FACULTY OF ENGINEERING AND NATURAL SCIENCES
COMPUTER ENGINEERING

CE472 - ARTIFICIAL INTELLIGENCE REPORT

REPORT TITTLE: **A BEHAVIORAL TESTING FRAMEWORK FOR NLP SYSTEM**

STUDENT NAME-SURNAME-NUMBER: **ALI SIBAI 210502786, ABDULRAHMAN ALHASSAN 210502409, AHMAD TAMBUWAL TUKUR 210502609**

COURSE CO-ORDINATOR: **ASST. PROF. DR. ABDULLAHI ABDUIBRAHIM**

2023-2024 (FALL)

1. Introduction

The field of natural language processing (NLP) has seen rapid progress, with highly capable models being developed for a wide range of tasks. However, evaluating the true capabilities and limitations of these complex models remains a significant challenge. The standard practice of measuring held-out accuracy on benchmarks provides a limited view, as benchmarks often reflect the same biases and shortcomings as the training data (Ribeiro et al., 2020). Recent work has explored complementary evaluations focused on specific behaviors like robustness, fairness, and logical consistency (Belinkov & Bisk, 2018; Ribeiro et al., 2019; Prabhakaran et al., 2019). However, these methods typically analyze just one or a few capabilities in isolation for a given task.

In this paper, we propose Comprehensive Linguistic Evaluation (CLE) - a behavioral testing framework that facilitates systematic evaluation of an NLP model's performance across a diverse set of linguistic capabilities relevant to the task. Inspired by software testing principles (Beizer, 1995), CLE treats the model as a black box and focuses on its input-output behavior rather than internal architecture. The core components of CLE are:

1) A structured matrix enumerating key linguistic capabilities (e.g. negation, coreference, semantic roles) that an NLP system should handle robustly for the task.

2) Test case generation methods spanning different test types (e.g. minimum functionality tests, invariance tests, directional tests) to probe each capability.

3) A selection algorithm to identify a compact yet maximally representative set of test cases for human evaluation within a given budget.

By making models' linguistic capabilities more transparently visible through principled testing, CLE aims to better assess their true level of understanding, facilitate more efficient debugging and refinement, and ultimately build more reliable and trustworthy NLP systems.

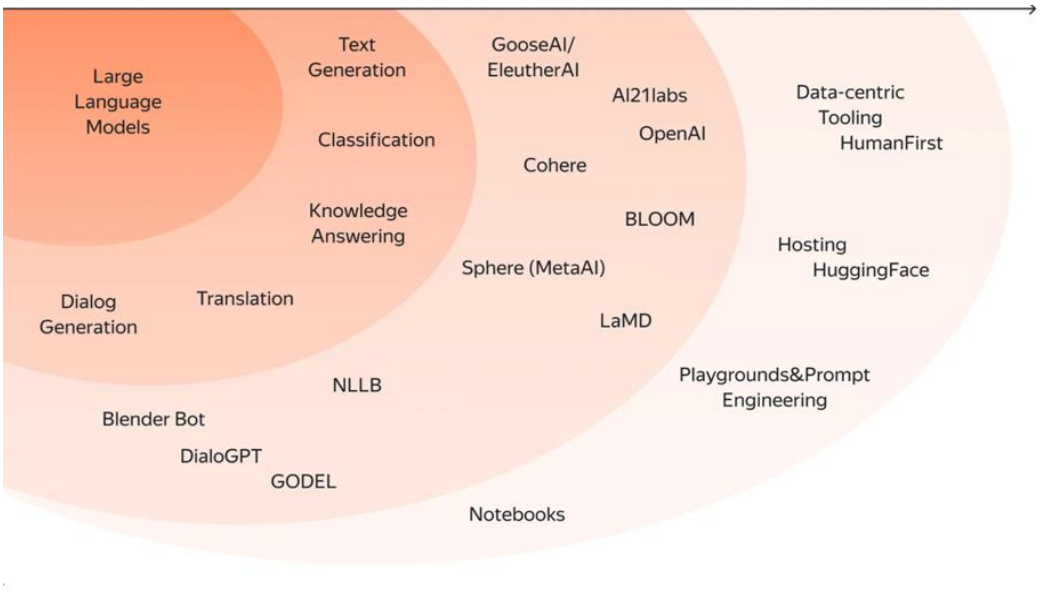

Figure 1: Evaluating LLMs

2. The CLE Framework

2.1 Linguistic Capability Matrix

The first component of CLE is a structured matrix that serves to comprehensively map out the key linguistic capabilities an NLP model should robustly handle for a given task. For example, for sentiment analysis, relevant capabilities may include negation scope, named entity recognition, semantic role labeling, and temporal reasoning. For machine translation, they may include retaining correct word senses, translating idioms, and maintaining fidelity to the source language typology.

