Parsing Arabic Dialects Revisited: New Benchmarks, Models, and Insights

Ahmed Elshabrawy,[‡*] Go Inoue,[†,‡*] Muhammed AbuOdeh,[‡] Nizar Habash[†,‡]
[†]Computational Approaches to Modeling Language (CAMeL) Lab,
New York University Abu Dhabi (NYUAD)
[‡]Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)
{ahmed.elshabrawy,go.inoue,muhammed.abuodeh}@mbzuai.ac.ae, nizar.habash@nyu.edu

Abstract

Parsing dialectal Arabic remains underexplored, with limited progress over the past two decades. Existing Modern Standard Arabic (MSA) parsers perform poorly on dialectal data, motivating the need for dialect-specific approaches. We revisit this task using modern neural models and present new results on Egyptian and Gulf Arabic dependency parsing. We demonstrate that even small amounts of dialectal training data yield substantial improvements in parsing accuracy. Our contributions include: (1) introducing a new annotated dataset for Gulf Arabic, (2) releasing a state-of-the-art multi-variety Arabic parser, and (3) employing dialect identification as a diagnostic tool to better understand how training data affects parsing performance across dialects and test sets.

Keywords: Arabic Dependency Parsing, Dialectal Arabic, Gulf Arabic Dataset, Egyptian Arabic, Language Resources, Annotated Corpora, Dialect Identification, Open-Source NLP Tools, Modern Standard Arabic (MSA)

1. Introduction

Arabic dialects, such as Egyptian (EGY), Gulf (GLF), Levantine, and Maghrebi, are the dominant spoken varieties across the Arab world. They pervade informal communication, social media, and everyday discourse, yet remain underrepresented in computational linguistics (Darwish et al., 2021). In contrast, Modern Standard Arabic (MSA) has long benefited from extensive linguistic resources and treebanks (Maamouri et al., 2004; Habash and Roth, 2009; Taji et al., 2017; Habash et al., 2022). As Table 1 shows, even strong MSA parsers like CamelParser2.0 (Elshabrawy et al., 2023) degrade significantly on dialectal input, underscoring the need for dialect-focused approaches.

While large language models (LLMs) raise questions about the role of traditional parsing, syntactic analysis remains crucial for tasks requiring interpretability and structure, especially in education, grammar-aware tools, and low-resource settings (Guo et al., 2024). Parsing also offers reliable scaffolding for downstream tasks such as error detection and information extraction, where LLMs may lack consistency or transparency (Kanayama et al., 2024; Li et al., 2025; Fan et al., 2025).

This paper revisits Arabic dialectal parsing, a task largely neglected for over two decades (Chiang et al., 2006). We adopt a modern neural dependency parsing architecture to build a state-of-the-art open-source parser for multiple variants of Arabic.

Our main contributions are: (a) the introduction of a **new annotated Gulf Arabic dependency treebank**, enabling the first parsing results for this dialect; (b) the release of a **state-of-the-art multi-variety Arabic parser (MSA, EGY, GLF)** (Table 1); and (c) the use of **dialect identification as a diagnostic tool** to assess parsing complexity and data impact.[1]

System	MSA	EGY	GLF	AVG
CamelParser2.0	**87.5**	73.3	72.7	77.8
Ours	87.2	**84.2**	**80.3**	**83.9**

Table 1: Labeled Attachment Score (LAS) performance of the MSA-trained CamelParser2.0 and of our system on MSA, EGY, GLF test sets.

2. Related Work

2.1. Arabic Treebanks

A number of Arabic treebanks have been developed, differing in size, syntactic formalism, and genre coverage. Among these, the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) has been the primary resource for MSA syntactic analysis. The PATB consists mainly of newswire and web texts. Although originally annotated using a phrase-structure representation, it has been converted into several dependency formalisms (Smrž et al., 2002; Habash and Roth, 2009; Taji et al., 2017).

Building on PATB and related efforts, several MSA dependency treebanks have been developed.

* Equal contribution.

[1]We make our code and data available at https://github.com/CAMeL-Lab/camel_parser_dialects.

Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks, pages 94–105
May 11, 2026.

PADT was the first Arabic dependency treebank and employs a multi-layered annotation scheme capturing functional morphology, analytical dependency syntax, and tectogrammatical representations of meaning (Smrž et al., 2002). The Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009) introduced a streamlined dependency representation inspired by traditional Arabic grammar. The Quranic Arabic Corpus (Dukes and Buckwalter, 2010) provided a hybrid dependency–constituency annotation of Quranic scripture. The Arabic Poetry Treebank (ArPoT) (Al-Ghamdi et al., 2021) applies the CATiB formalism, with extensions, to Classical Arabic poetry. The *i3rab* treebank (Halabi et al., 2021) adopts a dependency representation closely aligned with traditional Arabic grammatical theory. The Camel Treebank (CamelTB) (Habash et al., 2022) has further expanded MSA dependency resources across multiple genres and time periods.

In addition to MSA, several dialectal treebanks have been developed. The Linguistic Data Consortium (LDC) released dialectal resources derived from conversational transcripts, including the Levantine Arabic Treebank (LATB) (Maamouri et al., 2006) and the Egyptian Arabic Treebank (ARZTB) (Maamouri et al., 2014). More recently, a 22K-token Algerian Arabic treebank was introduced by Seddah et al. (2020).

2.2. Parsing Arabic and its Dialects

Parsing efforts in Arabic have predominantly focused on MSA, largely due to the presence of rich annotated datasets. Early work on MSA parsing includes Habash and Roth (2009), where they built a transition-based dependency parser using MaltParser (Nivre et al., 2006). This work was extended in subsequent systems with improved features, such as those by Marton et al. (2013) and Shahrour et al. (2016). Recent work on neural Arabic dependency parsing has explored a variety of modeling approaches, including multi-task easy-first dependency parsing (Kankanampati et al., 2020), dependency parsing as a sequence labeling task (Al-Ghamdi et al., 2023), and graph-based biaffine dependency parsing (Elshabrawy et al., 2023).

