Fast Detection of Duplicate Bug Report Using LDA-based Topic Modeling and Classification

Lakehead University

Department of Computer Science

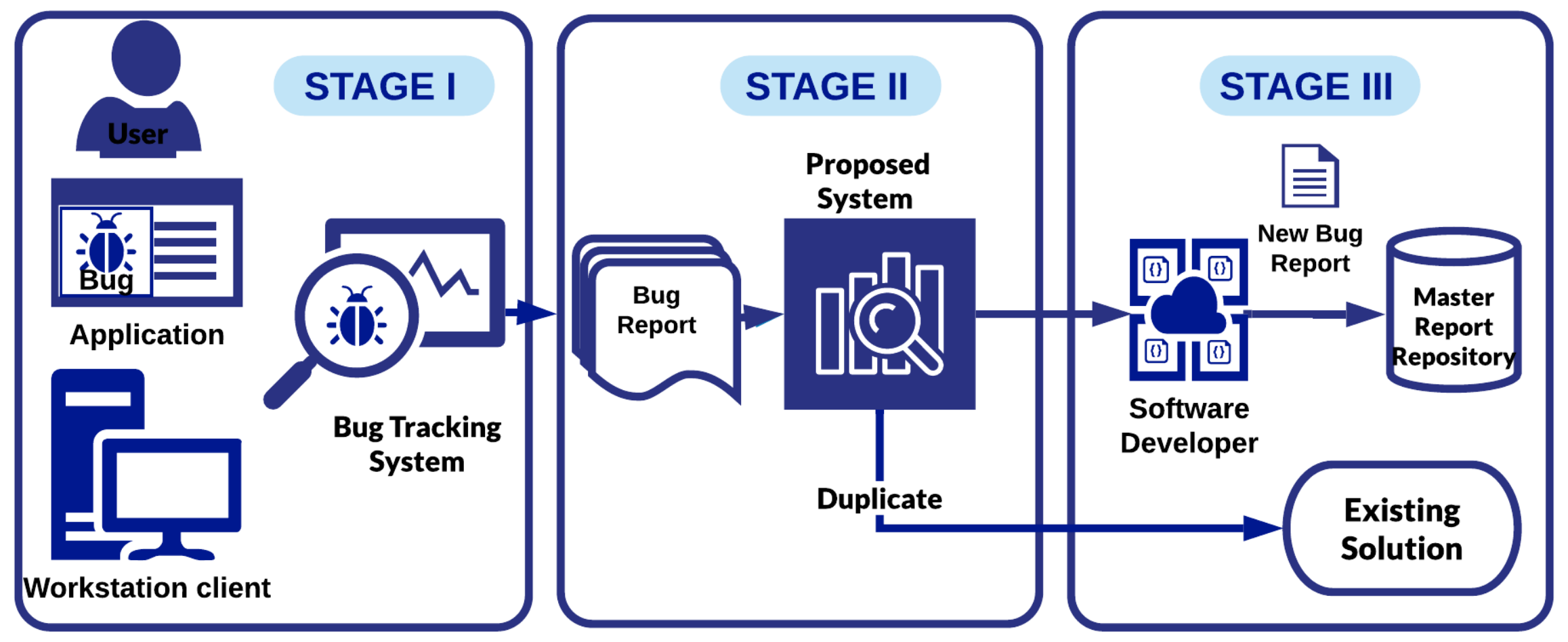
Team: Dhruvit Shah, Nishi Patel, Rinkal Mehta, Surya Yakkanti

Instructor : Dr. Thangarajah Akilan

Abstract

- During the development and maintenance of a software, a bug is detected by the bug tracking system and a bug report is generated.
- Identical problem: There can be already existing bug reports, which were reported to the developer earlier, but the same problem can be reported over and over.
- It causes waste of development time solving a solved bug. So, it is extremely important to detect the duplicate bug reports.
- The proposed solution predicts the overwhelming of incoming bug reports as duplicates entries or not using LDA-based clustering and ML-based classification in a time efficient manner.

Overview



Stage 1: A bug is detected by Bug Tracking System (BTS).

