Confronting the Interaction Problem in Arabic NLP: A Resolve Layer Framework

Anonymous ACL submission

Abstract

Among major languages, Arabic is unique in its dialectal variation, and each variety brings its own morphological complexity and orthographic ambiguity. Arabic NLP has advanced on these dimensions with broader dialect resources, larger corpora, and joint models, but these efforts treat each dimension as distinct, and most real Arabic input does not arrive that way. Real-world text moves freely across regional and subregional dialects, lexical choices, constructions, and spelling improvisations, with register often shifting within the same string. These properties arrive entangled, conditioning one another, and the standard pipeline (classify dialect, normalize spelling, restore diacritics, process) produces persistent failure modes that more data cannot substantively address. The gap is structural. This paper names it the *interaction problem* and proposes the Resolve Layer: a resolving step that identifies the input regime, applies a regime configuration that updates dynamically as the regime shifts, and delivers resolved interpretation downstream, with an output contract providing regime metadata and access to alternatives. The framework is modeled on how human listeners adapt to unfamiliar speech on first exposure and recalibrate as they continue listening, and it includes four experiments to validate its core claims.

1 Introduction

1.1 The puzzle

Arabic NLP has no shortage of sophisticated tools and resources, from morphological analyzers and dialect classifiers to pretrained transformers tuned to Arabic's properties (Antoun et al., 2020; Inoue et al., 2021). Yet systems still stumble on real-world text that human listeners find trivial. In the NADI 2022 shared task, a strong MarBERT baseline reached only 31.39% macro F1 on one test set and 16.94% on another for country-level dialect identification (Abdul-Mageed et al., 2022), and Keleg and Magdy (2023) found a fine-tuned MarBERT reached only 50.74% accuracy on QADI, with error analysis tracing many failures to ambiguous or multi-dialectal text that humans handle without difficulty. Recent surveys confirm the pattern across varieties and model types (Dahou et al., 2025; Mousi et al., 2024).

These are not small gaps to be closed through incremental tuning. Their consistency suggests that the difficulty lies in process formulation, not model capacity or resource scarcity: systems fail where the linguistic signal is underspecified or mixed in ways human listeners routinely accommodate, and that points to something missing in how the processing problem itself is formulated.

1.2 The interaction problem

Arabic presents morphological density, orthographic ambiguity, dialectal variation, register shifts, and code-switching, each well documented on its own. Recent surveys recognize that they co-occur: Darwish et al. (2021) list them as a combined modeling challenge, Bourahouat et al. (2023) note that challenges "arise at various stages" of the pipeline and jointly degrade performance, and Hamed et al. (2025) show that in code-switched Arabic, dialectal variation, orthographic creativity, and speaker-level choices are entangled in ways that defeat both annotators and models. The field knows these factors compound. What has not been done is to name the interaction itself as the central problem, which may be why no modeling method addresses it directly.

This paper calls it the *interaction problem*. The difficulty is not simply that Arabic has many properties that are hard to model, but that these properties interact in ways that confound systems treating each as a separable feature. A spelling that looks like an error in one regime may be standard in another; a phonetic feature that signals south-

ern Egyptian also signals socioeconomic position and educational background. The dimensions are entangled.

Current systems address each dimension with a separate tool: classifiers handle dialect, normalizers handle orthographic variation, diacritizers handle vocalization. None addresses the interaction. A dialect classifier labels text as Egyptian but does not reconfigure downstream morphological analysis to expect Egyptian patterns; a normalizer maps variant spellings to canonical forms but discards the sociolinguistic signal those variants carried.

The scale of this problem is unique to Arabic. Four hundred million speakers across dozens of countries produce text daily in varieties that were never standardized in writing, improvised into orthography without consensus norms (Habash et al., 2012a, 2018), so the space of variation grows faster than any corpus can capture. These are not merely co-occurring noise. They form a structured signal about which interpretive regime governs the text, and when systems miss it they fail in ways additional data alone cannot fix.

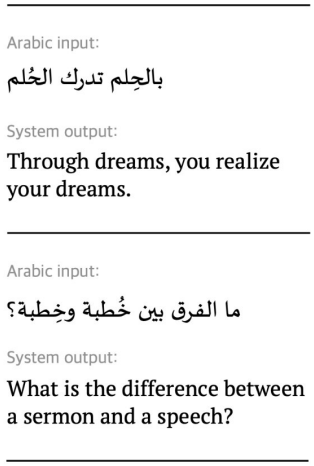

Figure 1: Regime misrecognition in a widely used translation service. In both examples, correct interpretation requires recognizing the active interpretive regime. In the first, an aphoristic pairing of *hilm* (forbearance) and *hulm* (dream) signals a Classical Arabic register whose semantic contrast is collapsed when treated as dialectal. In the second, a pedagogical question contrasting *khutbah* (sermon) and *khitbah* (betrothal proposal) presupposes a formal register that the system fails to identify.