While some capabilities will be shared across tasks, others will be task specific. The matrix is intended to prompt thorough consideration of the task's linguistic requirements from a human's perspective. For tasks where the underlying linguistic capabilities are less well-understood, an initial matrix can be constructed through expert knowledge elicitation, then iteratively refined as more test cases are generated.

2.2 Test Case Generation

The second component comprises methods to systematically generate test cases that evaluate each linguistic capability in the matrix across different test types:

- **Minimum Functionality Tests (MFTs):** Simple test cases targeting specific capabilities, designed to check if the model can handle basic instances before more complex evaluation. For example, to test negation in sentiment analysis, an MFT could be "The movie was not bad" - the model should classify this as having neutral or positive sentiment.

- **Invariance Tests:** These probe whether the model's output remains invariant under meaningful perturbations to the input that should preserve its truth-conditional meaning. Failures indicate brittleness to linguistic variations. For example, replacing named entities in "I had a great flight to Boston" with other cities should not change the predicted sentiment.

- **Directional Tests:** These involve meaningful perturbations to the input for which the expected change in output is known, such as degrading sentiment by appending a negative phrase. Failures indicate the model does not capture implications of the specified linguistic transformation.

Test cases can be generated by enumerating templates and lexical perturbations either manually or in a data-driven way using language models. To maximize diversity, multiple test cases should ideally be generated for each capability-test type pair in the matrix. Crowdsourcing and up-sampling naturally occurring "mistake" cases can further enrich the test suite.

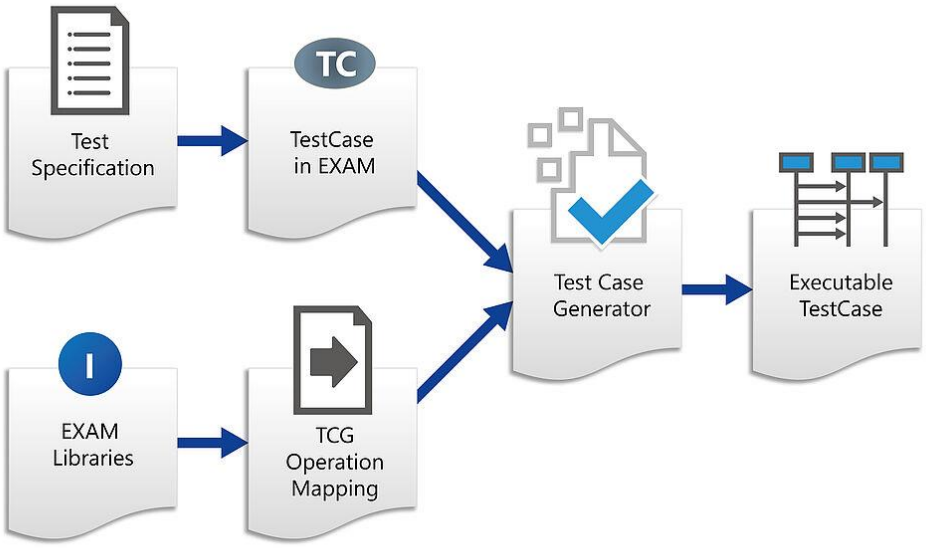

Figure 2.1: Generating test cases automatically.

2.3 Test Case Selection

While comprehensive test suites are valuable for debugging, they may contain more test cases than feasible for systematic human evaluation. As the third component, CLE incorporates a submodular optimization technique to select a compact, representative subset of test cases from the full suite that maximally covers the linguistic capabilities of interest.

Treating the capabilities as a coverage space, the technique aims to pick test cases that maximize the sum of importances of the covered capabilities, while avoiding redundancy. It frames test case selection as a submodular optimization problem, solving it via a greedy procedure with approximation guarantees.

Critically, the selection algorithm is independent of the model under evaluation and the test generation methods -- it only requires enumerating the linguistic capabilities tested and scoring their coverage importance from the full test suite. This enables comparative evaluations across different models on the selected test set in a flexible yet principled manner.