Dialectal Arabic parsing, however, has received limited attention. Early pioneering work by Chiang et al. (2006) explored a treebank transduction approach, leveraging MSA resources for parsing Levantine Arabic. Muller et al. (2020) investigated the transferability of multilingual language models to Algerian Arabic for dependency parsing. Abo Mokh et al. (2024) evaluated the performance of an MSA-trained parser on a small set of sentences in Gulf, Levantine, Egyptian, and Maghrebi dialects. In this work, we present empirical studies on building a state-of-the-art dialectal dependency parser for Egyptian and Gulf Arabic.

In this work, we adopt the CATiB dependency representation across MSA, EGY, and GLF for consistency. For MSA, we use CamelTB and train on both the CATiB-converted PATB and CamelTB data. For EGY, we automatically convert ARZTB into CATiB style. For GLF, we introduce the first dependency treebank for the dialect, consisting of over 25K tokens, manually annotated following CATiB guidelines with minor extensions. All evaluations are conducted in the CATiB framework.

2.3. Dialect Identification

Dialect Identification (DID) is the task of predicting the dialectal variety of a given speech or text segment (Etman and Beex, 2015). There has been a growing interest in DID, reflected in a series of shared evaluation campaigns, including MADAR and NADI (Bouamor et al., 2019; Abdul-Mageed et al., 2021, 2024), and the development of various datasets and tools (Zaidan and Callison-Burch, 2011; Bouamor et al., 2014; Salama et al., 2014; Alsarsour et al., 2018; Abu Kwaik et al., 2018; Zaghouani and Charfi, 2018; Salameh et al., 2018; Bouamor et al., 2019; Abdelali et al., 2021; Baimukan et al., 2022).

DID has been shown to improve downstream NLP performance by enabling dialect-aware system selection or adaptation, including in machine translation (Salloum et al., 2014) and morphological tagging (Obeid et al., 2022). In this work, we adopt DID not as an end task, but as an analytical tool to obtain finer-grained evaluation subsets, allowing us to better understand cross-variety parsing behavior and training data effects.

3. Arabic Syntax Representation

We follow the **Columbia Arabic Treebank (CATiB)** annotation scheme (Habash et al., 2009, 2022). Grounded in traditional Arabic grammar yet deliberately simplified, CATiB is intuitive for native speakers and well-suited for consistent annotation across Arabic varieties. Figures 1 and 2 are examples of CATiB annotated trees. Next is a very compact summary of the CATiB representation.

In terms of **tokenization**, CATiB segments all clitics except the definite article *Al+* 'the', and normalizes basewords to their uncliticized form.

CATiB uses six **part-of-speech** tags: *NOM* (common nominals), *PROP* (proper nouns), *VRB* (active verbs), *VRB-PASS* (passive verbs), *PRT* (particles, including prepositions and conjunctions), and *PNX* (punctuation). This compact tagset supports efficient annotation while maintaining syntactic adequacy.

As for **dependency relations**, CATiB uses eight: *SBJ* (subjects of verbs and topics of simple nominal

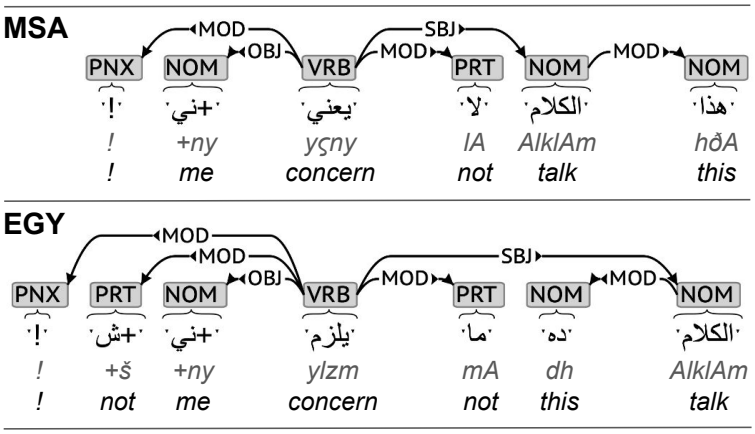

Figure 1: Parallel MSA and EGY CATiB dependency trees for 'this talk does not concern me.' Words are shown with transliteration and gloss.

Dataset	#Sentences				#Tree Tokens			
	ALL	TRAIN	DEV	TEST	ALL	TRAIN	DEV	TEST
PATB (Maamouri et al., 2004)	19,738	15,789	1,986	1,963	738,889	590,819	73,945	74,125
CamelTB (Habash et al., 2022)	13,337	9,397	2,022	1,918	241,910	171,735	35,418	34,757
ARZTB (Maamouri et al., 2014)	31,688	24,428	3,542	3,718	508,548	393,193	57,360	57,995
CamelTB-Gumar (Ours)	2,881	2,021	435	425	25,516	17,648	4,112	3,756
Total	67,644	51,635	7,985	8,024	1,514,863	1,173,395	170,835	170,633

Table 2: Dataset statistics for all treebanks used in this study. For each dataset, we report the total number of sentences and tree tokens (ALL), together with their counts in the TRAIN, DEV, and TEST splits. Data splits follow Diab et al. (2013) for PATB and ARZTB and Habash et al. (2022) for CamelTB.

sentences), *OBJ* (objects of verbs, prepositions, or deverbal nouns), *TPC* (topics of complex nominal sentences with explicit pronominal reference), *PRD* (predicative complements in extended copular constructions), *IDF* (the *idafa* possessive construction), *TMZ* (the *tamyiz* specification construction), **MOD** (general modification of verbs or nominals), and — (flat constructions, e.g., multiword proper names).

Figure 1 shows parallel MSA and EGY CATiB analyses. Both have five orthographic words, but clitic segmentation produces six tokens in MSA and seven in EGY. The trees show structural similarities, such as SVO order (though MSA often uses VSO), preverbal negation, and direct object pronominal clitics. They also highlight differences, including EGY suffix negation (ش+ *+š* 'not') and postposed demonstratives (ده *dh* 'this').

4. Arabic Treebanks

In this paper, we make use of four Arabic treebanks: two MSA, and two dialectal (one EGY and one GLF). The latter (CamelTB-Gumar) is a new treebank we introduce as part of this work and make publicly available. Table 2 shows tree token counts across splits for all treebanks. Two of the treebanks (one MSA and EGY) were converted from constituency representation to CATiB dependency; and two were created directly in CATiB dependency representation. We discuss each of these resources next, and finish the section with a comparison of lexical and structural differences among them.