1.3 The limits of statistical scaling

The standard response to these failures is more data and bigger models, but the variation space is not static. New expressions, spellings, and code-mixing patterns emerge daily across geographies and media, so a corpus collected this year will not capture next year's language. Statistical chasing assumes that with enough data, coverage converges toward completeness; for Arabic, the target moves faster than the chase.

Beyond coverage, statistical models trained on large corpora learn to exploit distributional regularities, but the conditioning they capture stays implicit and fixed, buried in learned weights. If a document shifts register mid-paragraph, or a speaker code-switches between varieties, the model has no mechanism to recognize the shift and reconfigure accordingly. Figure 1 illustrates this with two examples from a widely used translation service, where correct interpretation requires recognizing a shift in interpretive regime rather than resolving isolated lexical ambiguity. The interaction problem requires explicit, adaptive conditioning, and that is not what current architectures provide.

1.4 The adaptive resolution proposal

This paper proposes a different approach: an adaptive regime-based processing layer. If human listeners adapt to unfamiliar varieties within minutes, there must be something in how they process language that current systems lack, and the Resolve Layer is an attempt to specify what.

Regime-based processing, as used here, is handling each encountered regime with a regime configuration that can process it efficiently. The regime configuration is an adaptive set of expectations across phonology, morphology, syntax, and lexicon that constrains how ambiguous forms are interpreted. When a human listener identifies that a speaker is southern Egyptian, lower-educated, and speaking informally, they have not merely attached a label. They have configured a filter through which subsequent input will pass, and they now expect certain phonetic substitutions, morphological patterns, and lexical choices. When the input matches those expectations, processing is efficient, but when it deviates, the listener notices and recalibrates.

The Resolve Layer framework draws on this model. It specifies a two-phase architecture: intensive first-exposure processing that identifies a governing regime and sets its configuration, followed by regime-conditioned processing that applies it; when input stops matching expectations, surprisal spikes and the system re-escalates and reconfig-

ures. This paper does not implement the Resolve Layer but specifies functional requirements: what regime-based processing must accomplish, what outputs it must produce, and what it commits to about the relationship between resolution and rendering, ambiguity and uncertainty. Existing computational mechanisms suggest these requirements are realizable, but realizing them is future work.

2 Background

Arabic's morphological complexity and orthographic ambiguity are well documented (Habash, 2010). The root-and-pattern system, combined with concatenative affixation, lets a single orthographic word encode what English needs several words for, and without diacritics the same consonantal skeleton admits multiple decompositions. Disambiguation requires syntactic and semantic context that in turn requires correct morphological analysis, a circular dependency in which pipeline architectures propagate early errors with no mechanism for recovery. The severity is considerable: Zaghouani et al. (2014) found that 32% of words in MSA online comments contained spelling errors, and Habash et al. (2018) documented 27 orthographic variants for a single Egyptian Arabic phrase. The problem is not writer carelessness but the absence of consensus on how to spell words that never had standardized written forms.

2.1 The continuum of "Arabics"

Arabic is not a single language but a continuum spanning Classical Arabic, Modern Standard Arabic (MSA), and dozens of regional dialect systems (Ferguson, 1959). What is often overlooked is that no one speaks MSA natively. Despite its name, MSA suffers from a chronic poverty of standardization: no authoritative body governs usage and no reference works are universally accepted, which breeds variation and register elasticity (Badawi, 1973; Holes, 2004). It functions less as a fixed standard than as a flexible middle register, neither literally standard nor strictly modern, and poses its own challenges as a result.

The dialects vary substantially from one another and from MSA. Egyptian negation uses a circumfix *ma-...-sh* around the verb; Levantine uses *ma* as a preverbal particle without the suffix. Such systematic divergences have implications for tokenization, morphological analysis, and semantic processing. In written form, dialectal text lacks standardized orthography, and writers spell by phonetic intuition and personal habit, so the orthography is individually improvised.