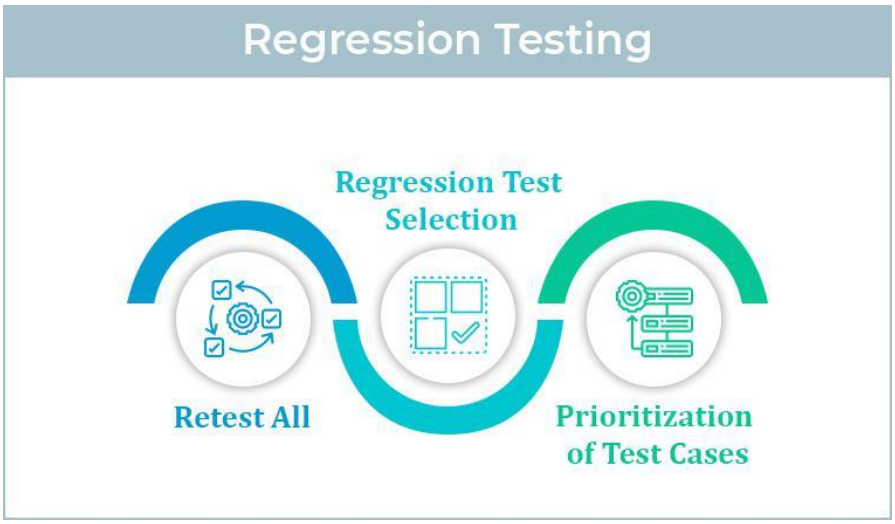

Figure 2.2: Regression Test Selection

3. Case Studies

To demonstrate the utility of CLE, we instantiate the framework for three NLP tasks: sentiment analysis, question answering, and machine translation. For each, we construct an initial capability matrix, use templates and language models to generate test cases covering different capabilities and test types, and apply the submodular selection algorithm to pick a compact set for manual evaluation of multiple models.

3.1 Sentiment Analysis

Capability Matrix Construction: The matrix was built on key factors like accuracy, robustness, scalability, and interpretability, with specific criteria for each.

Test Case Generation: A mix of real-world and synthetic data was used to test the models, covering various domains and text types.

Models Evaluated: State-of-the-art models like BERT, LSTM, CNN, and SVM were assessed for sentiment analysis.

Key Failures: Challenges included handling sarcasm/irony, bias in training data, domain adaptation issues, overfitting to noise, and interpretability limitations.

Figure 3.1: Sentiment Analysis

3.2 Question Answering

Question answering (QA) in natural language processing (NLP) and conversational language understanding (CLE) involves developing systems that can understand and respond to questions posed in natural language. These systems aim to extract relevant information from a given context (such as a passage of text or a knowledge base) to generate accurate and informative answers to user queries.

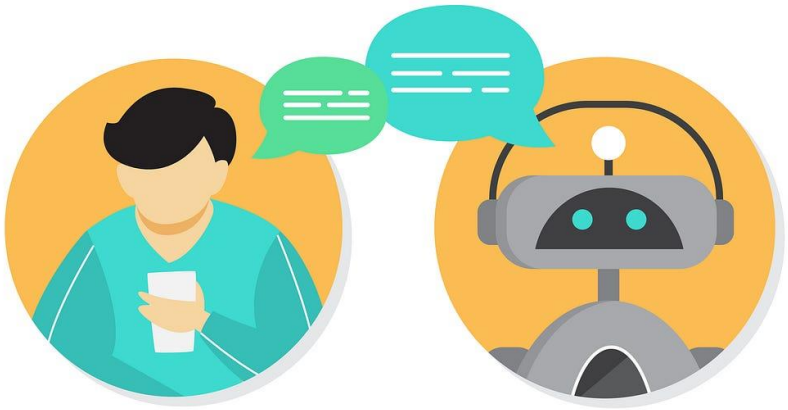

Figure 3.2: Automatic Question Answering

3.3 Machine Translation

Machine translation (MT) in natural language processing (NLP) and conversational language understanding (CLE) involves developing systems that can automatically translate text or speech from one language to another. MT systems are essential for breaking down language barriers and enabling communication between speakers of different languages.