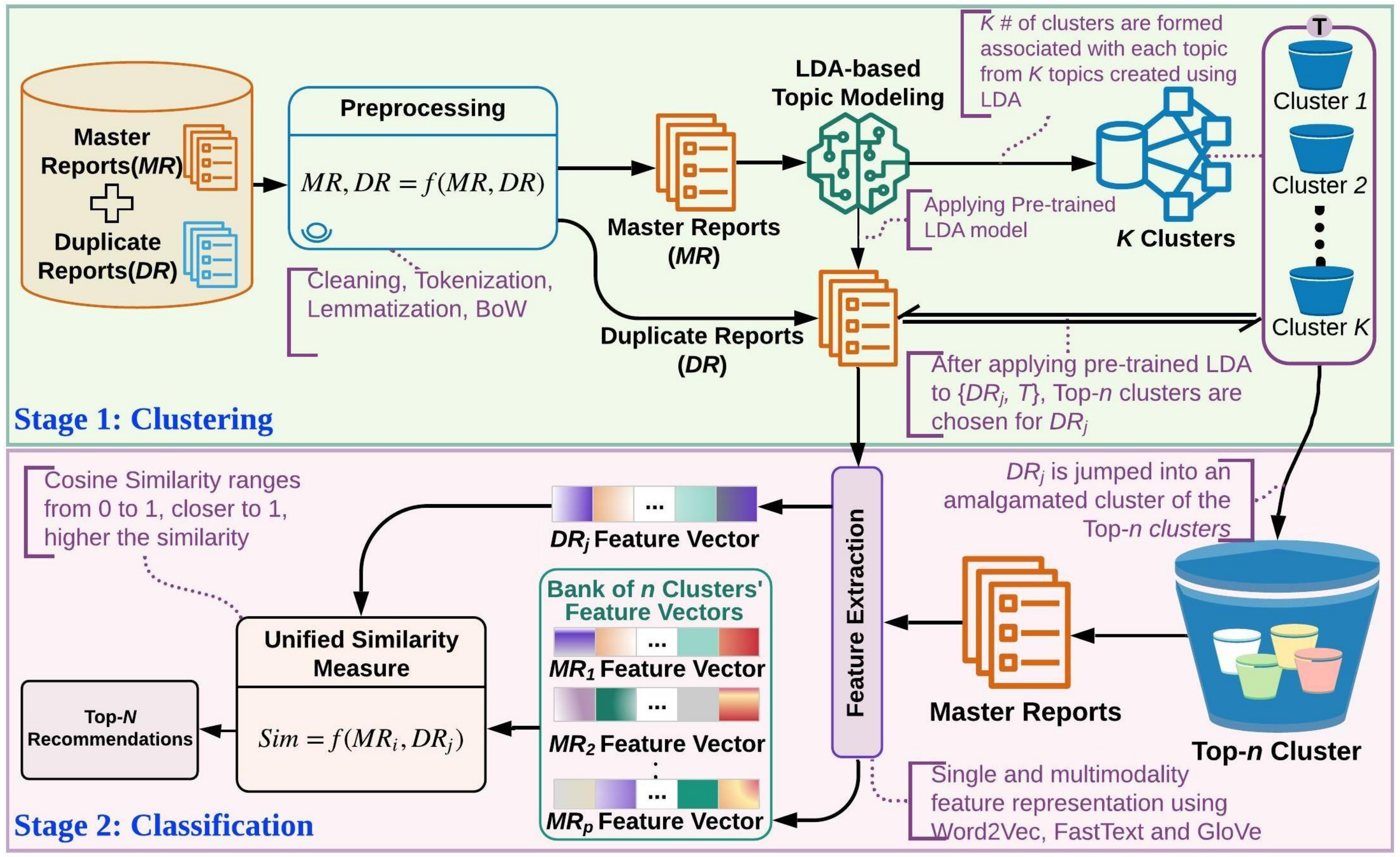
Stage 2: A bug report (BR) is generated by BTS is submitted to proposed system. It identifies the BR as duplicate or not.

Stage 3: If BR is found to be duplicate then it has an existing solution. Otherwise, it will be sent to the software developer to fix that bug, and its added to Master Report Repository.

