code-switching (Deuchar et al., 2014), which was collected and released for research purposes; no new data were collected and no personally identifiable information was accessed. While our methods aim to improve parsing and evaluation of spoken bilingual language, LLM-based annotation may propagate biases or misrepresent code-switching practices if applied without linguistic oversight, and should therefore be used cautiously outside research settings. We used AI-assisted tools (e.g., ChatGPT and Grammarly) solely for grammatical refinement and clarity of presentation.

References

Xuefeng Bai, Jialong Wu, Yulong Chen, Zhongqing Wang, Kehai Chen, Min Zhang, and Yue Zhang. 2025. Constituency parsing using llms. *IEEE Transactions on Audio, Speech and Language Processing*.

Damir Cavar and 1 others. 2024. Syntactic annotation of spoken and disfluent language. In *Proceedings of LREC-COLING 2024*.

Özlem Çetinoğlu and Çağrı Çöltekin. 2019. Challenges of annotating a code-switching treebank. In *Proceedings of the 18th international workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 82–90.

Herbert H. Clark and Thomas Wasow. 1998. Using "uh" and "um" in spontaneous speaking. In David E. Meyer and Steven Kornblum, editors, *Lexical and Syntactic Processing*, pages 199–228. Lawrence Erlbaum Associates.

Frances Adriana Laureano De Leon, Harish Tayyar Madabushi, and Mark Lee. 2024. Code-mixed probes show how pre-trained models generalise on code-switched text. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3457–3468.

Margaret Deuchar, Peter Davies, Judith Herring, María C. Parafita Couto, and Dan Carter. 2014. Building bilingual corpora. In Enlli M. Thomas and Ineke Mennen, editors, *Advances in the Study of Bilingualism*, pages 93–110. Multilingual Matters, Bristol.

Kaja Dobrovoljc. 2022a. Annotating ellipsis in dependency treebanks. *Language Resources and Evaluation*.

Kaja Dobrovoljc. 2022b. Spoken language treebanks in universal dependencies: An overview. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1798–1806.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2019. Improving surface-syntactic universal dependencies (sud): surface-syntactic relations and deep syntactic features. In *TLT 2019-18th International Workshop on Treebanks and Linguistic Theories*, pages 126–132. Association for Computational Linguistics.

Kim Gerdes and Sylvain Kahane. 2016. Surface-syntactic universal dependencies. In *Proceedings of COLING 2016*, pages 223–235. ACL.

Carlos Gómez-Rodríguez, Muhammad Imran, David Vilares, Elena Solera, and Olga Kellert. 2024. Dancing in the syntax forest: fast, accurate and explainable sentiment analysis with salsa. In *SEPLN–CEDI-PD 2024. Seminar of the Spanish Society for Natural Language Processing: Projects and System Demonstrations*, volume 3729 of *CEUR Workshop Proceedings*, pages 12–17, A Coruña, Spain.

Daniel Hardt and Maribel Romero. 2004. Ellipsis and the structure of discourse. In *Proceedings of SALT 14*.

Muhammad Huzaifah, Weihua Zheng, Nattapol Chanpaisit, and Kui Wu. 2024. Evaluating code-switching translation with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6381–6394.

Muhammad Imran, Olga Kellert, and Carlos Gómez-Rodríguez. 2024. A syntax-injected approach for faster and more accurate sentiment analysis. *arXiv preprint arXiv:2406.15163*.

Muhammad Imran, Olga Zamaraeva, and Carlos Gómez-Rodríguez. 2025. Synner: Syntax-infused named entity recognition in the biomedical domain. *JAMIA Open*.

Jesin James, Vithya Yogarajan, Isabella Shields, Catherine I Watson, Peter Keegan, Keoni Mahelona, and Peter-Lucas Jones. 2022. Language models for code-switch detection of te reo māori and english in a low-resource setting. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 650–660.

Sylvain Kahane. 2019. Interpreting and defining connections in dependency structures. In *5th international conference on Dependency Linguistics (Depling)*, pages 89–99. Association for Computational Linguistics.