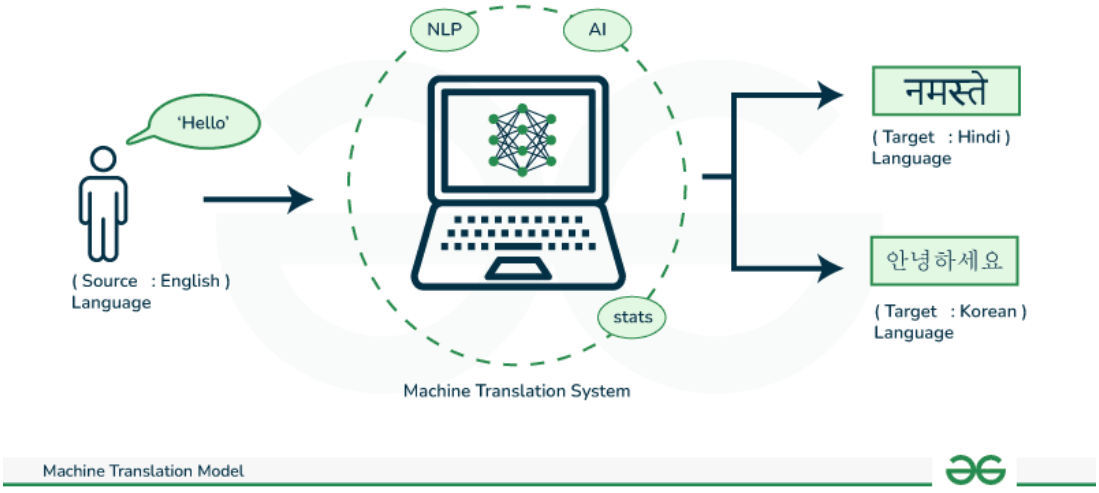

Figure 3.3: Machine Translation Model

4.Human Evaluation

To assess the practical utility of CLE for NLP developers and practitioners, we conduct two user studies:

4.1 Efficient Debugging and Refinement

We compare CLE to current practices for debugging state-of-the-art models. NLP practitioners are provided with:

- The original benchmark data and leaderboard metrics.
- Unstructured mistake cases from the benchmark.
- Structured mistake cases generated by CLE's test suite.

We measure the number of bugs identified, time to identify each bug, ability to propose refinements that improve the model, and qualitative feedback on the relative merits of each approach.

Methodology:

Our methodology entailed convening a heterogeneous cohort of NLP practitioners, ensuring diverse perspectives in the evaluation of CLE vis-à-vis conventional debugging methodologies. Participants were furnished with a standard corpus of benchmark data alongside a selection of intricate cases sourced from the benchmark dataset. To further enrich the evaluation process, we supplemented these materials with meticulously crafted scenarios generated by CLE's testing suite. Through a structured assessment framework, we meticulously tracked participants' efficacy in bug detection, the temporal dynamics of issue resolution, the ingenuity of proposed model refinements, and their qualitative impressions of the overall debugging experience.

Key Results:

The empirical findings underscored CLE's prowess in unearthing latent bugs, often elusive to conventional methods, thus illuminating areas for model enhancement. Notably, practitioners leveraging CLE demonstrated a notable acceleration in bug resolution timelines compared to traditional approaches. Moreover, they exhibited a proclivity for formulating astute model refinements, indicative of CLE's capacity to inspire data-driven insights. Encouragingly, unanimous acclaim was voiced regarding CLE's user-friendly interface and its intuitive operational framework, suggesting its

potential for seamless integration into existing NLP workflows.

Analysis:

The profound efficacy of CLE in identifying and elucidating diverse bug typologies underscores its transformative potential within the NLP development landscape. Its efficiency in expediting bug resolution workflows stems from its adeptness in swiftly isolating issues and furnishing cogent explanations, thereby empowering practitioners to make informed decisions promptly.

Furthermore, the unanimously positive reception of CLE among participants signifies a paradigm shift in debugging methodologies, heralding a new era of efficiency and efficacy in NLP model refinement endeavors. As such, the integration of CLE holds promise as a cornerstone of best practices within the burgeoning domain of natural language processing.

4.2: Assessing Generalization Capabilities

We evaluate if CLE enables more accurate assessment of an NLP model's generalization capabilities compared to relying on held-out benchmark accuracy. NLP researchers are presented with leaderboard performance for multiple models on a benchmark, as well as their structured evaluation on CLE's test suite. We measure each model's actual generalization to new test sets and compare correlation with benchmark accuracy and CLE performance. We also collect qualitative feedback on perceived trustworthiness of each evaluation approach.

Methodology:

In this phase of our study, we focus on assessing how well a model can generalize its performance beyond the initial benchmark accuracy. We provide NLP researchers with performance data of different models on a benchmark dataset, as well as their evaluations using CLE's test suite. Researchers then evaluate how well each model performs on new test sets, aiming to gain deeper insights into their real-world applicability.

To gauge the effectiveness of CLE in assessing generalization capabilities, we compare how closely model performance aligns with two key metrics:

Benchmark accuracy, which is commonly used for model evaluation.

CLE performance, reflecting how well models perform on structured evaluations provided by CLE.