Resource	MSA	Egyptian	Gulf	Lev.
Morph. DB	SAMA	CALIMA-ARZ	CALIMA-GLF	—
Disambig.	MADA-MIRA	MADA-ARZ	—	—
Pretr. LM	CAMeL-MSA	CAMeL-DA	CAMeL-DA	CAMeL-DA
Treebank	PATB 1–3	ARZ-TB 1–8	Gumar	—

Table 1: The Fragmented Arabic NLP Toolchain.

2.2 Dimensions of variation

The variation in Arabic text operates across many dimensions at once, and they interact in ways that shape interpretation (Nguyen et al., 2016). Dialect carries sub-dimensions of geography, age, socioeconomic status, and urban versus rural origin. Register spans formal written MSA through colloquial conversation, slang, liturgical language, and satire. Orthographic care ranges from carefully edited text to casual shortcuts, where what looks like a spelling mistake in one context is a deliberate informal convention in another. Speakers and writers also code-switch between varieties within a single text (Solorio and Liu, 2008), while mood, pragmatic stance, and individual mannerisms leave their own marks.

These dimensions do not present themselves one at a time; they arrive together, entangled in every word and phrase.

2.3 Parallel toolchains

The community has responded to dialectal variation by building separate systems for each variety. When an MSA morphological analyzer was evaluated on Levantine verb forms, recall fell to around 60% (Habash and Rambow, 2006); the response across the field has been to build separate, variety-specific resources, the Egyptian database CALIMA-ARZ among them (Habash et al., 2012b). The pattern has repeated across varieties and tools. Table 1 summarizes the resulting fragmentation across major tools and resources (Pasha et al., 2014; Obeid et al., 2020; Darwish and Mubarak, 2016).

The hyphens tell the story. The architecture scales by multiplication rather than adaptation: each new variety demands its own morpholog-

ical database, disambiguator, and treebank, and none of this helps when the input crosses columns mid-document. A document that begins in MSA, quotes a proverb in Classical Arabic, and shifts to Egyptian for commentary forces a user to stitch together tools from three columns, if they exist at all. The Resolve Layer proposes a different approach: instead of multiplying toolchains, it treats regime identification as the core problem, drawing on the existing tools as resources to invoke as needed rather than fixed toolchains to stitch together by hand.

3 Human Perceptual Adaptation

Native Arabic listeners face the same interacting dimensions that defeat NLP systems, yet adapt within seconds or minutes of meeting an unfamiliar speaker, and psycholinguistic research on accent adaptation models how. Clarke and Garrett (2004) found that listeners adjusted to foreign-accented English within about a minute, and Bradlow and Bent (2008) showed the adaptation generalizes: exposure to one speaker of an accent improves comprehension of others with similar accents. Kleinschmidt and Jaeger (2015) model this as Bayesian belief updating, with listeners maintaining probabilistic expectations and revising them as evidence accumulates. The key insight for Arabic NLP is that adaptation is not piecemeal: identifying one feature of the regime activates expectations across multiple levels at once.

3.1 First exposure and sustained processing

When a listener encounters unfamiliar speech or text, they invest cognitive effort to identify the governing patterns. Within the first few sentences, surface features point to a profile that is not just a label but a predictive model, raising expectations about morphological patterns, lexical items, and syntactic structures. The process resembles an eye exam: the optometrist tries combinations of lenses, adjusting the configuration until the image comes into focus. Processing then shifts from intensive identification to efficient filtering, and a word that could be parsed three ways in isolation may have only one plausible reading given the identified dialect and register. The listener applies the established configuration until something goes wrong.

3.2 Surprise-triggered recalibration

Sometimes the input stops matching expectations, creating surprisal, a measurable spike in processing difficulty that correlates with information-theoretic unexpectedness (Levy, 2008). Not all surprisal is regime-relevant: a rare proper noun may spike unexpectedness without signaling regime failure. What triggers re-escalation is surprisal in the morphological, syntactic, or phonological patterns that violate regime expectations.

Consider an example, text identified as Egyptian casual speech:

بيقرو عليا

Multiple readings remain open: "they are reading to me," "they are reading over me" (as in incantation), or "they are giving me the evil eye." Accumulating context weights interpretation toward the evil eye reading:

بيقرو عليا ويحسدو فيا اكمن عيوني عسلية

"They give me the evil eye because I have honey-colored eyes." The Egyptian casual regime holds. Then:

بيقرو عليا ويحسدو فيا اكمن عيوني عسلية مش هيك الغنية؟

The word هيك is Levantine, not Egyptian. Surprisal spikes at the morphological-constructional level: this is not a rare word in an Egyptian frame but a regime-inconsistent form. Re-escalation fires: the system recognizes that the speaker is Levantine, quoting an Egyptian song lyric. The preceding Egyptian text is reinterpreted as quoted material, and the Levantine regime configuration takes effect downstream.