Sylvain Kahane, Bernard Caron, Emmett Strickland, and Kim Gerdes. 2021. Annotation guidelines of ud and sud treebanks for spoken corpora. In *Proceedings of the 20th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2021)*, pages pp–35. Association for Computational Linguistics.

Olga Kellert, Nemika Tyagi, Muhammad Imran, Nelvin Licona-Guevara, and Carlos Gómez-Rodríguez. 2025. Parsing the switch: Llm-based universal dependency annotation for code-switched language. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics.

Boda Lin, Xinyi Zhou, Binghao Tang, Xiaocheng Gong, and Si Li. 2023. Chatgpt is a potential zero-shot dependency parser. *arXiv preprint arXiv:2310.16654*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Deep learning for ellipsis resolution. In *Proceedings of ACL 2016*. ACL.

Jason Merchant. 2001. *The Syntax of Silence: Sluicing, Islands, and the Theory of Ellipsis*. Oxford University Press.

Leif Arda Nielsen. 2004. *A Logical Approach to Ellipsis Resolution*. Ph.D. thesis, University of London.

Igor Sterner and Simone Teufel. 2025a. Code-switching and syntax: A large-scale experiment. *arXiv preprint arXiv:2506.01846*.

Igor Sterner and Simone Teufel. 2025b. Minimal pair-based evaluation of code-switching. *arXiv preprint arXiv:2506.01840*.

Yuanhe Tian, Fei Xia, and Yan Song. 2024. Large language models are no longer shallow parsers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7131–7142.

Ziyan Zhang, Yang Hou, Chen Gong, and Zhenghua Li. 2025. Self-correction makes llms better parsers. *arXiv preprint arXiv:2504.14165*.

A Distribution of SpokeBench

Table 3 reports the distribution of sentences in **SpokeBench** by dominant spoken code-switching phenomenon and overall structural complexity. The benchmark was designed to balance coverage of frequent spoken phenomena with a controlled set of highly complex constructions, enabling systematic evaluation of parsers under varied spoken-language conditions.

Category	**# Sentences**
Simple repetition	10
Complex repetition (CSW repetition, etc)	15
English contractions	10
Spanish contractions	10
Simple ellipsis	10
Complex ellipsis (break-of-thought, etc)	15
Simple discourse	10
Complex discourse (filler words, etc)	15
Highly complex (3+ phenomena)	12
No spoken phenomena (control)	20
Total	**126**

Table 3: Composition of the SpokeBench benchmark by dominant spoken code-switching phenomenon and complexity level.

B Examples of Spoken Code-Switching Phenomena

Table 4 provides representative examples of the spoken code-switching phenomena included in our taxonomy, drawn from the Miami Corpus. Each example illustrates a characteristic non-canonical structure encountered in conversational CSW, along with an English translation to support interpretability for non-Spanish speakers.

C Condensed Annotation Guidelines

C.1 General Annotation Principles

Annotations in SpokeBench follow UD conventions with targeted extensions for spoken, disfluent, and code-switched data, guided by the principles below:

- **UD-first consistency:** Annotators adhered to standard UD conventions unless a spoken-language phenomenon required an explicit, documented deviation.

Category	Example (Miami Corpus)	English Translation
Repetition	entonces then I will I will I will not be buying any stuff for you this weekend .	So then I will, I will, I will not be buying any stuff for you this weekend.
Discourse Elements	uhhuh bueno es verdad .	Uh-huh, well, it's true.
Ellipsis	pero si they are covered from .	But if they are covered from [it].
Contractions	and tú sabes it wasn't the same .	And you know it wasn't the same.
Compound Words	cuando uno va al swimming pool ahí no me gusta .	When one goes to the swimming pool, I don't like it there.
Break of Thought	you have you gotta show tu tía .	You have.. you gotta show your aunt.
Filler Words	fíjate que they gave them an honorary uh diploma .	Look, they gave them an honorary, uh, diploma.
Slang / Curse Words	I mean I'm thinking coño if I'm not here what the hell would happen to me ?	I mean, I'm thinking, damn, if I'm not here, what the hell would happen to me?
Enclisis	more jugo dile a Carla que te dé more jugo	More juice, tell Carla to give you more juice.