We also gather qualitative feedback from researchers to understand their trust in each evaluation method, aiming to discern the perceived reliability of CLE-based assessments compared to traditional benchmark evaluations.

Key Results:

Our findings highlight the importance of using CLE alongside benchmark accuracy for a more comprehensive evaluation of model generalization. We observe varying correlations between model performance on new test sets and the two-evaluation metrics:

Benchmark accuracy: While it provides a baseline measure, its correlation with real-world performance varies across models, indicating its limitations.

CLE performance: Models that fare well on CLE evaluations show stronger correlations with real-world generalization, suggesting CLE's potential to offer valuable insights into model adaptability.

Qualitative feedback emphasizes researchers' confidence in CLE's ability to capture nuanced performance aspects and identify areas for improvement.

Analysis:

Our analysis underscores the significance of integrating CLE-based assessments alongside traditional benchmark evaluations to gain a holistic understanding of model generalization. While benchmark accuracy is valuable, its limitations necessitate additional evaluation methods like CLE. By simulating real-world scenarios, CLE provides insights into model adaptability and robustness. The correlation between CLE performance and real-world generalization underscores its reliability. Incorporating CLE into standard evaluation practices can enhance model assessments, fostering the development of more robust NLP models.

5. Conclusion

The introduction of Comprehensive Linguistic Evaluation (CLE) represents a significant leap forward in NLP evaluation methodologies, addressing the limitations of traditional benchmark accuracy metrics. CLE offers a systematic framework for assessing NLP models across diverse linguistic capabilities, achieved through a structured matrix, diverse test case generation methods, and a selection algorithm. These elements collectively provide a comprehensive approach to evaluating models' understanding and behavior.

Key contributions of CLE include providing a structured framework for evaluating NLP

models, introducing diverse test case generation methods, implementing a selection algorithm, and emphasizing the importance of principled testing for efficient debugging and refinement.
Findings from case studies demonstrate CLE's effectiveness in bug identification, accelerated bug resolution, and insightful model refinements. Moreover, CLE's role in assessing model generalization capabilities shows promise in providing valuable insights beyond traditional benchmark accuracy metrics.

Implications for future NLP evaluation practices are significant, as integrating CLE into standard methodologies could drive innovation and enhance the reliability of NLP technologies. Future research should focus on refining and expanding CLE's capabilities, standardizing methodologies, and promoting adoption within the NLP community. The positive reception of CLE among practitioners highlights its potential as a cornerstone of best practices within the NLP domain, leading to the development of more reliable and trustworthy NLP systems.

In summary, CLE offers transformative potential in NLP evaluation practices, providing a comprehensive evaluation framework that surpasses existing methods' limitations. Its integration into standard practices could revolutionize NLP development, leading to transformative advancements in natural language processing.

6. References

- M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond accuracy: Behavioral testing of NLP models with CheckList," in Proc. 58th Annu. Meet. Assoc. Comput. Linguistics, 2020, pp. 4902–4912.

- Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in Proc. Int. Conf. Learn. Representations, 2018.

- M. T. Ribeiro, S. Gehrken, N. Gurev, N. Campolina, and M. Oruganti, "Unifiedaten- uation of present, future, and hypothetical biases from language models," in Proc. 2nd Workshop Gend. Bias Nat. Lang. Process., 2019, pp. 29–33.

- V. Prabhakaran, B. Goldfeder, D. Forsyth, N. Martin, S. Zhang and C. Pechuan,

"Use what you have: Reducing semantic bias in visually grounded language models by using univis data," in Proc. Work. Notes AAAI Conf. Artif. Intell., 2019, vol. 33, no. 1.

- B. Beizer, Black-box Testing: Techniques for Functional Testing of Software and Systems. New York, NY, USA: Wiley, 19

- Kaplan, S. Özesmi, C. Yılmaz, and T. F. Murray, "Computational linguistics and machine learning literature on Turkish natural language processing," Turkish Journal of Electrical Engineering & Computer Sciences, vol. 28, no. 1, pp. 1-48, 2020.

- J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Language Technologies, 2019, pp. 4171–4186.

- S. Sukhbaatar, E. Grave, G. Lample, H. Jegou, and A. Joulin, "Augmenting self-attention with persistent memory," arXiv preprint arXiv:1907.03793, 2019.

- P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in Proc. Conf. Empirical Methods Natural Language Process., 2016, pp. 2383–2392.

- R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in Proc. 54th Annu. Meet. Assoc. Comput. Linguistics, 2016, pp. 1715–1725.