When surprisal exceeds a threshold, the listener re-escalates and reconsiders the regime; the cycle repeats. Current Arabic NLP systems do not do this. A dialect classifier may label text as Egyptian, but that label does not reconfigure the morphological analyzer: it is metadata, not a regime configuration.

4 Proposing the Resolve Layer

4.1 Design principles

Four principles guide the Resolve Layer framework.

Regime configuration as first-class component. The regime configuration is not metadata attached after processing but a set of coordinated expectations that condition all processing. The system identifies the input regime early, and that identification activates the configuration that shapes how the rest of the input is handled.

Two-phase processing. Inspired by human perceptual adaptation, processing occurs in two phases: intensive initial identification, then efficient regime-conditioned analysis. The first phase invests effort to identify the governing regime and set its configuration; the second applies it efficiently.

Surprise-triggered re-escalation. When input violates expectations, the system re-escalates, gathers new evidence, and reconfigures, allowing adaptation to mid-text shifts such as code-switching or register changes.

Resolution precedes rendering. Resolution (determining what the text means) is the analytical work; diacritization (producing explicitly marked output) is a rendering function needed only for some downstream tasks.

4.2 How regime configuration works

Operationally, a detected regime triggers a regime configuration to handle it: a multi-axis structure of coordinated expectations across dialect, register, orthographic convention, and other dimensions that together control interpretation. Identifying the regime as southern Egyptian, informal, with loose orthography is not merely assigning three labels: a corresponding configuration takes effect as a filter. That filter shapes how the rest of the string is interpreted, which morphological decompositions are likely, what vocabulary is expected, and how much weight apparent spelling irregularities get versus dialectal conventions. The configuration persists until a surprisal event disturbs it, at which point it adaptively reconfigures: axes may be added, removed, or recalibrated.

Regime-conditioned interpretation operates at the phrase and construction level, not just individual tokens. The same surface string can mean entirely different things depending on regime. The expression كسر خاطري والله means "he really disappointed me" in Egyptian but "he really made me pity him" in Gulf. The construction فرحانة فيك means "she is happy for you" in Gulf, whereas in Egyptian it means "she is gloating at you." The negation ما هي فرحانة فيك means "she is not happy for you" in Gulf; in Egyptian, however, it means "the fact is she is gloating at you," where ما هي functions as an emphatic rather than a negator. These are not lexical ambiguities resolvable by dictionary lookup but constructional meanings that shift with regime.

4.3 The resolving process

The term "layer" indicates position in a processing pipeline between raw input and downstream tasks, emphasizing that regime identification is a necessary stage, not optional metadata. It does not commit to a specific neural architecture; what form regime identification takes computationally remains deliberately open.

Resolution proceeds through three stages. The system processes an initial span of 10–30 tokens with intensive effort (span length is subject to empirical testing), gathering evidence about dialect, register, and orthographic norms, and from it identifies a regime and sets the configuration. Metadata such as document source, platform, and user profile can serve as priors before any text is read. Subsequent input is processed efficiently through the active configuration, with ambiguous forms resolved against regime expectations. The system monitors surprisal, and when input stops matching, a new first-exposure phase begins on the problematic span.

Resolution means the system has accounted for the interacting factors and produced a reading. This is not normalization: a resolved output says this string means X, not that it has been converted to canonical form Y. The distinction matters because normalization discards information, whereas resolution preserves the original form while providing the interpretation.

4.4 Diacritization as rendering

Current Arabic NLP often treats diacritization as the disambiguation mechanism: the input is ambiguous, the diacritizer restores vowel marks, the output is unambiguous. This conflates two distinct functions. The analytical question is what the text means; the rendering question is how to display it with diacritics. A diacritizer produces plausible vocalization even for strings it misunderstands, so the output looks correct while encoding the wrong reading, and downstream processes inherit the error as new textual truth.

The Resolve Layer separates interpretation from diacritization: resolution determines meaning through regime identification without reshaping the input, and diacritization follows only when a presentation mode needs it. Many downstream tasks work directly with resolved interpretation and need no diacritics at all.