4.1. PATB (MSA Newswire)

The Penn Arabic Treebank (PATB) introduced earlier in Section 2 is the largest Arabic treebank to date.[2] PATB focuses on MSA for newswire and was created in constituency representation (Maamouri et al., 2004). Following Habash and Roth (2009), we use an automatically converted version to CATiB created with the Arabic-C2D utility.[3] We use the data splits recommended by Diab et al. (2013). The same conversion and splits are used in CamelParser 1.0 and 2.0 (Shahrour et al., 2016; Elshabrawy et al., 2023). In this paper we only use PATB as part of the training.

[2] LDC2010T13, LDC2011T09, LDC2010T08

[3] Arabic-C2D GitHub repository.

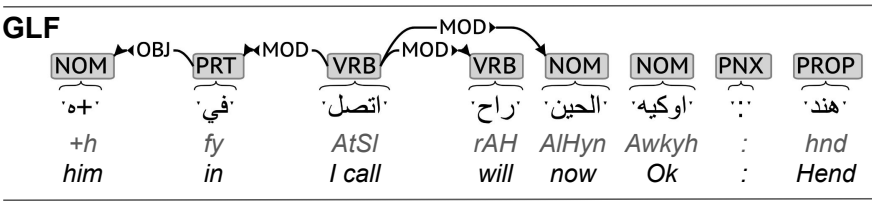

Figure 2: GLF CATiB tree for 'Hend: OK, I'll call him now.' Words are shown with transliteration and gloss.

4.2. CamelTB (MSA Mixed Genres)

The Camel Treebank (CamelTB) includes 13 subcorpora comprising selections of texts from pre-Islamic poetry to social media online commentaries, and covering a range of genres from religious and philosophical texts to news, novels, and student essays (Habash et al., 2022). The CamelTB was annotated directly in CATiB representation. We use the data split recommended by Habash et al. (2022). The CamelTB Train is used with PATB Train as the MSA Train baseline, as used in CamelParser2.0. The CamelTB Dev and Test sets are our only MSA test sets. We chose to focus on CamelTB as *MSA* for testing because of its genre diversity.

4.3. ARZTB (EGY Forums & Chat)

For Egyptian Arabic, we use the Egyptian Arabic Treebank (ARZTB) (Maamouri et al., 2014), a constituency representation treebank containing data from discussion forums, text messaging, and chat.[4] We converted to the CATiB formalism following the approach of Habash and Roth (2009) discussed above for PATB.[3] We use the data splits recommended by Diab et al. (2013). We make our Arabic-C2D conversion map publicly available.

4.4. CamelTB-Gumar (GLF Internet Novels)

We introduce the Gulf Treebank (CamelTB-Gumar), the first dependency treebank for Gulf Arabic, annotated in the Columbia Arabic Treebank (CATiB) scheme (Habash and Roth, 2009), and following the updated guidelines used in the Camel Treebank (Habash et al., 2022). The CamelTB-Gumar data comprises the development set of the 200K-word morphologically annotated portion of the Gumar Corpus (Khalifa et al., 2016, 2018). We define our own splits for CamelTB-Gumar following Diab et al. (2013)'s guidelines. We make the treebank and splits publicly available.

We leverage the existing gold tokenization and POS tags to create syntactic annotations, starting from automatic parses generated by CamelParser1.0 (Shahrour et al., 2016), trained on PATB data. Annotation was carried out by a team of native speakers with extensive experience in linguistic annotation, hired through a professional annotation firm and fairly compensated. An expert Arabic native speaker linguist manually reviewed and corrected 125 sentences (1,218 tokens) from the training data, representing ∼5% of the full corpus. The resulting agreement scores were very high: POS accuracy reached 99.9%, while syntactic scores were 99.0% UAS, 99.2% LS, and 98.9% LAS. These results indicate that the annotations are of excellent quality.

The CamelTB-Gumar differs markedly from other Arabic treebanks due to its genre: conversational internet novels. The text contains frequent fragments, dialogue formatted as *(Name: Speech)*, ellipsis, elided vocatives, missing punctuation, and interrupted phrases, making it syntactically more challenging and distinct from formal newswire treebanks. Figure 2 illustrates a typical example, where a minimally punctuated sentence is analyzed as four fragments under a single root.

[4] LDC2018T23

	MSA	EGY	GLF
PATB (**MSA**)	**91.2** / 92.6	82.3 / 96.4	65.5 / 74.1
CamelTB (**MSA**)		85.7 / **97.4**	72.0 / 87.9
ARZTB (**EGY**)			75.7 / 83.9

Table 3: Pairwise cosine similarity (%) across treebanks (**Lexical** / **Structural**). Columns correspond to reference treebanks: MSA (CamelTB), EGY (ARZTB), and GLF (CamelTB-Gumar).

4.5. Treebank Analysis

We compare the four treebanks in terms of lexical and structural similarity. Table 3 reports pairwise cosine similarity over lexical distributions and structural dependency bigrams.

Lexical Similarity The two MSA treebanks (PATB and CamelTB) are the most similar lexically (91.2%), while EGY treebank shows moderate similarity to MSA treebanks (82–86%). The GLF treebank is the most divergent variety, with similarity to MSA as low as 65.5%. Overall, the results indicate

a lexical gradient (MSA > EGY > GLF), suggesting increasing difficulty for cross-treebank transfer involving GLF.

Clitic distributions further illustrate these lexical differences. Core clitics such as +و *w*+ 'and', +ه *h*+ 'his', +ب *b*+ 'with/by', and +ل *l*+ 'to/for' are frequent across all treebanks.Dialect-specific clitics appear prominently in ArzTB, including the future marker +ح *H*+ 'will' and the negation suffix +ش *š*+ 'not', which are absent from MSA and GLF. Conversely, GLF exhibits distinctive forms such as +ج *j*+ 'you/your [f.s.]' and +ش *š*+ 'what' that do not occur in the other datasets.

Structural Similarity For structural comparison, we compute cosine similarity over the frequencies of 235 dependency bigram types (e.g., *VRB-SBJ-NOM*) in the union of all treebanks. Table 3 shows that structural similarity is generally higher than lexical similarity. ARZTB is highly similar to both MSA treebanks (96–97%), even exceeding the similarity between PATB and CamelTB. This may reflect the broader genre and stylistic diversity of CamelTB and ARZTB, compared to the more homogeneous PATB (newswire) and CamelTB-Gumar (internet novels). GLF remains the most distinct variety, though its structural distance (74–88%) is less pronounced than its lexical distance. Overall, the results suggest that cross-treebank variation is driven more by lexical differences than by major syntactic restructuring.

