DSGA 1011: Assignment 4

Full Name
Net ID

Q0. 1.

Please provide a link to your github repository, which contains the code for both Part I and Part II. TODO

Q2. 1.

Describe your transformation of dataset.

Our transformation strategy creates out-of-distribution (OOD) evaluation data by applying multiple realistic text modifications that preserve semantic meaning and sentiment labels. The transformation uses a multi-stage approach combining synonym replacement, verb tense changes, contraction expansion/contraction, and controlled typographical errors.

Transformation Components

1. Synonym Replacement (Two-Tier Approach): We employ a two-tier synonym replacement strategy with part-of-speech (POS) based probabilities:

- **Domain-Specific Groups (Primary):** We maintain curated synonym groups for movie review terminology, including:
  - Movie/film terms: {`movie, film, picture, motion picture`}
  - Acting/performance: {`acting, performance, portrayal, playing`}
  - Story/narrative: {`story, narrative, tale, plot, screenplay, script`}
  - Quality adjectives: {`good, great, excellent, fine, fantastic, outstanding`} and {`bad, terrible, awful, poor, horrible`}
  - Common verbs/adverbs: {`watch, see`}, {`very, extremely, really, quite`}

  These groups are stored as sets and automatically build bidirectional lookup dictionaries, ensuring no redundancy and easy maintenance.
- **WordNet Fallback:** For words not in domain groups, we query WordNet synsets using NLTK's `wordnet.synsets()` with appropriate POS tags. We filter synonyms to ensure they are alphabetic, differ from the original word, and are not duplicates.
- **POS-Based Probabilities:** Replacement probabilities vary by word type to balance semantic preservation with transformation aggressiveness:
  - Adjectives: 80% (`p_synonym_adj = 0.80`)
  - Adverbs: 80% (`p_synonym_adv = 0.80`)
  - Nouns: 65% (`p_synonym_noun = 0.65`)
  - Verbs: 80% (`p_synonym_verb = 0.80`)

  Higher probabilities for adjectives/adverbs allow semantic variation while maintaining sentiment, while verbs and nouns are transformed more conservatively to preserve core meaning.

2. Verb Tense Changes: With probability 75% (`p_tense_change = 0.75`), we change verb tenses using a comprehensive mapping dictionary covering common verb forms (e.g., present ↔ past: `is` ↔ `was`, `goes` ↔ `went`, `watches` ↔ `watched`). This introduces temporal variation while preserving semantic content.

3. Contraction Expansion/Contraction: With probability 20% (`p_contraction = 0.20`), we randomly expand contractions (e.g., `don't` → `do not`) or contract expanded forms (e.g., `I am` → `I'm`). This simulates natural language variation in formality and style.

4. Controlled Typographical Errors: We introduce realistic typing errors using QWERTY keyboard proximity simulation:

- Probability: 40% per eligible word (`p_typo = 0.40`)
- Maximum: 10 typos per example (`max_typos_per_example = 10`)
- Method: For words longer than 3 characters, we randomly select a character position (excluding first and last) and replace it with a nearby QWERTY key. For example, `a` can be replaced by `s`, `q`, `w`, or `e`; `e` by `w`, `r`, `d`, or `s`.
- Constraint: Typos are only introduced if the word was not already transformed by synonym replacement, ensuring transformations don't compound.

Implementation Details

The transformation pipeline processes text as follows:

1. **Tokenization:** Use NLTK's `word_tokenize()` to preserve sentence structure
2. **POS Tagging:** Tag tokens with NLTK's `pos_tag()` to identify word types
3. **Transformation Stages:** Apply transformations in order: (1) contraction handling, (2) synonym replacement (domain groups then WordNet), (3) verb tense changes, (4) typo introduction
4. **Capitalization Preservation:** All replacements maintain the original word's capitalization pattern (uppercase, lowercase, title case)
5. **Detokenization:** Reconstruct sentences using `TreebankWordDetokenizer()` to restore natural spacing and punctuation

Why This Transformation is Reasonable

- **Realistic Variations:** Synonym usage, tense changes, contractions, and typos all occur naturally in user-generated content, especially in informal movie reviews.
- **Preserves Semantics:** All transformations maintain the core meaning and sentiment of the original text, ensuring labels remain valid.
- **Appropriate Complexity:** The transformation is non-trivial (requires semantic understanding) but not extreme (text remains readable and interpretable).
- **Test-Time Plausibility:** These variations are commonly observed in real-world test scenarios, making the OOD evaluation meaningful.

Example Transformation

Original: "Titanic is the best movie I have ever seen. The acting was excellent and the story was compelling."

Transformed: "Titanic was the finest film I have ever seen. The performance was outstanding and the narrative was compelling."

Changes: `is` → `was` (tense), `best` → `finest` (synonym), `movie` → `film` (domain group), `acting` → `performance` (domain group), `excellent` → `outstanding` (domain group), `story` → `narrative` (domain group).