5 Output Contract

5.1 Output structure and confidence

The Resolve Layer produces structured output at two levels. Per unit, each resolved text unit carries its analysis (morphological decomposition, lemma, part of speech as appropriate), a confidence score between 0 and 1, and simplified metadata (e.g., "Egyptian, informal"). Per job, each processed span carries an overall confidence score reflecting stability across the span and summary metadata about the identified regime(s). The default output is intentionally lean.

Confidence scores must be calibrated, reflecting the system's actual accuracy on a given output. Modern neural systems are notoriously poorly calibrated (Guo et al., 2017), and uncalibrated confidence is nearly useless downstream. The Resolve Layer treats calibration as a first-class requirement so downstream systems know when to flag uncertainty or seek human review.

5.2 Ambiguity vs. uncertainty

The Resolve Layer distinguishes two types of indeterminacy that current systems conflate.

Orthographic uncertainty occurs when the system lacks confidence but a unique correct reading exists, because the string is too short or context insufficient. The uncertainty reflects system limitations, not genuine ambiguity in the input. Example: الدولة المحتلة

Semantic ambiguity (true ambiguity) occurs when multiple readings remain viable even with full information, as in punning or deliberate literary devices. The input genuinely supports more than one interpretation, and no amount of additional context could or should collapse it to one. Example: حلمك يا شيخ علام

Current systems represent both as probability distributions, conflating what is "genuinely ambiguous" with what is "uncertain." The Resolve Layer distinguishes them because downstream handling should differ: uncertain outputs need additional context or human review, while ambiguous outputs should pass downstream with their multiple readings intact.

5.3 Branched output

When more than one reading is equally valid and the regime cannot rule one out, the Resolve Layer outputs branches with equal confidence scores. For instance, حلمك (*hilmak*) branches into "your dream" and "go easy."

Before branching, the system checks broader context to see if one reading can be eliminated. Branching is rare by design: genuine semantic ambiguity that survives full contextual analysis is uncommon, and most apparent ambiguity resolves once the regime is identified. In high-stakes domains, recognizing input as Quranic recitation activates canonical readings where branching may be suppressed even when plausible.

6 Relation to Existing Systems

6.1 Existing tools as components

The Resolve Layer does not replace existing tools. Morphological databases such as SAMA and CALIMA-ARZ encode valuable knowledge about Arabic word structure, and pretrained models like CAMeLBERT capture distributional regularities across varieties. These become components a regime-aware system invokes selectively, based on the identified regime, rather than tools users must manually select and stitch together.

6.2 Dialect identification vs. regime identification

Dialect identification, as currently practiced, labels text. A classifier produces an output like "Egyptian" or "Gulf," but the label sits inert: it does not change how the rest of the input is processed. Regime identification activates a configuration that does. Once the system reads the regime as Egyptian, informal, southern, with loose orthography, that configuration brings in the matching morphological expectations and register patterns, a tolerance for non-standard spellings, and lexical probabilities weighted toward Egyptian vocabulary. Where dialect identification stops at a label, regime identification triggers a configuration that carries through toward an interpretation and stays ready to adapt and reconfigure again.

6.3 Joint modeling and its limits

Joint modeling has shown substantial gains by combining diacritization, lemmatization, and morphological tagging (Zalmout and Habash, 2017, 2020). These systems learn shared representations across tasks, but they condition on regime implicitly: the conditioning is opaque and fixed at training time, then applied uniformly. If a document shifts from MSA to Egyptian mid-paragraph, the

joint model cannot detect this and adjust. The Resolve Layer's explicit, adaptive conditioning addresses exactly this gap.

6.4 Domain adaptation vs. regime identification

Regime-aware processing is related to but distinct from domain adaptation (Ramponi and Plank, 2020; Ben-David et al., 2010). Domain adaptation assumes a known target domain, adapted to once through fine-tuning or auxiliary signals. Regime identification assumes the regime is unknown at inference time, may shift within a document, and must be detected and acted on dynamically. The two address different failure modes: domain adaptation closes a gap between training and deployment distributions; regime identification handles distributional structure internal to the input itself.