Frequent structural tuples further support this pattern. Core configurations such as *PRT-OBJ-NOM*, *NOM-IDF-NOM*, and *VRB-OBJ-NOM* are almost categorically head-first across all treebanks, indicating stable word order. Greater variation appears in subject and modifier relations: for example, *VRB-SBJ-NOM* is predominantly head-first in MSA (80–85%) but much less so in the dialectal treebanks (36–39%). Similarly, nominal verb modifier relations (e.g., *VRB-MOD-NOM*) show stronger head-first tendencies in MSA treebanks than in the dialectal treebanks. These differences point to subtle shifts in syntactic realization in the dialects, despite overall structural similarity.

5. Experimental Setup

In this section, we describe the data, model configurations, and evaluation protocol used to study multi-variety Arabic parsing.

5.1. Data

Evaluation Data Table 4 shows the development and test splits we use in our experiment. For MSA, we use the CamelTB development and test sets and exclude PATB because CamelTB is more diverse (genre-wise) and therefore more representative of MSA evaluation. For EGY (ARZTB), we follow the recommended splits of Diab et al. (2013). For GLF (CamelTB-Gumar), we use our predefined splits (Section 4.4).

		#Sentences	#Tree Tokens
DEV	**MSA**	2,022	35,418
	EGY	3,542	57,360
	GLF	435	4,112
TEST	**MSA**	1,918	34,757
	EGY	3,718	57,995
	GLF	425	3,756

Table 4: **DEV** and **TEST** set statistics used in our evaluation. MSA refers to CamelTB, EGY to ARZTB, and GLF to CamelTB-Gumar. We report the number of sentences and tree tokens for each variant.

Training Data	#Sentences	#Tree Tokens
MSA$_{\{\text{PATB, CamelTB}\}}$	25,186	762,554
EGY$_{\text{ARZTB}}$	24,428	393,193
GLF$_{\text{CamelTB-Gumar}}$	2,021	17,648
MSA+EGY	49,614	1,155,747
MSA+GLF	27,207	780,202
EGY+GLF	26,449	410,841
MSA+EGY+GLF	51,635	1,173,395

Table 5: **TRAIN** data statistics for the MSA (PATB and CamelTB), EGY (ARZTB), and GLF (CamelTB-Gumar) treebanks, as well as their aggregated combinations. We report the total number of sentences and dependency tree tokens used in each setup.

Training Data and its Combination For training, we use PATB and CamelTB train splits as a base representative for MSA training. This matches the training setup of CamelParser2.0. For EGY (ARZTB), we follow the recommended training splits of Diab et al. (2013). For GLF (CamelTB-Gumar), we use our predefined training splits (Section 4.4). Furthermore, we examine how training data composition affects cross-variety parsing through controlled comparisons. We train single-variety models (MSA, EGY, GLF) as in-domain baselines; two-variety models (MSA+EGY, MSA+GLF, EGY+GLF) to assess mixed setups; and an all-variety model (MSA+EGY+GLF) to test joint training. Architecture and hyperparameters are fixed; only the data mix varies. Training sizes are shown in Table 5.

Parameter	Value	Parameter	Value
bert_pooling	mean	mlp_dropout	0.33
clip	5.0	mu	0.9
encoder_dropout	0.1	n_arc_mlp	500
epochs	10	n_bert_layers	4
eps	1e-08	n_rel_mlp	100
fix_len	20	nu	0.999
lr	5e-5	update_steps	4
lr_rate	10	warmup	0.1
min_freq	2	weight_decay	0
mix_dropout	0.0		

Table 6: Hyperparameters used to train our parsing models.

5.2. Dependency Parser

We adopt a biaffine dependency parser (Dozat and Manning, 2017), implemented in SuPar (Zhang et al., 2020), using BERT-based embeddings as input. As a baseline comparison point, we include CamelParser2.0 (Elshabrawy et al., 2023), which uses CAMeLBERT-MSA (Inoue et al., 2021) as its contextual encoder backbone. In our models, we instead use CAMeLBERT-MIX for contextual embeddings, as it is better suited for handling multiple Arabic variants, following the recommendations of Inoue et al. (2021).

For training, we use the hyperparameter settings of CamelParser2.0 (Elshabrawy et al., 2023), summarized in Table 6 for reproducibility. All models are trained and evaluated on a single NVIDIA V100 GPU, with a total computational cost of approximately 200 GPU minutes. Each parsing model has approximately 110M parameters.

5.3. Dialect Identification for Subset-Level Evaluation

Arabic is inherently diglossic (Ferguson, 1959), and even datasets labeled as a particular dialect often contain a mixture of text from different variants. To enable dialect-specific analysis of parsing results, we estimate the distribution of predicted dialect labels using the dialect identification system of Salameh et al. (2018) implemented in CAMeL Tools (Obeid et al., 2020). We use MODEL-26 with regional-level labels, achieving an F_1 score of 84.0 on the evaluation set reported by Salameh et al. (2018). We select the dialect label with the highest predicted probability among the three target variants: MSA, EGY, and GLF.

5.4. Evaluation Metric

We report Labeled Attachment Score (LAS), defined as the percentage of tokens for which both the predicted head (parent) and the dependency relation label are correct.

		$\lvert\mathcal{S}_{\text{ALL}}\rvert$	$\%\mathcal{S}_{\text{MSA}}$	$\%\mathcal{S}_{\text{EGY}}$	$\%\mathcal{S}_{\text{GLF}}$
Dev	**MSA**	2,022	**63.4**	2.6	34.0
	EGY	3,542	8.4	**55.3**	36.3
	GLF	435	0.5	3.0	**96.6**
Test	**MSA**	1,918	**64.6**	2.9	32.5
	EGY	3,718	10.1	**55.6**	34.3
	GLF	425	0.7	4.7	**94.6**

Table 7: Distribution of dialectal varieties in the development and test sets. For each evaluation subset (MSA, EGY, GLF), we report the total number of sentences ($\lvert\mathcal{S}_{\text{ALL}}\rvert$) and the percentage of sentences identified as MSA, EGY, and GLF ($\%\mathcal{S}_{\text{MSA}}$, $\%\mathcal{S}_{\text{EGY}}$, $\%\mathcal{S}_{\text{GLF}}$).

