Classifying Stack Overflow Questions using attention based mechanisms

Background Information

1. **StackOverflow**

   A website that is used to answer queries/issues across different domains of development and thus disseminating information.

2. **Quality-signaling segments**

   Parts of a question that can be identified for detecting the quality of it.

3. **Low Quality Posts**

   Posts that lack the clarity, context, or effort needed for others to give a meaningful answers.

Background Information

4. Cohen's kappa

$$\kappa = \frac{p_0 - p_e}{1 - p_e}$$

A statistic that is used to measure how much two raters agree beyond what you'd expect by chance.

p_0 → proportion of **observed** agreement b/w 2 raters

p_e → proportion of agreement expected by chance

RESEARCH PAPER #1

An Empirical Study of Obsolete Answers on Stack Overflow

2019

Key Findings

1. This study reveals that **58.4%** of obsolete answers on Stack Overflow were flagged within 24hrs of being posted.

2. **20.5%** of obsolete answers are ever updated, highlighting significant challenges in maintaining answer quality over time.
   Only **6.3%** got a new, updated answer added.

3. Some technologies are much more likely to have obsolete answers. (according to the paper; node.js, ajax, android)

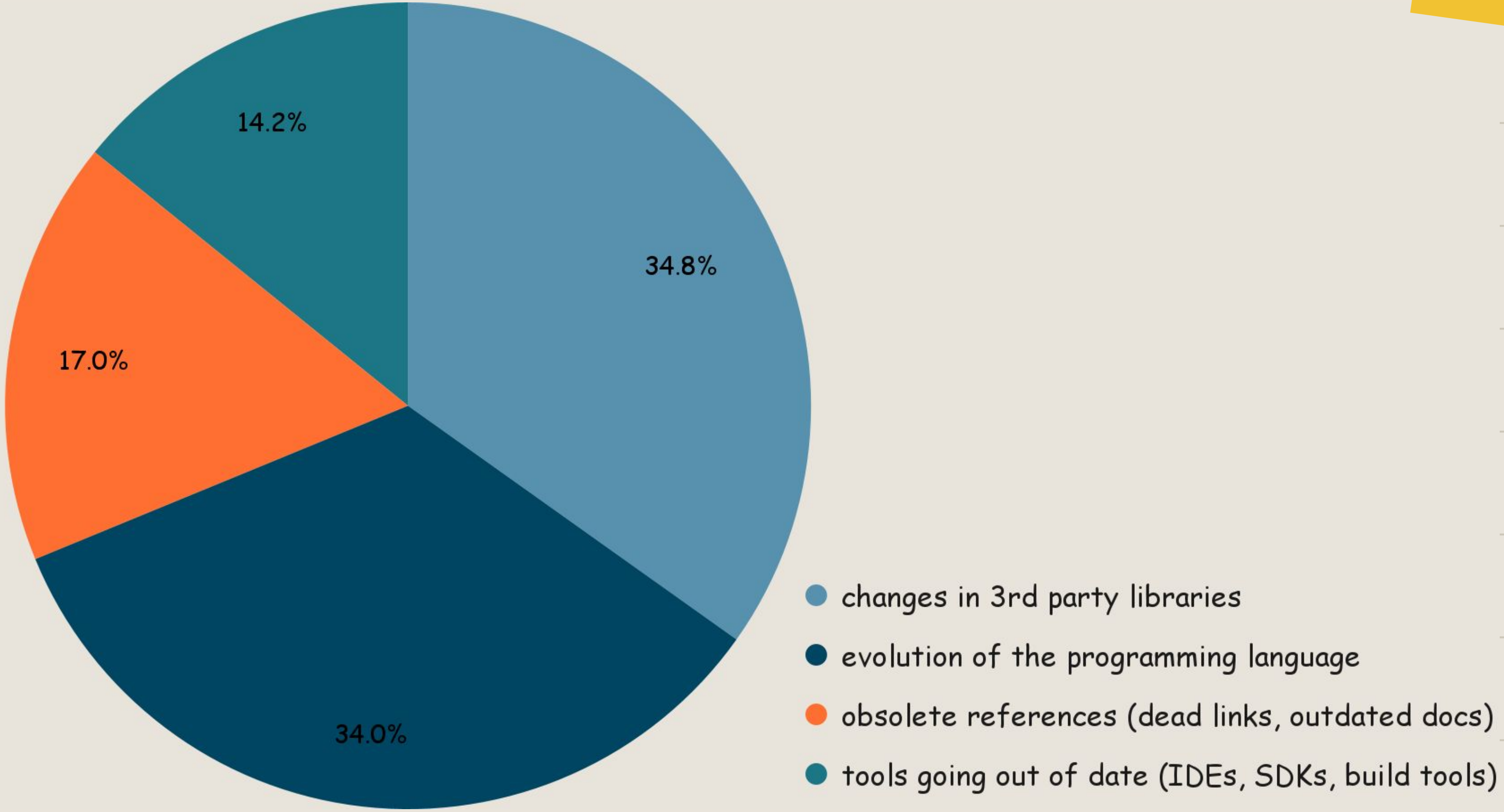
4. Certain **tags** are more prone to get obsolete.