6.5 Parallel toolchains do not scale

As discussed in Section 2.3, the community has responded to dialectal variation with parallel toolchains that scale by multiplication: supporting N dialects requires roughly N sets of morphological databases, disambiguators, and treebanks. The Resolve Layer scales by adaptation instead. Rather than building a new toolchain for each dialect, it puts regime identification at the center, identifying the active regime and invoking existing variety-specific resources as needed. Handling a previously unsupported dialect then becomes a matter of extending the regime space so the system recognizes that dialect's patterns, not constructing a separate toolchain. This still requires linguistic evidence, but advances in deep learning and pretraining let the Resolve Layer learn from existing corpora and morphological databases, not just invoke them: the resources built for individual dialects become training signal for what each variety looks like and how regime shifts manifest.

7 Implementation and Future Work

7.1 The contribution is the analysis

This paper identifies the interaction problem, examines why current methods do not address it, and specifies what regime-based processing must accomplish. The architectural claim is that adaptive interpretation belongs immediately after tokenization, not downstream from normalization and diacritization, where interpretive decisions are made in discrete steps that do not collaborate and each locks in errors the other cannot revisit.

7.2 Existing mechanisms suggest paths

The computational toolkit for realizing the Resolve Layer may already exist, at least in part. Adaptive Computation Time (Graves, 2016) lets networks allocate variable computation to different inputs, mapping onto the two-phase model. Early-exit transformers (Xin et al., 2020; Zhou et al., 2020; Liu et al., 2020) exit at intermediate layers when confidence is high, supporting efficient regime-conditioned processing. Adapter modules (Houlsby et al., 2019) enable regime-specific processing without full retraining, and in-context learning (Brown et al., 2020) suggests a path to regime conditioning through regime-appropriate examples in context. Another mechanism routes inputs through learned experts via a gating network trained end-to-end with the task: Mixture-of-Experts architectures (Shazeer et al., 2017; Fedus et al., 2022). The Resolve Layer adapts this pattern by conditioning routing on explicitly identified regime (dialect, register, code-switching state, formality), keeping the active regime configuration inspectable to downstream consumers. These mechanisms suggest the requirements are realizable, though the question of how to assemble them remains open.

7.3 A diffusion analogy

One way to conceptualize the resolving process is by analogy to diffusion models. Arabic input arrives with uncertainty distributed across interacting dimensions (dialect, register, orthographic convention), a structured underspecification rather than corruption. Conditioned on accumulating regime evidence, the system progressively sharpens its interpretation, reweighting candidate analyses under the active regime configuration rather than committing early, and re-expanding uncertainty when surprisal signals that the regime no longer fits. This restates the optometrist analogy of Section 3 in computational terms: with noise schedules modeling orthographic and dialectal variation and denoising conditioned on regime identification, diffusion principles could offer a principled path from underspecified input to resolved interpretation.

7.4 Open questions

Several empirical questions must be answered to guide realization. Can regimes be reliably identified from 10–30 tokens, and which input features are most diagnostic? How should identification reconfigure processing, whether through routing, adapter activation, or attention modulation? What training signal teaches a system to identify multidimensional regimes, and can regime structure be learned from raw text with variety labels rather than supervised annotation? And how should surprisal be measured for re-escalation, distinguishing true regime shift from local anomaly?

7.5 Validation agenda

The framework's premises are empirically testable, and four experiments would provide evidence for or against its core claims. First, using existing resources such as CAMeLBERT and dialect-labeled corpora (MADAR (Bouamor et al., 2018), NADI (Abdul-Mageed et al., 2022)), test whether a regime, operationalized as a multi-axis configuration (dialect + register + formality), can be reliably identified from short spans and drive adaptive reconfiguration. Second, on a test set of mixed varieties, compare uniform processing, dialect identification followed by dialect-specific models, and an oracle with perfect regime identification, measuring morphological tagging accuracy to establish whether regime conditioning adds value. Third, annotate ambiguous cases as semantic ambiguity versus orthographic uncertainty and measure whether systems can distinguish them and whether downstream tasks benefit. Fourth, on texts with embedded code-switches, measure whether surprisal-based re-escalation correctly identifies switch points and improves accuracy on post-switch spans. Positive results would validate the framework; negative ones would show which components need revision.

8 Implications

The analysis developed in this paper has implications beyond any particular implementation strategy. If the interaction problem as framed is valid, several consequences follow.

First, current evaluation practices may be measuring the wrong object. Benchmarks typically test text where dialect and register stay consistent, rewarding systems that do well when nothing changes but concealing the failures that occur when conventions shift mid-document. A system can score high on isolated tasks yet prove a gullible listener in real situations that demand regime identification and reconfiguration. Judging performance must therefore include robustness under regime shifts and sensitivity to midstream surprise points.