Challenges

- Ambiguity on lexical, syntactic and semantic levels.
- High memory problem.
- Word sense disambiguation.
- Data-related Problems.
- High processing time requirement.
- Platform compatibility.

Proposed Model



Master Reports(*MR*)
Duplicate Reports(*DR*)
Preprocessing
$MR, DR = f(MR, DR)$
Cleaning, Tokenization, Lemmatization, BoW
Master Reports (*MR*)
Duplicate Reports (*DR*)
LDA-based Topic Modeling
K # of clusters are formed associated with each topic from *K* topics created using LDA
Applying Pre-trained LDA model
K Clusters
T
Cluster *1*
Cluster *2*
Cluster *K*
After applying pre-trained LDA to {DR_j, *T*}, Top-*n* clusters are chosen for DR_j
Stage 1: Clustering
Cosine Similarity ranges from 0 to 1, closer to 1, higher the similarity
DR_j Feature Vector
...
DR_j is jumped into an amalgamated cluster of the Top-*n clusters*
Feature Extraction
Bank of *n* Clusters' Feature Vectors
MR_1 Feature Vector
MR_2 Feature Vector
MR_p Feature Vector
Unified Similarity Measure
$Sim = f(MR_i, DR_j)$
Top-*N* Recommendations
Master Reports
Top-*n* Cluster
Single and multimodality feature representation using Word2Vec, FastText and GloVe
Stage 2: Classification

Stage 1: Clustering

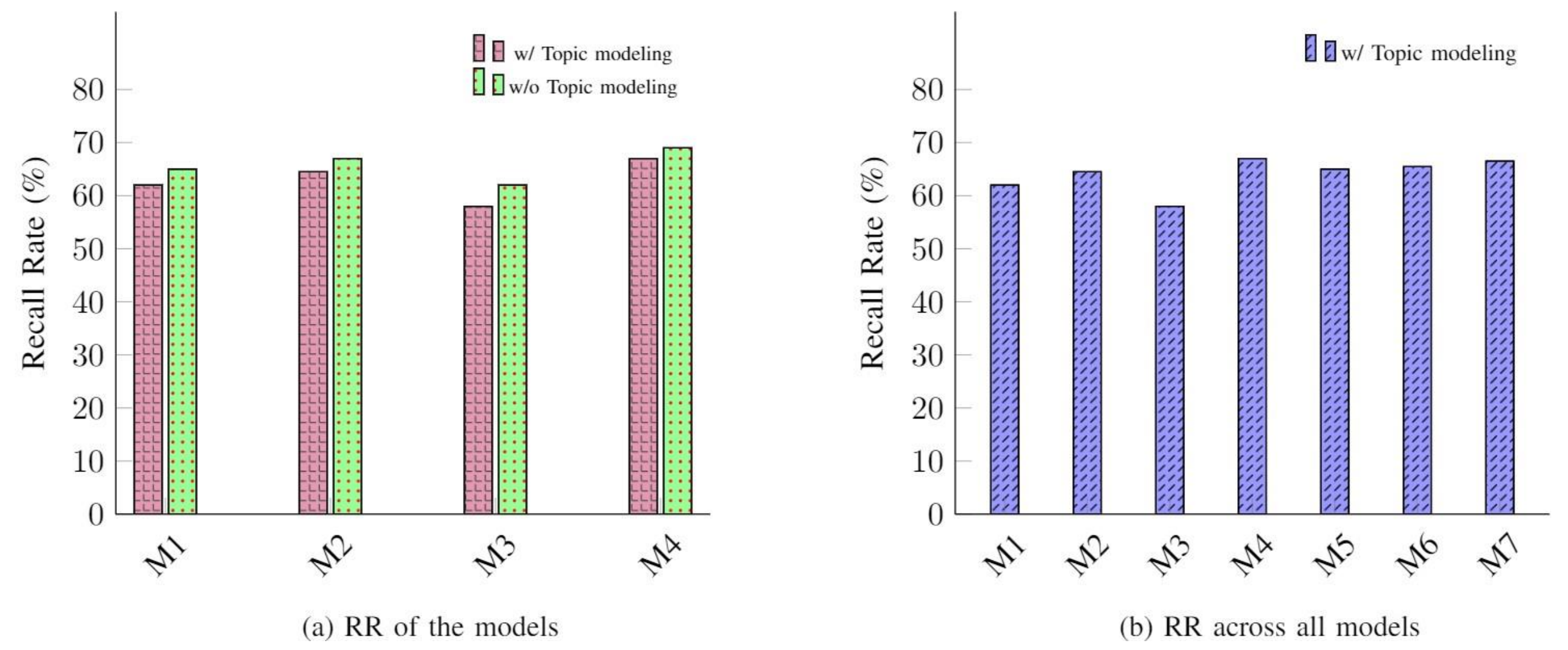
- LDA is applied on the preprocessed master reports to form clusters.
- Pre-trained LDA is applied on the preprocessed duplicate report to find the most relevant cluster in which associated master report may exist.