Q3. 1

Report & Analysis

Accuracy Results

Model	Original Test Set	Transformed Test Set
Baseline (no augmentation)	XX.XX%	XX.XX%
With data augmentation	XX.XX%	XX.XX%

Table 1: Accuracy on original and transformed test sets.

Note: Replace XX.XX% with actual accuracy values after running evaluations.

Analysis and Discussion

(1) Model Performance on Transformed Test Data After Augmentation:

After applying data augmentation during training, the model's performance on the transformed test data [improved/decreased/remained similar] from XX.XX% to XX.XX%. This suggests that [the augmentation strategy successfully exposed the model to similar transformations during training, improving robustness / the augmentation was insufficient or introduced noise that hindered learning / etc.].

(2) Impact of Data Augmentation on Original Test Data:

Data augmentation [enhanced/diminished] the model's performance on the original test set from XX.XX% to XX.XX%. This indicates that [the augmented training data helped the model learn more generalizable features that also benefit performance on standard test data / the augmentation introduced distribution shift that hurt performance on the original distribution / etc.].

Intuitive Explanation

The observed results can be explained by considering how data augmentation affects model training:

- **Improved OOD Robustness:** When the model is trained on augmented data that includes synonym replacements, tense changes, contractions, and typos, it learns to focus on semantic patterns rather than memorizing specific lexical choices. This makes it more robust to the lexical variations present in the transformed test set.
- **Regularization Effect:** The augmented training examples act as a form of regularization, preventing the model from overfitting to exact word sequences and encouraging it to learn more generalizable representations.
- **Trade-off with Original Distribution:** If augmentation is too aggressive or introduces patterns not present in the original distribution, it may hurt performance on the original test set. Conversely, if augmentation is well-calibrated, it can improve generalization to both original and transformed data.
- **Semantic Preservation:** Since our transformations preserve semantic meaning and sentiment, the model learns that different surface forms can express the same underlying sentiment, improving its ability to handle natural language variation.

Limitation of the Data Augmentation Approach

One key limitation of this data augmentation approach for improving OOD performance is that it only addresses **lexical and surface-level variations** (synonyms, typos, contractions, tense changes) but does not handle **semantic or structural distribution shifts**. For example:

- The augmentation cannot help with domain shifts where the underlying sentiment indicators change (e.g., reviews from a different genre or cultural context where "dark" might be positive rather than negative).

- It does not address cases where the transformation introduces subtle semantic shifts that our heuristics fail to detect, potentially creating label noise.
- The approach assumes that synonym replacement and other transformations perfectly preserve sentiment, which may not always hold true in practice (e.g., "terrible" vs. "poor" may have different intensity levels).
- The augmentation is limited to transformations we can programmatically define, missing more complex linguistic variations like paraphrasing, style changes, or discourse-level modifications that occur in real OOD scenarios.

To truly improve OOD robustness, we would need to combine lexical augmentation with other techniques such as adversarial training, domain adaptation, or learning from diverse data sources that capture broader distribution shifts.

Part II. Q4

Statistics Name	Train	Dev
Number of examples	4225	466
Mean sentence length	10.96	10.91
Mean SQL query length	60.90	58.90
Vocabulary size (natural language)	868	444
Vocabulary size (SQL)	644	393

Table 2: Data statistics before any pre-processing.

Statistics Name	Train	Dev
T5 fine-tuned model		
Mean sentence length (tokens)	17.10	17.07
Mean SQL query length (tokens)	216.37	210.05
Vocabulary size (natural language, token IDs)	791	465
Vocabulary size (SQL, token IDs)	555	395

Table 3: Data statistics after pre-processing.

Q5

Design choice	Description
Data processing	Describe the data processing steps you undertook, if any.
Tokenization	Describe how you did the tokenization for the inputs to the encoder and decoder. If you use anything else than the default T5 tokenizer, specify what you use, and why you choose to use it.
Architecture	Describe the components in the T5 architecture that you chose to fine-tune. Did you fine-tune the entire model, specific layers?
Hyperparameters	List the key hyperparameters that you used, including the learning rate, batch size, and stopping criteria.

Table 4: Details of the best-performing T5 model configurations (fine-tuned)

Q6.

System	Query EM	F1 score
Dev Results		
T5 fine-tuned		
Full model	XX.XX	XX.XX
Test Results		
T5 fine-tuning	XX.XX	XX.XX

Table 5: Development and test results. Use this table to report quantitative results for both dev and test results.

Quantitative Results:

Qualitative Error Analysis:

Error Type	**Example Of Error**	**Error Description**	**Statistics**
Error name	Snippet from datapoint examplifying error	Describe the error in natural language	Provide statistics in the form "COUNT/TOTAL" on the prevalence of the error. TOTAL is the number of relevant examples (e.g. number of queries, for query-level error), and COUNT is the number of examples that showed this error.

Table 6: Use this table for your qualitative analysis on the dev set.

Q7.

Provide a link to a google drive which contains a model checkpoint used to generate outputs you have submitted. TODO

Extra Credit:

If you are doing extra credit assignment, please describe your system here, as well as provide a link to a google drive which contains a model checkpoint used to generate outputs you have submitted. Optional TODO