6. Results

In this section, we present parsing results across MSA, Egyptian (EGY), and Gulf (GLF) Arabic under different training configurations. We begin with dialect identification statistics to motivate subset-level evaluation, then compare single- and multi-variety models on DEV and TEST, and conclude with a detailed comparison against CamelParser2.0.

6.1. Dialect Identification

Table 7 presents the proportion of sentences predicted as MSA, EGY, and GLF in our evaluation sets. EGY (ARZTB) is mostly predicted as EGY, with a notable portion labeled as GLF. GLF (CamelTB-Gumar) is almost entirely predicted as GLF, while MSA (CamelTB) is largely predicted as MSA. These trends suggest that the predicted labels generally align with the expected dialectal focus of each dataset, while highlighting some degree of variety and possibly genre mixing, especially in ARZTB. When considering all datasets combined, the predicted dialect distribution is nearly balanced among the three variants. Given this, we evaluate performance by predicted dialect rather than by treebank, using subsets of sentences identified as MSA, EGY, and GLF (denoted $\mathcal{S}_{\text{MSA}}$, $\mathcal{S}_{\text{EGY}}$, and $\mathcal{S}_{\text{GLF}}$, respectively). For example, $\mathcal{S}_{\text{MSA}}$ denotes the set of sentences in a given evaluation dataset predicted as MSA, while $\mathcal{S}_{\text{EGY}}$ and $\mathcal{S}_{\text{GLF}}$ denote those predicted as EGY and GLF.

6.2. Dependency Parsing

DEV Set Results Table 8 reports LAS scores on the DEV sets under different training configurations. Overall, the full-data model (MSA+EGY+GLF) achieves the highest macro-average performance (83.1 in $\mathcal{S}_{\text{ALL}}$), improving over CamelParser2.0 by 6.1 LAS points. It also yields the strongest dialect-specific macro-averages ($\mathcal{S}_{\text{MSA}}$: 88.7, $\mathcal{S}_{\text{EGY}}$: 83.7, $\mathcal{S}_{\text{GLF}}$: 83.9).

Training Data	MSA_{DEV}				EGY_{DEV}				GLF_{DEV}				AVG			
	$\mathcal{S}_{\text{ALL}}$	$\mathcal{S}_{\text{MSA}}$	$\mathcal{S}_{\text{EGY}}$	$\mathcal{S}_{\text{GLF}}$	$\mathcal{S}_{\text{ALL}}$	$\mathcal{S}_{\text{MSA}}$	$\mathcal{S}_{\text{EGY}}$	$\mathcal{S}_{\text{GLF}}$	$\mathcal{S}_{\text{ALL}}$	$\mathcal{S}_{\text{MSA}}$	$\mathcal{S}_{\text{EGY}}$	$\mathcal{S}_{\text{GLF}}$	$\mathcal{S}_{\text{ALL}}$	$\mathcal{S}_{\text{MSA}}$	$\mathcal{S}_{\text{EGY}}$	$\mathcal{S}_{\text{GLF}}$
MSA	86.9	86.8	87.4	87.1	72.1	80.7	70.0	72.8	71.5	77.8	78.6	71.2	76.8	81.8	78.7	77.0
EGY	80.0	80.0	81.6	79.7	83.3	84.1	82.4	85.3	65.6	88.9	76.4	65.1	76.3	84.3	80.1	76.7
GLF	65.7	64.7	69.8	68.8	58.0	57.4	57.2	60.1	71.2	83.3	70.7	71.1	65.0	68.5	65.9	66.7
MSA+EGY	86.8	86.7	**87.5**	87.4	**84.1**	84.8	**83.3**	85.7	67.3	**94.4**	74.3	66.9	79.4	88.6	81.7	80.0
MSA+GLF	87.0	86.8	87.4	87.4	73.5	80.9	71.7	74.1	**79.5**	88.9	**87.1**	**79.1**	80.0	85.5	82.1	80.2
EGY+GLF	80.7	80.6	82.9	80.8	83.3	84.3	82.3	85.2	76.9	88.9	77.1	76.8	80.3	84.6	80.8	80.9
MSA+EGY+GLF	86.9	86.7	87.4	87.2	83.9	**85.0**	82.9	**86.0**	78.6	**94.4**	80.7	78.4	**83.1**	**88.7**	**83.7**	**83.9**
CamelParser2.0	**87.2**	**87.1**	**87.5**	**87.6**	72.4	80.9	70.0	73.7	71.4	77.8	77.1	71.2	77.0	81.9	78.2	77.5

Table 8: **LAS** scores on the **DEV** sets for models trained on various data combinations evaluated on the MSA, EGY, and GLF. We report overall scores ($\mathcal{S}_{\text{ALL}}$) as well as dialect-specific subset scores ($\mathcal{S}_{\text{MSA}}$, $\mathcal{S}_{\text{EGY}}$, $\mathcal{S}_{\text{GLF}}$). The AVG column shows the macro-average across the three DEV sets. Bold indicates the best result within each column.