Table 4: Representative examples of spoken code-switching phenomena from the Miami Corpus, with English translations.

- **Single-root constraint:** Every sentence was required to contain exactly one syntactic root, including minimal, elliptical, or fragmentary utterances (e.g., *si mmh*). The root was assigned to the lexical item carrying the greatest semantic weight.
- **Ambiguity tolerance:** Structural ambiguity was treated as an inherent property of spoken CSW data rather than as an annotation error. Annotators were instructed to prefer linguistically plausible analyses over maximally specific ones.
- **Internal consistency:** All annotations were reviewed for internal consistency, with particular attention to agreement between UPOS tags and dependency relations.
- **Conservative tokenization:** Original corpus tokenization was preserved whenever possible. Token splitting or insertion of additional rows was permitted only when necessary to support accurate syntactic analysis.

C.2 Treatment of Spoken-Language Phenomena

This subsection summarizes the core annotation rules applied to recurring spoken-language phenomena in SpokeBench. These rules were designed to minimize speculative reconstruction while ensuring consistent and transparent treatment across annotators.

Repetition and Repairs Repeated tokens that function as speech repairs or disfluencies were annotated using the dependency relation `reparandum`. Annotators were instructed to identify the intended syntactic structure and attach repeated or overridden material as dependents of the corresponding head in the corrected structure. This procedure was applied uniformly to both monolingual and code-switched repetitions.

Contractions English and Spanish contractions (e.g., *wasn't*, *del*) were split into their component morphemes by inserting additional tokens immediately following the original form. The expanded tokens were then annotated according to standard UD conventions. This approach preserves morphological transparency while enabling accurate dependency assignment.

Multiword Expressions and Compounds Fixed expressions and conventionalized compounds (e.g., *you know*, *a lot*) were treated as single syntactic units rather than annotated compositionally. Annotators combined such expressions using underscore-separated tokens and assigned UPOS and dependency relations corresponding to their discourse or syntactic function.

Ellipsis and Fragmented Utterances Elliptical or truncated constructions were handled according to the recoverability of the intended meaning. Tokens that could be removed without altering the core interpretation of the utterance were labeled as `reparandum`. Tokens whose syntactic heads were missing and could not be confidently reconstructed were assigned the relation `dep` and attached to the sentence root. In fragmentary utterances lacking an explicit predicate, the most semantically salient token was designated as the root.

Discourse Markers and Fillers Discourse markers, interjections, and hesitation sounds (e.g., *uh*, *mmh*) were annotated with `UPOS=INTJ` and assigned the dependency relation `discourse`. Fillers were treated as a distinct subclass to facilitate later analysis but followed the same structural annotation principles.

C.3 Review and Adjudication Procedure

Each sentence in SpokeBench was independently annotated by at least two annotators. Annotations were subsequently reviewed using a graded acceptability scale that distinguished between acceptable variation and substantive errors. Reviewers were instructed to assess whether an analysis was linguistically defensible under UD rather than whether it matched their personal preference. Sentences flagged as borderline or unacceptable were escalated for expert discussion. Final annotations were determined through deliberation involving senior linguists on the project, ensuring consistency across the benchmark. This process resulted in a gold-standard dataset that reflects informed consensus while acknowledging the inherent ambiguity of spoken code-switched language.

D DECAP Interaction Protocol

The details of the DECAP agent workflow are shown below, in Algorithm 1.

Algorithm 1 DECAP per-sentence interaction rule

Require: Tokenized sentence $S = \{(t_i, \text{lang}_i)\}_{i=1}^{n}$
Ensure: UD parse with validated structure and confidence annotations
1: $\mathcal{S} \leftarrow \text{SPH}(S)$ {Detect spoken phenomena; propose tokenization edits}
2: $\mathcal{L} \leftarrow \text{LSR}(\mathcal{S})$ {Apply language-specific normalization and MWEs}
3: $\mathcal{C} \leftarrow \textsc{Core}(\mathcal{L})$ {Assign UPOS, heads, and dependency relations}
4: $\mathcal{F} \leftarrow \text{V/R}(\mathcal{S}, \mathcal{L}, \mathcal{C})$ {Enforce single root, acyclicity, and consistency}
5: $\mathcal{F}$ includes confidence scores and logs any structural repair
6: **return** $\mathcal{F}$