5. People do provide evidence when they call something obsolete in the comments (and it's usually the most upvoted solution)

RESEARCH PAPER #1

An Empirical Study of Obsolete Answers on Stack Overflow

2019



RESEARCH PAPER #1

An Empirical Study of Obsolete Answers on Stack Overflow

2019

Methodology Used

1. The authors combined **qualitative analysis** (coding done in multiple phases by two authors - Cohen's kappa of 0.76–0.96) and **quantitative analysis** of 52,177 answer threads from Stack Overflow's data dump (August 2017), spanning 12,629 tags.
2. Heuristic used to classify was done by randomly sampling 669 candidate obsolete answers and manually checking them (using docs/API references where needed) to see if the answer really was obsolete.
3. Tags prone to obsolete answers were computed using a basic formula. (obsolescence ratio)
4. Crawled 5.5M links in 7.3M answers to see which URLs still returned HTTP 200; found 11.9% dead links

RESEARCH PAPER #1

An Empirical Study of Obsolete Answers on Stack Overflow

2019

Relevance

The paper is crucial for the project as it helps understand and demonstrates how Stack Overflow content evolves and becomes outdated over time, particularly for rapidly evolving technologies and the need to track and gauge this trend.

RESEARCH PAPER #2

Analysis of Modern Release Engineering Topics

2020

Key Findings

1. **How** do I set up/fix/configure X?

   What - definitions, explanations, clarifications

   Why - causes, rationale, “why does this break”

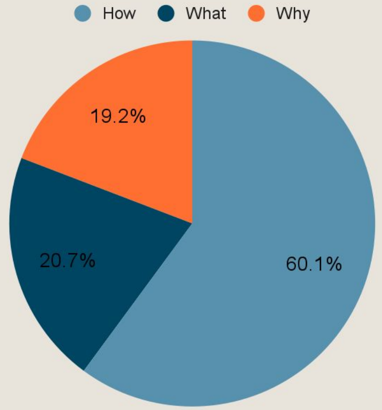
2. What’s popular vs. what devs ignore

   Popular - Merge conflicts, branching & remote upstream and feature expansion

   Least popular - web deployment

3. They checked correlation between popularity metrics and difficulty metrics and found it to be statistically negative; more popular topics tend to be less difficult

What kind of questions release engineers ask



RESEARCH PAPER #2

Analysis of Modern Release Engineering Topics

2020

Methodology used

1. **SOTorrent** dataset - A dump based on the official Stack Overflow (2008-2019)
2. Pre-processing → Stripped HTML tags, codeblocks, URLs, punctuation, numbers and stopwords and applied Porter stemming
3. Authors inspected per topic, the top 30 words and assigned human readable labels

Relevance

This helps in exploring and understanding various ways of tagging questions and identifying their and building what features to look for when building an attention based mechanism

RESEARCH PAPER #3

How do professionals perceive legacy systems and software modernization?

2014

Key Findings

1. How practitioners define a legacy system?
2. Industry vs academia: perception gap

Methodology Used

1. Exploratory **qualitative** phase (grounded theory approach + interviews)

   Conducted semi-structured, in-depth interviews with 26 practitioners from 22 organizations in various domains. Open coding transcripts, memping and snowball sampling.
2. **Quantitative** validation phase

   They conducted a structured online survey (198 responses)directly from the interview findings to validate those results.

RESEARCH PAPER #3

How do professionals perceive legacy systems and software modernization?

2014

Key Findings

1. How practitioners define a legacy system?
2. Industry vs academia: perception gap

Methodology Used

1. Exploratory **qualitative** phase (grounded theory approach + interviews)
   Conducted semi-structured, in-depth interviews with 26 practitioners from 22 organizations in various domains. Open coding transcripts, memping and snowball sampling.
2. **Quantitative** validation phase
   They conducted a structured online survey (198 responses)directly from the interview findings to validate those results.

RESEARCH PAPER #4

Improving Low Quality Stack Overflow Post Detection

2014

Key Findings

1. Stack Overflow's automatic system that pushes potentially "**low-quality**" posts into a review queue for humans to check has roughly ⅓ of them being tagged falsely.
2. Proposed a new system that reduces the queue by 44%.
3. Popularity metrics and readability metrics are powerful signals for classification.

RESEARCH PAPER #4

Improving Low Quality Stack Overflow Post Detection

2014

Methodology Used

1. Simple **textual** features

   Body length, number of URLs, title length
2. **Readability** and structural metrics

   Number of sentences, percentage of code, entropy of terms
3. User popularity/reputation features
   - upvotes/downvotes received
   - badges (question/answer badges)
   - favorites, close/delete votes
4. Built a quality function by defining a linear scoring model.
5. They use the score distribution to classify “very good” (A+B) and “very bad” (C+D)

RESEARCH PAPER #4

Improving Low Quality Stack Overflow Post Detection

2014

Relevance

This paper gives a comprehensive understanding of what features to include and what to look for while trying to build a model as it explores a variety of features for improving the detection of low quality posts

Proposed Project

Building an LSTM model based on the Kaggle Dataset that could potentially give insights into words that should be paid attention to while trying to classify a post.

Research Question:
Can attention-based models outperform feature-based methods in detecting low-quality posts by identifying quality-relevant text segments

Dataset

The dataset contains 60k Stack Overflow questions

- It ranges from 2016-2020
- Classification is done 3 ways:
    - HQ: High-quality posts without a single edit.
    - LQ_EDIT: Low-quality posts with a negative score, and multiple community edits. However, they still remain open after those changes.
    - LQ_CLOSE: Low-quality posts that were closed by the community without a single edit.
- Split the dataset into training and validation sets (70/30 split)

Proposed Project
Goals for building a LSTM classifier

1.	**Build an embedding layer**	Use a vocabulary size appropriate for the corpus and an embedding dimension
2.	**Self-attention layer**	The attention mechanism should compute attention weights across all timesteps.
3.	**Expected Results for evaluation**	The self-attention layer should produce a context vector that represents the salient parts of the question that helps in visualizing attention weights

Questions?