Training Data	MSA_{TEST}				EGY_{TEST}				GLF_{TEST}				AVG			
	$\mathcal{S}_{\text{ALL}}$	$\mathcal{S}_{\text{MSA}}$	$\mathcal{S}_{\text{EGY}}$	$\mathcal{S}_{\text{GLF}}$	$\mathcal{S}_{\text{ALL}}$	$\mathcal{S}_{\text{MSA}}$	$\mathcal{S}_{\text{EGY}}$	$\mathcal{S}_{\text{GLF}}$	$\mathcal{S}_{\text{ALL}}$	$\mathcal{S}_{\text{MSA}}$	$\mathcal{S}_{\text{EGY}}$	$\mathcal{S}_{\text{GLF}}$	$\mathcal{S}_{\text{ALL}}$	$\mathcal{S}_{\text{MSA}}$	$\mathcal{S}_{\text{EGY}}$	$\mathcal{S}_{\text{GLF}}$
MSA	87.3	87.0	90.6	**88.2**	73.0	81.4	70.7	73.6	73.3	66.7	72.5	73.3	77.9	78.4	77.9	78.4
EGY	79.2	79.5	86.3	77.4	83.9	85.3	83.1	84.8	68.7	75.0	65.6	68.8	77.3	79.9	78.3	77.0
GLF	65.4	64.9	71.7	66.6	58.7	58.4	58.4	59.5	73.8	66.7	77.3	73.7	66.0	63.3	69.1	66.6
MSA+EGY	87.1	86.9	91.2	87.5	**84.4**	85.7	**83.6**	**85.3**	70.1	66.7	68.3	70.2	80.5	79.8	81.0	81.0
MSA+GLF	87.2	86.9	90.6	**88.2**	74.4	81.3	72.7	74.0	**81.0**	**83.3**	**79.9**	**81.1**	80.9	**83.8**	81.1	81.1
EGY+GLF	80.0	80.0	86.5	79.3	83.8	85.0	83.1	84.8	79.4	75.0	73.0	79.7	81.1	80.0	80.9	81.3
MSA+EGY+GLF	87.2	86.9	**91.8**	87.7	84.2	**86.1**	**83.6**	84.6	80.3	66.7	78.3	80.5	**83.9**	79.9	**84.6**	**84.3**
CamelParser2.0	**87.5**	**87.3**	91.0	88.0	73.3	82.2	70.9	73.5	72.7	66.7	72.5	72.7	77.8	78.7	78.1	78.1

Table 9: **LAS** on the **TEST** sets for models trained on different data combinations. Bold indicates the best result within each column.

Examining individual DEV sets reveals three main patterns. **First**, single-variety training limits cross-variety generalization. The MSA-only model performs well on MSA_{DEV} (86.9) but drops markedly on EGY and GLF, and similar degradation is observed for EGY-only and GLF-only setups. Our MSA-only model is also slightly below CamelParser2.0 under the same setting, likely due to the pretrained encoder choice: as noted by Inoue et al. (2021), CAMeLBERT-MSA (used by CamelParser2.0) is optimal for pure MSA, whereas CAMeLBERT-MIX (used here) favors dialectal and mixed scenarios, prioritizing robustness over peak MSA performance. **Second**, combining varieties consistently improves robustness. In all cases, two-variety models outperform their single-variety counterparts on the relevant DEV sets (e.g., MSA+GLF over GLF-only on GLF_{DEV}), indicating positive cross-variety transfer rather than interference. **Third**, even small amounts of additional dialectal data can have a large impact. Adding GLF to MSA+EGY substantially boosts GLF performance (67.3→78.6 in $\mathcal{S}_{\text{ALL}}$) with minimal effect elsewhere, suggesting that limited low-resource data can effectively leverage transfer from higher-resource varieties. Although some two-variety models achieve slightly higher scores on specific subsets, the differences are marginal. Overall, the full-data model offers the best macro-average and the most balanced cross-variety performance, supporting a single unified parser.

TEST Set Results Table 9 confirms that the trends observed on DEV generalize to TEST. The full-data model (MSA+EGY+GLF) achieves the highest overall macro-average (83.9 in $\mathcal{S}_{\text{ALL}}$), substantially outperforming CamelParser2.0 (77.8 AVG). It also yields the best dialect-specific averages ($\mathcal{S}_{\text{EGY}}$: 84.6, $\mathcal{S}_{\text{GLF}}$: 84.3). We note that $\mathcal{S}_{\text{MSA}}$ on GLF_{TEST} is much lower than on DEV; however, $\mathcal{S}_{\text{MSA}}$ sentences constitute less than 1% of the GLF subsets (Table 7), making this metric highly sensitive to small fluctuations and not indicative of overall trends. As in DEV, single-variety models perform best in-domain but degrade elsewhere, whereas multi-variety training improves cross-variety robustness. The unified model achieves the best overall average and remains competitive with the dedicated MSA baseline on MSA_{TEST}, reflecting the same trade-off observed on DEV. Overall, TEST results confirm that incor-

Section	Metric	MSA_{TEST}		EGY_{TEST}		GLF_{TEST}		Avg
		BEST	CP2.0	BEST	CP2.0	BEST	CP2.0	Δ
Overall	**UAS**	89.6	**89.8**	**86.7**	78.4	**82.5**	76.8	+4.6
	LS	92.9	**93.1**	**91.6**	83.1	**85.1**	80.6	+4.3
	LAS	87.2	**87.5**	**84.2**	73.2	**80.3**	72.7	+6.1
Relation	**MOD**	94.7	**94.9**	**93.2**	85.6	**85.6**	80.7	+4.1
	—	79.3	**79.9**	**80.1**	61.5	**74.1**	72.1	+6.7
	OBJ	94.1	**94.3**	**92.8**	86.4	**90.3**	86.6	+3.3
	IDF	97.1	97.1	**95.2**	91.8	**97.1**	91.3	+3.1
	SBJ	**89.2**	89.1	**87.6**	77.7	**83.1**	77.5	+5.1
	PRD	**87.9**	87.8	**89.2**	74.4	**88.6**	70.9	+10.9
	TPC	35.6	**39.1**	**68.2**	23.8	**55.2**	38.1	+19.3
	TMZ	**69.3**	64.2	**33.3**	6.1	–	–	+16.2
Root	**F-score**	77.3	**77.4**	**80.4**	71.4	**68.1**	66.1	+3.6
	Precision	93.1	**93.2**	**88.1**	78.2	**95.5**	92.7	+4.2
	Recall	66.0	**66.1**	**73.9**	65.6	**52.9**	51.3	+3.2

Table 10: Comparison between our multi-variety **BEST** system setup (MSA+EGY+GLF) and CamelParser2.0 (**CP2.0**) on MSA, EGY, and GLF test sets. Results are organized by overall accuracy, relation-level F-scores, and ROOT identification. Cells marked "–" indicate that the relation did not occur in the dataset. Bold indicates higher scores; Δ denotes the average improvement of BEST over CP2.0.

porating all dialectal data yields the most balanced and strongest parser across varieties.

6.3. Analysis

To help us understand where the improvement in performance is happening, we compare our **BEST** system setup (*MSA+EGY+GLF*) and CamelParser2.0 across MSA, EGY, and GLF Test sets (Table 9). We present our analysis from three perspectives: (1) overall parsing accuracy in terms of Unlabeled Attachment Score (UAS) and Label Score (LS) in addition to LAS, (2) relation-level F-score, and (3) ROOT parent identification. While CamelParser2.0 remains highly competitive on MSA, BEST yields substantial gains on dialectal Arabic without sacrificing MSA performance. The detailed results are presented in Table 10.