Second, the interaction problem should prompt a reexamination of the assumption that linguistic variation in Arabic can be addressed with more data collection. Arabic variation is not only extensive but adaptive and generative: writers improvise orthography and register spontaneously across contexts. Handling it as a statistical issue conflates underspecification with ambiguity, confusing orthographic uncertainty with a flat multiplicity of form. A regime-based approach reframes the challenge as dynamic interpretation rather than data coverage.

Third, separating linguistic resolution from diacritic rendering has downstream benefits. When diacritization is the mechanism of disambiguation, the diacritizer's interpretive choices are locked in and propagate as truth about the original's communicative intent. Treating it as a separate rendering function lets uncertainty be handled more cautiously.

The regime-based framing remains open and may require revision or replacement as empirical work tests its limits. Less tenable is the assumption that Arabic NLP can scale only by multiplying tools and data. Describing the interaction problem does not solve it, but it centers what must be accounted for.

9 Conclusion

This paper has argued that persistent difficulties in Arabic NLP stem not from the individual properties of Arabic text but from their interaction. The Resolve Layer framework makes regime identification, modeled on the adaptation human listeners perform routinely, a first-class component that conditions subsequent processing and adapts when input violates expectations. It specifies what such processing must accomplish and what its outputs must look like, inviting a research program around how to realize them. The interaction problem will not yield to more data or tools built along current lines; it requires a different formulation of the processing task.

Limitations

What this paper offers is an analysis of an under-theorized phenomenon and a proposal of how it can be confronted. It names the interaction problem and specifies what a regime-aware approach to processing must accomplish. The Resolve Layer is a proposed framework, not accompanied by implementation results or benchmark numbers. While further design elaboration is needed, the work that lies directly ahead is engineering work. The routing mechanisms sketched in Section 7.2 need to be assembled into a pipeline. Training a regime identifier is a key priority, combined with an investigation to determine the minimum text length needed for regime identification. And then the architectural claim, that adaptive interpretation belongs immediately after tokenization, has to face the test of running code.

The validation agenda is an initial prospective one. Section 7.5 indicates four experiments designed to provide evidence for or against the framework's core claims. They have not been run. They are specific enough to be picked up as written.

The regime configuration in this paper requires more design detailing. Once a regime is identified, it gets handled by a multi-axis regime configuration spanning dialect, register, orthographic convention, and other dimensions. Future work will explore how those axes are weighted against one another, how they combine when they pull in opposite directions, how the system decides which axis carries the most diagnostic load in a given input.

The argument is developed entirely through Arabic where the interaction problem is most acute and most clearly visible. The claim that the framework generalizes to any language with co-occurring dimensions of variation is, however, not yet demonstrated. What would close this gap is comparative work. Languages with different variation profiles, Mandarin with its literary-colloquial split, Norwegian with its written-spoken asymmetry, South Asian languages with their script-register entanglements, would either confirm the structural similarity or expose the scope of the framework as narrower than claimed. Either outcome would be productive.

References

Abdul-Mageed, Muhammad, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97. ACL.

Antoun, Wissam, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of OSACT 2020*, pages 9–15. LREC.

Badawi, Said. 1973. *Mustawayāt al-ᶜArabiyya al-muᶜāṣira fī Miṣr* [Levels of Contemporary Arabic in Egypt]. Dar al-Ma'arif, Cairo.

Ben-David, Shai, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79(1–2):151–175.

Bouamor, Houda, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic dialect corpus and lexicon. In *Proceedings of LREC 2018*.

Bourahouat, Ghizlane, Manar Abourezq, and Najima Daoudi. 2023. Systematic review of the Arabic natural language processing. *Journal of Theoretical and Applied Information Technology*, 101(4):1547–1562.

Bradlow, Ann R. and Tessa Bent. 2008. Perceptual adaptation to non-native speech. *Cognition*, 106(2):707–729.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS 2020*, pages 1877–1901.

Clarke, Constance M. and Merrill F. Garrett. 2004. Rapid adaptation to foreign-accented English.

Journal of the Acoustical Society of America, 116(6):3647–3658.

Dahou, Abdelghani, Abdelhalim Hafedh Dahou, Mohamed Amine Cheragui, Amin Abdedaiem, Mohammed Abdulaziz Aide Al-qaness, Mohamed Abd Elaziz, Ahmed A. Ewees, and Zhonglong Zheng. 2025. A survey on dialect Arabic processing and analysis: Recent advances and future trends. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(8):1–45.