E FLEX-UD Severity Examples

Catastrophic Errors. Catastrophic errors correspond to violations that undermine the global structural validity or interpretability of a parse. These include cases where a dotted multiword expression (MWE) is required by the gold annotation but omitted by the system; reparandum tokens attached to unrelated subtrees, resulting in incorrect root selection or major structural distortion; and invalid head references that persist after canonicalization. Such errors receive large penalty contributions in FLEX-UD, typically in the range of $P = 0.25$ to 0.6 per issue (clipped), reflecting their disproportionate impact on parse quality.

Minor Errors. Minor errors correspond to linguistically plausible deviations that do not compromise the overall structure of the parse. Examples include tolerant POS substitutions (e.g., VERB↔AUX), near-miss dependency relations (e.g., `obj`↔`obl`), and other UPOS or DEPREL mismatches that fall within predefined closeness classes. These errors incur small penalty contributions, typically in the range of $P = 0.01$ to 0.05 per issue, allowing FLEX-UD to distinguish acceptable variation from substantive failure.

F Extended Results

The extended results for stanza-based monolingual and multilingual parsers are presented here. Table 5 presents performance under standard attachment-based metrics (LAS, UAS, CLAS, UPOS-LAS) for English-only, Spanish-only, and multilingual parsers across spoken CSW categories. Table 6 reports the corresponding FLEX-UD component scores and aggregated results for the same systems. These tables complement the main results by illustrating that the qualitative performance hierarchies and metric sensitivities discussed in Section 6 persist across parser types, even when overall accuracy differs substantially.

G Extended Linguistic Analysis

Bilingualism as a Complexity Multiplier The difficulty hierarchies observed for UPOS-LAS and LAS are systematically amplified in bilingual and code-switched input, suggesting that bilingualism acts as a complexity multiplier at both production and parsing levels. In spontaneous bilingual speech, challenges in lexical retrieval, grammatical alignment, and online planning frequently give rise to repetition, reformulation, ellipsis, and increased structural complexity, all of which reduce the recoverability of syntactic relations (Dobrovoljc, 2022a). Ellipsis often relies on discourse context or shared knowledge, leaving structure underspecified from a purely syntactic perspective, while repetition and reformulation introduce competing attachment sites despite surface redundancy.

Category	Eng Monolingual Parser				Spa Monolingual Parser				Multilingual Parser			
	LAS	UAS	CLAS	U-LAS	LAS	UAS	CLAS	U-LAS	LAS	UAS	CLAS	U-LAS
Repetition	0.08	0.18	0.06	0.26	0.15	0.30	0.04	0.63	0.16	0.34	0.09	0.62
Repetition+	0.07	0.19	0.05	0.38	0.21	0.36	0.11	0.57	0.23	0.42	0.13	0.60
Contr. (EN)	0.17	0.29	0.13	0.60	0.03	0.04	0.02	0.39	0.05	0.08	0.05	0.31
Contr. (ES)	0.02	0.06	0.01	0.24	0.30	0.42	0.28	0.72	0.11	0.24	0.08	0.40
Ellipsis	0.15	0.29	0.07	0.57	0.16	0.23	0.03	0.44	0.19	0.33	0.10	0.63
Ellipsis+	0.21	0.34	0.10	0.56	0.15	0.26	0.04	0.45	0.24	0.36	0.14	0.59
Discourse	0.21	0.32	0.18	0.56	0.08	0.17	0.04	0.50	0.24	0.37	0.16	0.59
Discourse+	0.23	0.32	0.17	0.53	0.15	0.21	0.08	0.34	0.32	0.38	0.23	0.61
Complex	0.12	0.16	0.12	0.30	0.17	0.25	0.06	0.55	0.20	0.29	0.10	0.52
None	0.13	0.17	0.07	0.26	0.22	0.29	0.15	0.60	0.29	0.36	0.19	0.64
Overall	**0.15**	**0.23**	**0.11**	**0.44**	**0.15**	**0.24**	**0.08**	**0.52**	**0.18**	**0.29**	**0.12**	**0.53**

Table 5: Performance of traditional parsers across spoken code-switching categories under standard UD metrics.