Overall Accuracy BEST slightly trails CamelParser2.0 on MSA in UAS (89.6 vs. 89.8) but yields substantial gains on dialectal data: +8.3 on EGY and +5.7 on GLF. A similar pattern holds for LS, with near parity in MSA (92.9 vs. 93.1) and clear improvements in EGY (+8.5) and GLF (+4.5). These gains culminate in LAS, where BEST improves markedly on EGY (+11.0) and GLF (+7.6) while remaining comparable on MSA (87.2 vs. 87.5).

Relation-Level Analysis BEST consistently outperforms CamelParser2.0 on both dialectal varieties, while remaining largely on par in MSA. The largest gains occur in syntactically marked constructions. Predicate (PRD), Topic (TPC), and Tamyiz (TMZ), the three least frequent relations, show the greatest relative improvements: +10.9, +19.3, and +16.2, respectively. Core grammatical relations (SBJ, OBJ, IDF) also exhibit steady gains (3–5 points), indicating broad and systematic improvements rather than isolated effects.

Root Identification ROOT parent identification shows clear gains in EGY and modest improvements in GLF, while remaining on par in MSA. In EGY, F-score increases by +9.0, driven by improvements in both precision (+9.9) and recall (+8.3), reflecting more accurate clause boundary detection and predicate selection. In GLF, gains are smaller (+2.0 F_1), with very high precision (95.5) but considerably lower recall (52.9). This precision–recall imbalance is consistent with the previously discussed fragmentation observed in CamelTB-Gumar sentences (Section 4.4).

Overall, our BEST setup outperforms CamelParser2.0 on average across test sets, driven by substantial improvements on EGY and GLF, while remaining only marginally lower on MSA. These results demonstrate robust cross-variety improvements without materially sacrificing performance on standard Arabic. Nevertheless, there remains room for improvement across all varieties.

7. Conclusion and Future Work

This work revisits dialectal Arabic parsing, a task that has seen limited progress despite the well-documented shortcomings of MSA-trained parsers on dialectal data. We present new empirical results on Egyptian and Gulf Arabic, demonstrating that

even modest amounts of dialectal annotation yield substantial improvements in parsing accuracy. Our contributions include a newly annotated Gulf Arabic dataset, a strong multi-variety Arabic parser, and a dialect identification–based analytical framework that clarifies how training data influences performance across dialects and evaluation settings.

In the future, we aim to move toward a truly pan-Arabic parser capable of robust generalization across varieties. This will involve expanding annotation to additional dialects and genres, evaluating full pipelines from raw text, and further examining cross-variety transfer effects. We also plan to use the new treebanks to systematically assess large language models on Arabic parsing. More broadly, by releasing datasets, models, and tools, we hope to foster a more inclusive and empirically grounded ecosystem for Arabic NLP research.

Acknowledgments

We thank the anonymous reviewers for their insightful and constructive comments. We also acknowledge the support of the High Performance Computing Center at New York University Abu Dhabi.

Limitations

Our work is limited by the availability of annotated data for Arabic dialects. Although we introduce a new evaluation set for Gulf Arabic, it remains relatively small compared to MSA treebanks. Parsing performance is also bounded by the noise and variability in dialectal orthography, especially in user-generated content. Additionally, while dialect identification helps analyze parsing behavior, it may introduce analysis errors in cases of code-switching or mixed dialect input. Finally, our models have been trained and evaluated on relatively clean, segmented input, and their robustness to noisy or unsegmented text is left to future work.

Ethical Considerations

Working with dialectal Arabic raises several ethical considerations. First, dialects are regionally and socially marked; care must be taken to avoid reinforcing stigmas or privileging certain varieties over others. Our choice to include Gulf and Egyptian Arabic was driven by data availability, but we acknowledge that this leaves out many underrepresented dialects. Second, our Gulf Arabic dataset was annotated by Arabic native speakers, and we ensured they were fairly compensated for their work. We emphasize the importance of fair payment when working with community annotators on dialectal resources. Finally, as dialectal tools become more widespread, they may be used for surveillance or social media monitoring. We urge future users and developers to consider these implications and adopt responsible use policies.

8. Bibliographical References

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2021. QADI: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 1–10, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Amr Keleg, Abdel-Rahim Elmadany, Chiyu Zhang, Injy Hamed, Walid Magdy, Houda Bouamor, and Nizar Habash. 2024. NADI 2024: The fifth nuanced Arabic dialect identification shared task. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 709–728, Bangkok, Thailand. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Noor Abo Mokh, Daniel Dakota, and Sandra Kübler. 2024. Out-of-domain dependency parsing for dialects of Arabic: A case study. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 170–182, Bangkok, Thailand. Association for Computational Linguistics.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. Shami: A corpus of Levantine Arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sharefah Al-Ghamdi, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2021. A dependency treebank for classical Arabic poetry. In *Proceedings of the Sixth International Conference on Dependency Linguistics (Depling, SyntaxFest 2021)*, pages 1–9, Sofia, Bulgaria.

Sharefah Al-Ghamdi, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2023. Fine-tuning BERT-based pre-trained models for Arabic dependency parsing. *Applied Sciences*, 13(7).

Israa Alsarsour, Esraa Mohamed, Reem Suwaileh, and Tamer Elsayed. 2018. DART: A large dataset of dialectal Arabic tweets. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2022. Hierarchical aggregation of dialectal data for Arabic dialect identification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4586–4596, Marseille, France. European Language Resources Association.

Houda Bouamor, Hanan Alshikhabobakr, Behrang Mohit, and Kemal Oflazer. 2014. A Human Judgement Corpus and a Metric for Arabic MT Evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 207–213, Doha, Qatar.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR Shared Task on Arabic Fine-Grained Dialect Identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop (WANLP19)*, Florence, Italy.

David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic dialects. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 369–376, Trento, Italy. Association for Computational Linguistics.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R El-Beltagy, Wassim El-Hajj, et al. 2021. A panoramic survey of natural language processing in the Arab world. *Communications of the ACM*, 64(4):72–81.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic treebanks and associated corpora: Data divisions manual.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing.