Darwish, Kareem, 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, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the Arab world. *Communications of the ACM*, 64(4):72–81.

Darwish, Kareem and Hamdy Mubarak. 2016. Farasa: A new fast and accurate Arabic word segmenter. In *Proceedings of LREC 2016*, pages 1070–1074.

Fedus, William, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23:1–39.

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

Graves, Alex. 2016. Adaptive computation time for recurrent neural networks. arXiv:1603.08983.

Guo, Chuan, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *ICML 2017*, pages 1321–1330. PMLR.

Habash, Nizar. 2010. *Introduction to Arabic Natural Language Processing*, volume 3 of Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

Habash, Nizar, Mona Diab, and Owen Rambow. 2012a. Conventional orthography for dialectal Arabic. In *Proceedings of LREC 2012*, pages 711–718.

Habash, Nizar, Ramy Eskander, and Abdelati Hawwari. 2012b. A morphological analyzer for Egyptian Arabic. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, pages 1–9.

Habash, Nizar, Fadhl Eryani, Salam Khalifa, Owen Rambow, Dana Abdulrahim, Alexander Erdmann, Reem Faraj, Wajdi Zaghouani, Houda Bouamor, Nasser Zalmout, Sara Hassan, Faisal Al-Shargi, Sakhar Alkhereyf, Basma Abdulkareem, Ramy Eskander, Mohammad Salameh, and Hind Saddiki. 2018. Unified guidelines and resources for Arabic dialect orthography. In *Proceedings of LREC 2018*, pages 3387–3396.

Habash, Nizar and Owen Rambow. 2006. MAGEAD: A morphological analyzer and generator for the Arabic dialects. In *Proceedings of COLING-ACL 2006*, pages 681–688.

Hamed, Injy, Caroline Sabty, Slim Abdennadher, Ngoc Thang Vu, Thamar Solorio, and Nizar Habash. 2025. A survey of code-switched Arabic NLP. In *Proceedings of COLING 2025*. Forthcoming.

Holes, Clive. 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown University Press.

Houlsby, Neil, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *ICML 2019*, pages 2790–2799.

Inoue, Go, 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 WANLP 2021*, pages 92–104. ACL.

Keleg, Amr and Walid Magdy. 2023. Arabic dialect identification under scrutiny: Limitations of current benchmarks and models. In *Proceedings of ArabicNLP 2023*. ACL.

Kleinschmidt, Dave F. and T. Florian Jaeger. 2015. Robust speech perception: Recognize the familiar, generalize to the similar, and adapt to the novel. *Psychological Review*, 122(2):148–203.

Levy, Roger. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Liu, Weijie, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: A self-distilling BERT with adaptive inference time. In *Proceedings of ACL 2020*, pages 6035–6044.

Mousi, Basel, Nadir Durrani, Fahim Ahmad, Md Arafat Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2024. ARADICE: Benchmarks for dialectal and cultural capabilities in LLMs. arXiv:2409.11404.

Nguyen, Dong, A. Seza Doğruöz, Carolyn P. Rosé, and Franciska de Jong. 2016. Computational sociolinguistics: A survey. *Computational Linguistics*, 42(3):537–593.

Obeid, Ossama, 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 LREC 2020*, pages 7022–7032.

Pasha, Arfath, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of LREC 2014*, pages 1094–1101.

Ramponi, Alan and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of COLING 2020*, pages 6838–6855.

Shazeer, Noam, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR 2017*.

Solorio, Thamar and Yang Liu. 2008. Learning to predict code-switching points. In *Proceedings of EMNLP 2008*, pages 973–981.

Xin, Ji, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of ACL 2020*, pages 2246–2251.

Zaghouani, Wajdi, Nizar Habash, Houda Bouamor, Alla Rozovskaya, Behrang Mohit, Abeer Hebert, and Kemal Oflazer. 2014. Large scale Arabic error annotation: Guidelines and framework. In *Proceedings of LREC 2014*, pages 2362–2369.

Zalmout, Nasser and Nizar Habash. 2017. Don't throw those morphological analyzers away just yet: Neural morphological disambiguation for Arabic. In *Proceedings of EMNLP 2017*, pages 704–713.

Zalmout, Nasser and Nizar Habash. 2020. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. In *Proceedings of ACL 2020*, pages 8297–8307.

Zhou, Wangchunshu, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. 2020. BERT loses patience: Fast and robust inference with early exit. In *NeurIPS 2020*, pages 18330–18341.