Category	Eng Monolingual					Spa Monolingual					Multilingual				
	ID	UPOS	HEAD	DEPREL	Final	ID	UPOS	HEAD	DEPREL	Final	ID	UPOS	HEAD	DEPREL	Final
Repetition	10.7	10.7	10.7	10.7	25.6	61.5	65	47.5	49	58.9	10.7	4.8	4.8	4.8	22.1
Repetition+	12.6	12.6	12.6	13	28.7	69.3	57	52.6	48.3	60.6	6.73	6.73	6.73	7.06	24
Contr. (EN)	31.6	27.3	27.3	27.3	35.6	45	38.5	29.5	31.5	38.9	17.3	17.3	17.3	17.3	26.4
Contr. (ES)	34.7	34.7	34.7	34.7	39.6	57.7	74.4	62.7	63.8	63.3	18.3	18.3	18.3	18.3	26.4
Ellipses	12.5	7.18	7.18	7.18	25	70.9	52.2	45.4	47.7	58.8	6.09	4.27	4.27	4.27	21.3
Ellipses+	5.53	5.53	5.53	5.53	23.3	73.0	51.5	43.8	43.4	60.8	5.53	5.53	5.53	5.53	23.3
Discourse	8.7	8.7	8.7	8.7	23.3	77.5	52	32	36.5	50.8	9.8	9.8	9.8	9.8	25.2
Discourse+	32.8	32.8	32.8	32.8	42.8	57.3	41.3	31	30.3	41.6	31.0	31.0	31.0	31.0	41.5
Complex	9.23	9.92	9.23	9.23	24.5	60	54.6	45.7	45.7	54.3	10.7	11.4	10.7	10.7	26.2
None	13.7	8.75	8.75	8.75	27.1	81.5	70.1	52	53.2	66.5	8.25	8.25	8.25	8.25	25.6
Overall	17.2	15.8	15.7	15.7	29.5	65.3	55.6	44.2	44.9	55.4	12.4	11.7	11.6	11.7	26.2

Table 6: Component-wise evaluation (ID, UPOS, HEAD, DEPREL and aggregated Final) across spoken-CSW categories for three parser classes.

These phenomena do not introduce new classes of syntactic difficulty but intensify existing sources of ambiguity. The consistent internal hierarchies observed across bilingual and multilingual parsers, despite differences in overall accuracy, indicate that these effects are driven by properties of the input rather than parser-specific weaknesses. LLM-based parsers make this interaction particularly visible, exhibiting sharper performance drops for repetition and ellipsis than for canonical constructions. Overall, bilingualism reshapes the distribution and salience of non-canonical structures, magnifying the impact of ambiguity and recoverability on parsing performance.

H Prompts Used for DECAP

The prompt structures for the 4 agents used in the DECAP pipeline are described below. Each prompt corresponds to a dedicated agent with a narrowly scoped responsibility in the annotation process, progressing from spoken-phenomena detection to language-specific normalization, core UD assignment, and final verification and ranking.

Prompt for Spoken-Phenomena Handler (SPH)

System Prompt (SPH).
You are the **Spoken-Phenomena Handler (SPH)** for Spanish–English conversational data. Your responsibility is to detect spoken-language phenomena, propose minimal tokenization edits (e.g., contraction splits and multiword expressions), and produce a single structured JSON object for downstream agents. Return *only* valid JSON and behave deterministically.

User Prompt (SPH).
Input: A JSON array of tokens representing one sentence, including sentence ID, token index, surface form, and language tag.
Goal: Produce a validated SPH JSON object that:

- identifies spoken-language phenomena (e.g., repetition, ellipsis, discourse, fillers),
- proposes necessary tokenization edits (contractions, enclisis, MWEs),
- constructs a complete and consistent proposed ID mapping, and
- reports a brief summary and confidence score.