Stage 2: Classification

- **Home cluster:** The duplicate report jumps into the selected cluster to find the most similar master report.
- **Finding the MR:**
  - Similarity between feature vectors of the duplicate report and the master reports in the corresponding cluster individually.
  - Top-n cosine similarities would be selected which would result in top-n master reports.

Results

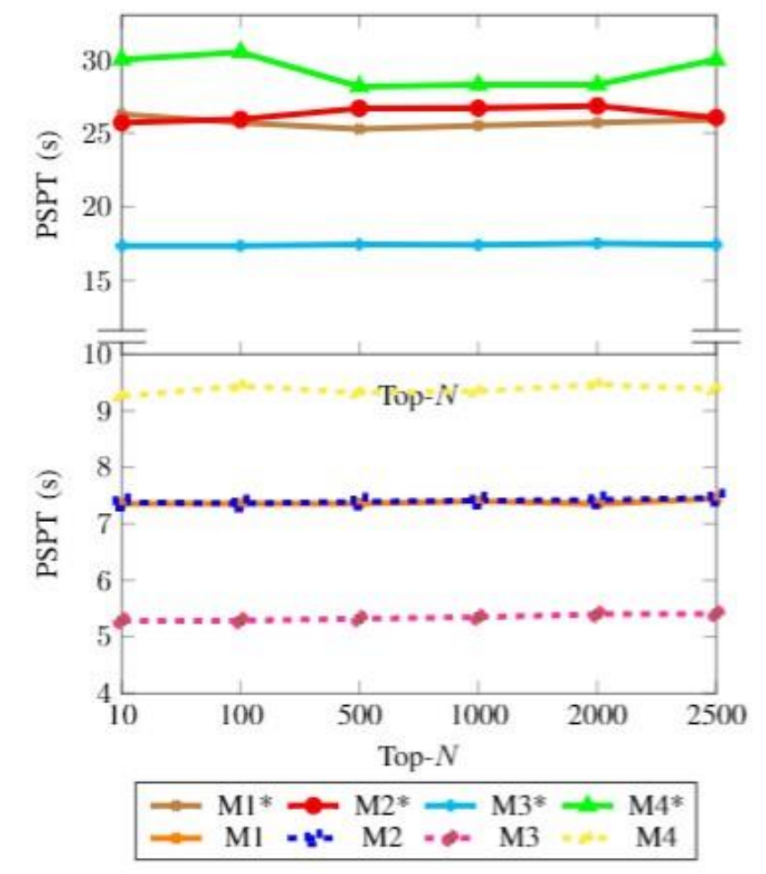
Performance Comparison of All the Models for Top-2.5K Recommendation.



(a) RR of the models

(b) RR across all models

Per Sample Processing Time of all the Models:(*) - without Proposed Topic Modeling.

