Learning to Describe Solutions for Bug Reports Based on Developer Discussions

The University of Texas at Austin

https://github.com/panthap2/describing-bug-report-solutions

Sheena Panthaplackel(spantha@cs.utexas.edu), Junyi Jessy Li, **Milos Gligoric, Raymond J. Mooney**
The University of Texas at Austin

Motivation

- When a bug is reported, developers engage in a dialogue to collaboratively understand it and ultimately resolve it with the necessary code changes.
- To expedite bug resolution, we aim to guide developers in better absorbing content relevant towards implementing the solution from long bug report discussions.

Task

Generate concise natural language description of the solution by synthesizing relevant content in the discussion **when it emerges in real-time**

Title: Incorrect distance

Utterance #1
Seeing negative distance when using 1D grid.

Utterance #2
Probably a bug in `getL1Distance(int x1, int x2)`.

Utterance #3
We do `x1 - x2`, which will be negative if `x1 < x2`.

Utterance #4
We should compute its absolute value.

NL Solution Description
Compute absolute value of `x1 - x2` in `getL1Distance`

Data

~12.3K bug reports for open-source Java projects that are linked to a commit/PR

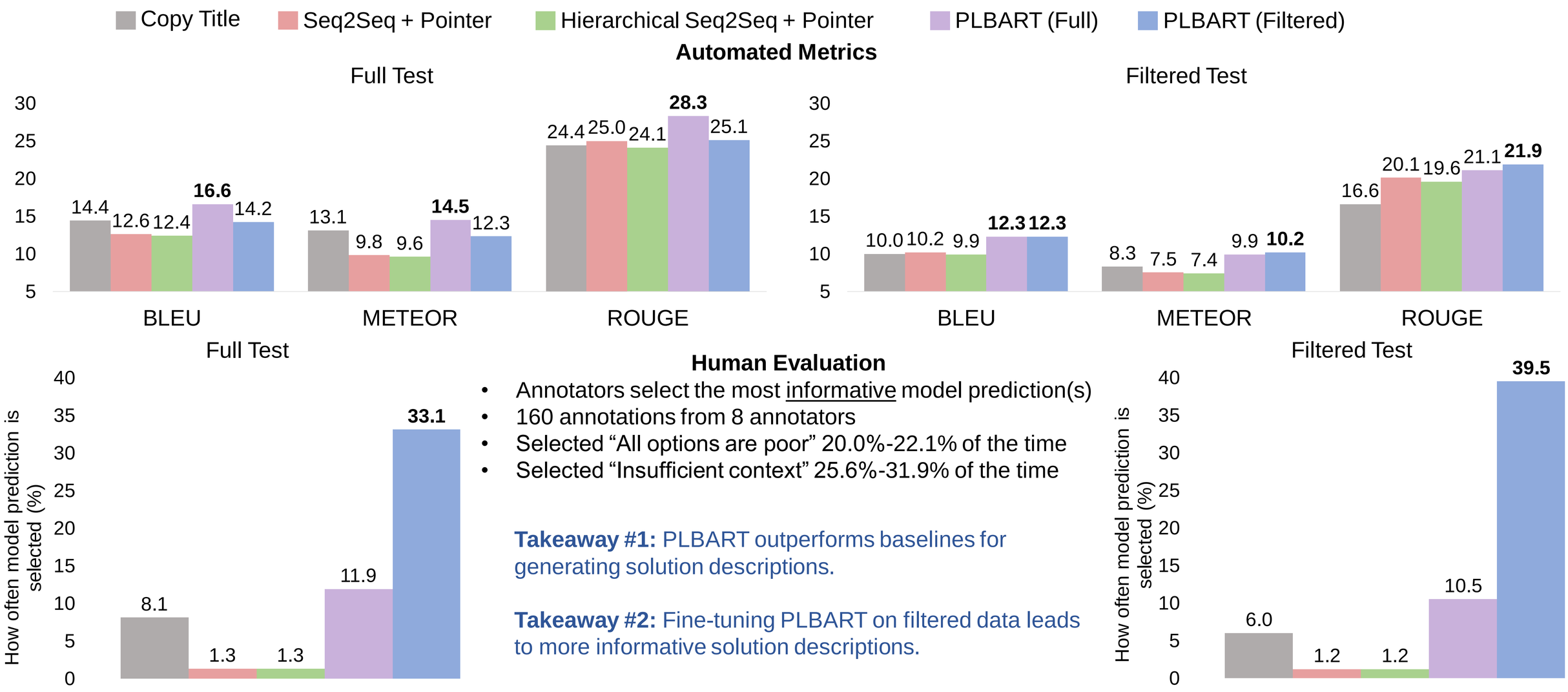
Time step of commit/PR

Commit message/PR title

dev007 added a commit that referenced this issue

After filtering noisy examples with generic descriptions, uninformative descriptions, and discussions with insufficient context: **~5.9K**

Benchmarking Generating Solution Descriptions



Human Evaluation

- Annotators select the most informative model prediction(s)
- 160 annotations from 8 annotators
- Selected “All options are poor” 20.0%-22.1% of the time
- Selected “Insufficient context” 25.6%-31.9% of the time

Takeaway #1: PLBART outperforms baselines for generating solution descriptions.

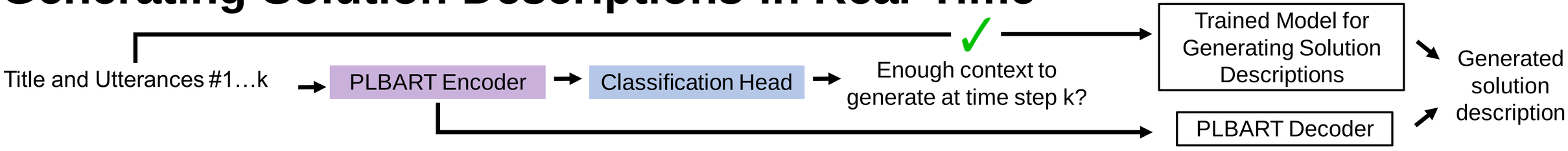
Takeaway #2: Fine-tuning PLBART on filtered data leads to more informative solution descriptions.

Model	Prediction
Copy Title	black screen appears when we seek over an ad group .
Seq2Seq + Pointer	fix black ads
Hierarchical Seq2Seq + Pointer	fix seeking in ad tag
PLBART (Full)	suppress closing shutter when seeking over an ad group
PLBART (Filtered)	suppress closing the shutter when seeking to an unprepared period
Reference	prevent shutter closing for within-window seeks to unprepared periods

Open Challenges:

- Disentangling content related to the solution from elaboration of the problem
- Synthesizing content when information related to the solution spans multiple utterances
- Bimodal reasoning across language and in-lined code

Generating Solution Descriptions in Real-Time



Human Evaluation (120 annotations from 60 annotators)	**Pipelined**	**Joint**
Scenario #1: System generates at time step k	64.6%	63.6%
Is there sufficient context about the solution at time step k?	39.0%	33.8%
Rate the informativeness of the generated description: *1 - Incomprehensible, completely incorrect, irrelevant* *2 - Generic, rephrasing the problem* *3 - Includes some useful information but does not capture the solution* *4 - Partially captures solution* *5 - Completely captures solution*	3.3	3.3
Scenario #2: System refrains from generating	35.4%	36.4%
Is there sufficient context about the solution at any point in the discussion?	34.2%	37.0%

Takeaway #3: When sufficient context is available, system output is useful.

Takeaway #4: Balancing the trade-off between generating too early and deferring to later time steps is an open challenge.