Key Rules (excerpt):

- Contractions and enclitic forms are split only when they correspond to multiple UD nodes.
- Fixed multiword expressions are collapsed into dotted nodes when they function as a single syntactic unit.
- Repetitions and repairs are marked as `reparandum` and linked to the intended head.
- Elliptical or stranded tokens are labeled conservatively and anchored to the root when necessary.

Additional rules governing ID assignment, mandatory MWEs, validation checks, and output schema are provided in the full prompt specification in the GitHub (. . .).

Output: A single JSON object containing normalized tokens, spoken-phenomena labels, a proposed ID map, confidence scores, and brief summary notes. No explanatory text is permitted.

Prompt for Language-Specific Resolver (LSR)

System Prompt (LSR).
You are the **Language-Specific Resolver (LSR)** for Spanish–English code-switched conversational data. Your single responsibility is to accept the SPH JSON and apply conservative language-specific normalization (contraction/enclisis expansion, MWE confirmation/creation, lemma suggestions, and language-confidence scores). Respect SPH edits; return *only* one JSON object and behave deterministically.

Input: The SPH JSON object with keys including `sentence_id`, `original_tokens`, `tokens`, and `proposed_id_map`.
Goal: Produce an updated JSON that:

- confirms or conservatively expands standard contractions/enclitics (INTEGER-SHIFT),
- confirms or (rarely) creates high-precision MWEs as dotted nodes (whitelist + conservative rule),
- supplies lowercase `lemma` suggestions and `lsr_confidence` per token,
- updates `proposed_id_map` consistently for any insertions.

Key Rules (excerpt):

- **Respect SPH authority**: do not undo SPH tokenization edits; only add splits when SPH omitted a clear, standard contraction/enclitic.
- **Contraction handling (INTEGER-SHIFT)**: expand only standard contractions that map to separate UD nodes; insert new integer proposed_IDs and shift subsequent IDs +1, reflecting changes in `proposed_id_map`.
- **MWE creation (DOTTED IDs)**: auto-combine mandatory whitelist MWEs (case-insensitive). Create other MWEs only under conservative, high-confidence criteria; add dotted node `<start>.1` placed after the span.
- **Lemmas & language confidence**: provide best-effort lemmas (verbs $\rightarrow$ infinitive, nouns $\rightarrow$ singular) or `null`; set `lsr_confidence` in [0.0,1.0] and one-line `lsr_notes` for nontrivial edits.
- **IDs and validation**: do not change original `token_index`; ensure unique proposed_IDs, complete `proposed_id_map`, and consistency after integer-shifts.

Examples (excerpt):

- **Contraction split**: `"don't"` $\rightarrow$ `"do"` (orig id) + inserted `"not"` (new id); update map so orig id maps to `["5","6"]`.
- **MWE (whitelist)**: `"pitta bread"` $\rightarrow$ keep 6,7 integer rows and add dotted 6.1 with `split_token="pitta_bread"`; reflect in map.

Additional details on whitelist entries, edge cases, full schema, and validations are available in the full prompt spec on the project repository (...).

Output: A single JSON object with keys `"sentence_id"`, `"tokens"` (including `proposed_ID`, `split_token`, `lang_tag`, `lemma`, `lsr_notes`, `lsr_confidence`, `mwe`), `"proposed_id_map"`, and `"summary_notes"`. No additional text is permitted.

Prompt for Core UD Assigner (Core)

System Prompt (Core).
You are the **Core UD Assigner** for Spanish–English conversational data. Your task is to assign UD-style annotations (UPOS, HEAD_ID, DEPREL) following the Miami Gold Subset spoken-language rules. Respect tokenization and proposed_IDs from upstream (SPH + LSR) and spoken-language directives (spoken_label / spoken_anchor). Annotate only dotted MWE nodes (n.1); leave MWE component integer rows unannotated. Ensure exactly one root. Return *only* a single JSON object (no commentary).

User Prompt (Core).
Input: LSR JSON object with keys: `sentence_id`, `tokens` (`proposed_ID`, `split_token`, `lang_tag`, `lemma`, `lsr_confidence`, `mwe`, `spoken_label`, `spoken_anchor`), and `proposed_id_map`.
Goal: Produce an annotated JSON with UD fields for every annotatable node, obeying spoken-label mappings and UD constraints.