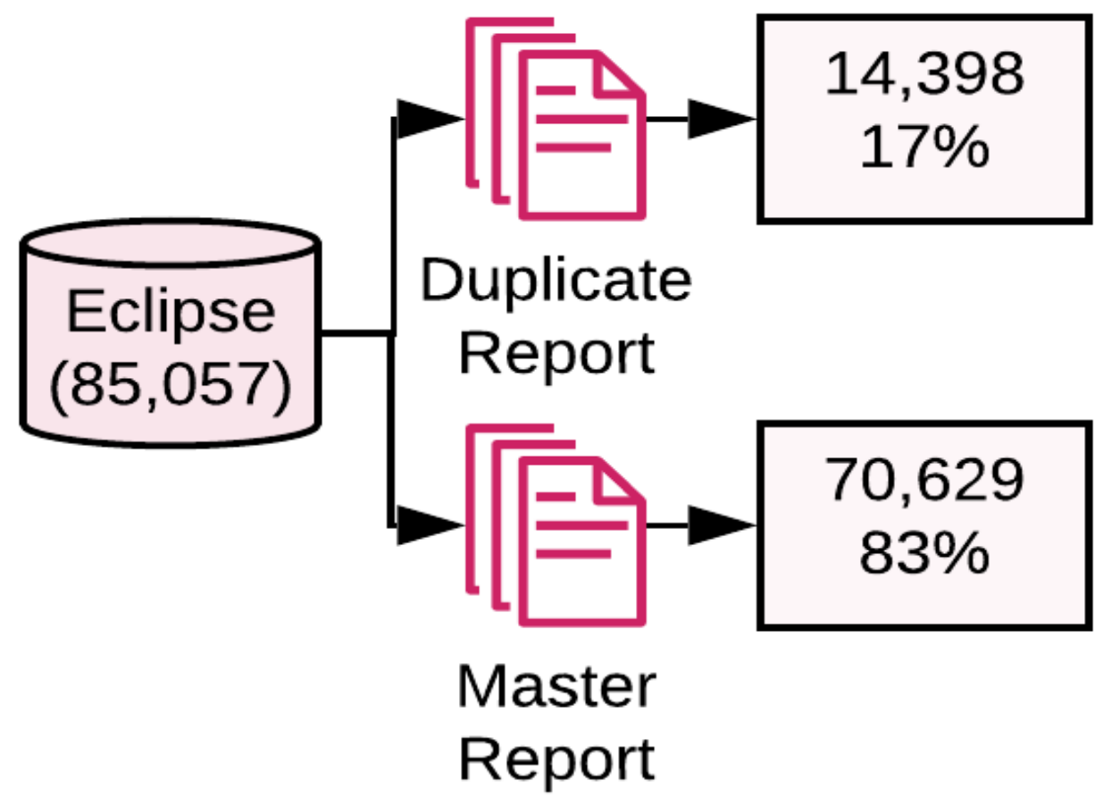


Key Components:

- **Latent Dirichlet Allocation (LDA):** Used for topic modelling and clustering.
- **Feature Extraction Techniques:**
  Single-Modality Feature Extraction: It employs the following three feature extractors individually Word2Vec (**M1**),FastText (**M2**), and GloVe (**M3**).
  Multi-Modality Feature Extraction: The proposed approach exploits multi-modality feature extraction by integrating multiple feature vectors to enhance the performance. Consequently, four multi-modal feature representations are introduced as follows: fusion of FastText and GloVe (**M4**), fusion of GloVe and Word2Vec (**M5**), fusion of FastText andWord2Vec (**M6**), and fusion of FastText, GloVe and Word2Vec(**M7**).
- **Cosine similarity & Euclidean similarity:** Used for document classification.

Preprocessing: Data cleaning, Tokenizing, Removal of stop-words, Lemmatization, Bag of Words (BOW)

Dataset



- Eclipse Dataset is used.
- It contains 85,156 Data rows.
- After preprocessing, 85,027 Data rows are left.
- It includes 70,629 master reports and 14,398 duplicate report.

References

1. A. Sureka and P. Jalote, Detecting duplicate bug report using character n-gram-based features, 2010.
2. J. Zou, L. Xu, M. Yang, M. Yan, D. Yang, and X. Zhang, Duplication detection for software bug reports based on topic model, 2012.
3. C. Sun *et al.*, A discriminative model approach for accurate duplicate bug report retrieval, 2010.
4. J. Zou *et al.*, Auto-mated duplicate bug report detection using multi-factor analysis, 2016.
5. P. Runeson *et al.*, Detection of duplicate defect reports using natural language processing, 2007.