Kais Dukes and Tim Buckwalter. 2010. A Dependency Treebank of the Quran using Traditional Arabic Grammar. In *Proceedings of the Conference on Informatics and Systems (INFOS)*, Cairo, Egypt.

Ahmed Elshabrawy, Muhammed AbuOdeh, Go Inoue, and Nizar Habash. 2023. CamelParser2.0: A state-of-the-art dependency parser for Arabic. In *Proceedings of ArabicNLP 2023*, pages 170–180, Singapore (Hybrid). Association for Computational Linguistics.

Asma Etman and Louis Beex. 2015. Language and Dialect Identification: A Survey. In *Proceedings of the Intelligent Systems Conference (IntelliSys)*, London, UK.

Rui Fan, Shu Li, Tingting He, and Yu Liu. 2025. Aspect-based sentiment analysis with syntax-opinion-sentiment reasoning chain. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3123–3137, Abu Dhabi, UAE. Association for Computational Linguistics.

Charles F Ferguson. 1959. Diglossia. *Word*, 15(2):325–340.

Ping Guo, Yubing Ren, Yue Hu, Yunpeng Li, Jiarui Zhang, Xingsheng Zhang, and Heyan Huang. 2024. Teaching large language models to translate on low-resource languages with textbook prompting. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15685–15697, Torino, Italia. ELRA and ICCL.

Nizar Habash, Muhammed AbuOdeh, Dima Taji, Reem Faraj, Jamila El Gizuli, and Omar Kallas. 2022. Camel treebank: An open multi-genre Arabic dependency treebank. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2672–2681, Marseille, France. European Language Resources Association.

Nizar Habash, Reem Faraj, and Ryan Roth. 2009. Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Nizar Habash and Ryan Roth. 2009. CATiB: The Columbia Arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec, Singapore. Association for Computational Linguistics.

Dana Halabi, Ebaa Fayyoumi, and Arafat Awajan. 2021. l3rab: A new Arabic dependency treebank based on Arabic grammatical theory. *Transactions on Asian and Low-Resource Language Information Processing*, 21(2):1–32.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Hiroshi Kanayama, Yang Zhao, Ran Iwamoto, and Takuya Ohko. 2024. Incorporating syntax and lexical knowledge to multilingual sentiment classification on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4810–4817, Bangkok, Thailand. Association for Computational Linguistics.

Yash Kankanampati, Joseph Le Roux, Nadi Tomeh, Dima Taji, and Nizar Habash. 2020. Multitask easy-first dependency parsing: Exploiting complementarities of different dependency representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2497–2508, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Salam Khalifa, Nizar Habash, Dana Abdulrahim, and Sara Hassan. 2016. A large scale corpus of Gulf Arabic. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4282–4289, Portorož, Slovenia. European Language Resources Association (ELRA).

Salam Khalifa, Nizar Habash, Fadhl Eryani, Ossama Obeid, Dana Abdulrahim, and Meera Al Kaabi. 2018. A morphologically annotated corpus of Emirati Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Xiang Li, Zhiyi Yin, Hexiang Tan, Shaoling Jing, Du Su, Yi Cheng, Huawei Shen, and Fei Sun. 2025. PRDetect: Perturbation-robust LLM-generated text detection based on syntax tree. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 8290–8301, Albuquerque, New Mexico. Association for Computational Linguistics.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, and Dalila Tabessi. 2006. Developing and Using a Pilot Dialectal Arabic Treebank. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Genoa, Italy.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Mohamed Maamouri, Ann Bies, Seth Kulick, Michael Ciul, Nizar Habash, and Ramy Eskander. 2014. Developing an Egyptian Arabic Treebank: Impact of dialectal morphology on annotation and tool development. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Yuval Marton, Nizar Habash, and Owen Rambow. 2013. Dependency parsing of Modern Standard Arabic with lexical and inflectional features. *Computational Linguistics*, 39(1):161–194.

Benjamin Muller, Benoit Sagot, and Djamé Seddah. 2020. Can multilingual language models transfer to an unseen dialect? a case study on North African Arabizi.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 2216–2219, Genoa, Italy.

Ossama Obeid, Go Inoue, and Nizar Habash. 2022. Camelira: An Arabic multi-dialect morphological disambiguator. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 319–326, Abu Dhabi, UAE. Association for Computational Linguistics.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Ahmed Salama, Houda Bouamor, Behrang Mohit, and Kemal Oflazer. 2014. YouDACC: the Youtube Dialectal Arabic Comment Corpus. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 1246–1251, Reykjavik, Iceland.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained Arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Wael Salloum, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash, and Mona Diab. 2014. Sentence level dialect identification for machine translation system selection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 772–778, Baltimore, Maryland. Association for Computational Linguistics.

Djamé Seddah, Farah Essaidi, Amal Fethi, Matthieu Futeral, Benjamin Muller, Pedro Javier Ortiz Suárez, Benoît Sagot, and Abhishek Srivastava. 2020. Building a user-generated content North-African Arabizi treebank: Tackling hell. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1139–1150, Online. Association for Computational Linguistics.

Anas Shahrour, Salam Khalifa, Dima Taji, and Nizar Habash. 2016. CamelParser: A system for Arabic syntactic analysis and morphological disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 228–232, Osaka, Japan. The COLING 2016 Organizing Committee.

Otakar Smrž, Jan Šnaidauf, and Petr Zemánek. 2002. Prague Dependency Treebank for Arabic: Multi-Level Annotation of Arabic Corpus. In *Proceedings of the International Symposium on Processing of Arabic*, pages 147–155, Manouba, Tunisia.

Dima Taji, Nizar Habash, and Daniel Zeman. 2017. Universal dependencies for Arabic. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, Valencia, Spain.

Wajdi Zaghouani and Anis Charfi. 2018. ArapTweet: A Large Multi-Dialect Twitter Corpus for Gender, Age and Language Variety Identification. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Omar F Zaidan and Chris Callison-Burch. 2011. The Arabic Online Commentary Dataset: an Annotated Dataset of Informal Arabic With High Dialectal Content. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 37–41.

Yu Zhang, Zhenghua Li, and Min Zhang. 2020. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305, Online. Association for Computational Linguistics.