Key Principles (high-level):

- **Respect upstream**: follow SPH/LSR spoken_label and recommended head (spoken_anchor) when provided.
- **Spoken-label → UD mapping**: e.g., `reparandum` → DEPREL=`rep`; `dep` → DEPREL=`dep`; `discourse/filler` → UPOS=`INTJ` and DEPREL=`discourse` (attach to root unless syntactically integrated).
- **Single root**: exactly one token with `HEAD_ID="0"` and `DEPREL="root"`; prefer finite VERB, else central NOUN/PRON, else communicative token.
- **Allowed tag sets**: UPOS and DEPREL must be chosen from the Miami-approved lists (use only allowed values).

Output Schema (required, single JSON object):

```
{
 "sentence_id": "<id>",
 "annotated_tokens": [
   {
     "proposed_ID": "<str>",
     "FORM": "<split_token or empty>",
     "LEMMA": "<lemma or empty>",
     "UPOS": "<allowed UPOS or empty>",
     "HEAD_ID": "<proposed_ID or '0' or empty>",
     "HEAD_FORM": "<FORM of head or 'root' or empty>",
     "DEPREL": "<allowed DEPREL or empty>",
     "core_confidence": 0.0-1.0,
     "core_notes": "<one-line justification>"
   }, ...
 ],
 "summary_notes": "<one-line summary>"
}
```

Essential validations (must pass): ...

Return: exactly one JSON object following the schema above. Do not include any additional text.

Prompt for Verifier & Ranker (V/R)

System Prompt (V/R).
You are the Verifier & Ranker. Merge SPH, LSR and Core outputs for one sentence; enforce ID/HEAD consistency; repair structural issues (single root, cycles); compute per-token `final_confidence` and `penalty`; remap proposed_IDs to sheet integers when requested; and emit sheet-ready rows plus an adjudication log. Respect authoritative ordering: Core ▷ LSR ▷ SPH. Return *only* the JSON described below. **User Prompt (v/R).**

Input & Goal.
Input: JSON bundle with keys `"sph"`, `"lsr"`, `"core"`. Goal: produce validated final token table (all integer rows, inserted integers, dotted MWE nodes), mapping to sheet IDs, plus `adjudication_log` and a one-line `final_summary`.

Condensed procedure (deterministic):

- Canonicalize nodes from SPH original order, inserting LSR/SPH-added integers and dotted MWEs.
- Merge annotations (prefer Core; fill from LSR then SPH).
- Validate/remap HEADs (HEAD_FORM match, then SPH spoken_anchor, else attach to root) and log.
- Enforce single root by priority (finite VERB, AUX, central NOUN/PRON, else highest confidence) and log.
- Detect repair cycles using combined_conf; reattach lowest node(s) to root until acyclic; log numeric rationale.
- Compute final_confidence and penalty (per rules) and add adjudication notes.
- Remap to sheet_IDs (1..N) and produce sheet_HEAD_ID mapping.
- Emit final JSON and validate

Required output shape (exact):

```
{
 "sentence_id":"<id>",
 "final_tokens":[
   {
     "sentence_id":<id>,
     "orig_token_index":<int>,
     "split_token":"<str>",
     "ID":"<proposed_ID>",
     "sheet_ID":<int>,
     "FORM":"<str or blank>",
     "LEMMA":"<str or blank>",
     "UPOS":"<or blank>",
     "HEAD_ID":"<proposed_ID or '0'>",
     "sheet_HEAD_ID":<int or 0>,
     "HEAD":"<HEAD_FORM or blank>",
     "DEPREL":"<or blank>",
     "final_confidence":0.0-1.0,
     "penalty":0.0-1.0,
     "adjudication_note":"<short>"
   },
   ...
 ],
 "adjudication_log":[ "...", "..." ],
 "final_summary":"<one-line>"
}
```

Determinism & logging: use deterministic tie-breakers (prefer lower sheet_ID). Each structural fix must add one bullet to `adjudication_log` stating what changed, why (include numeric confidences), and which signals supported the decision.