Mining Information from Programming Video Tutorials

Esteban Parra

Florida State University

Tallahassee, Florida

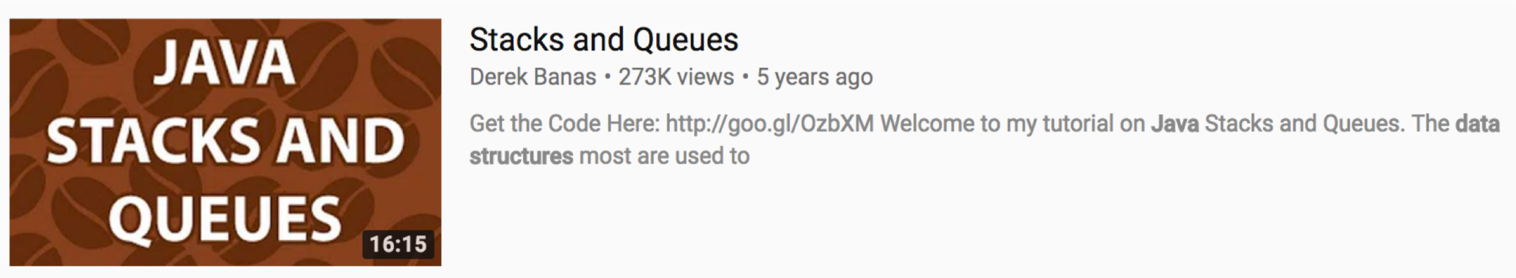
parrarod@cs.fsu.edu

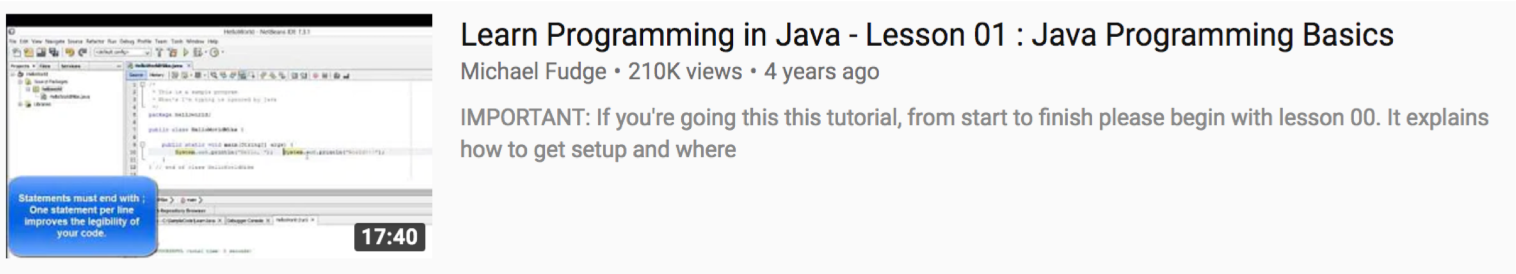
SERENE

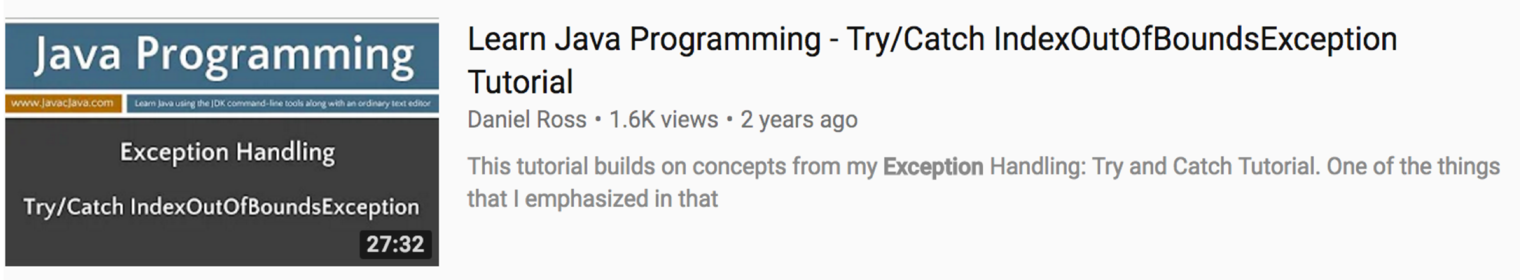
Software Engineering Video Tutorials

Deliver introductory or in-depth information regarding software engineering topics, such as:

- Programming language syntax
- Algorithms or data structures
- Use of APIs
- Error solving

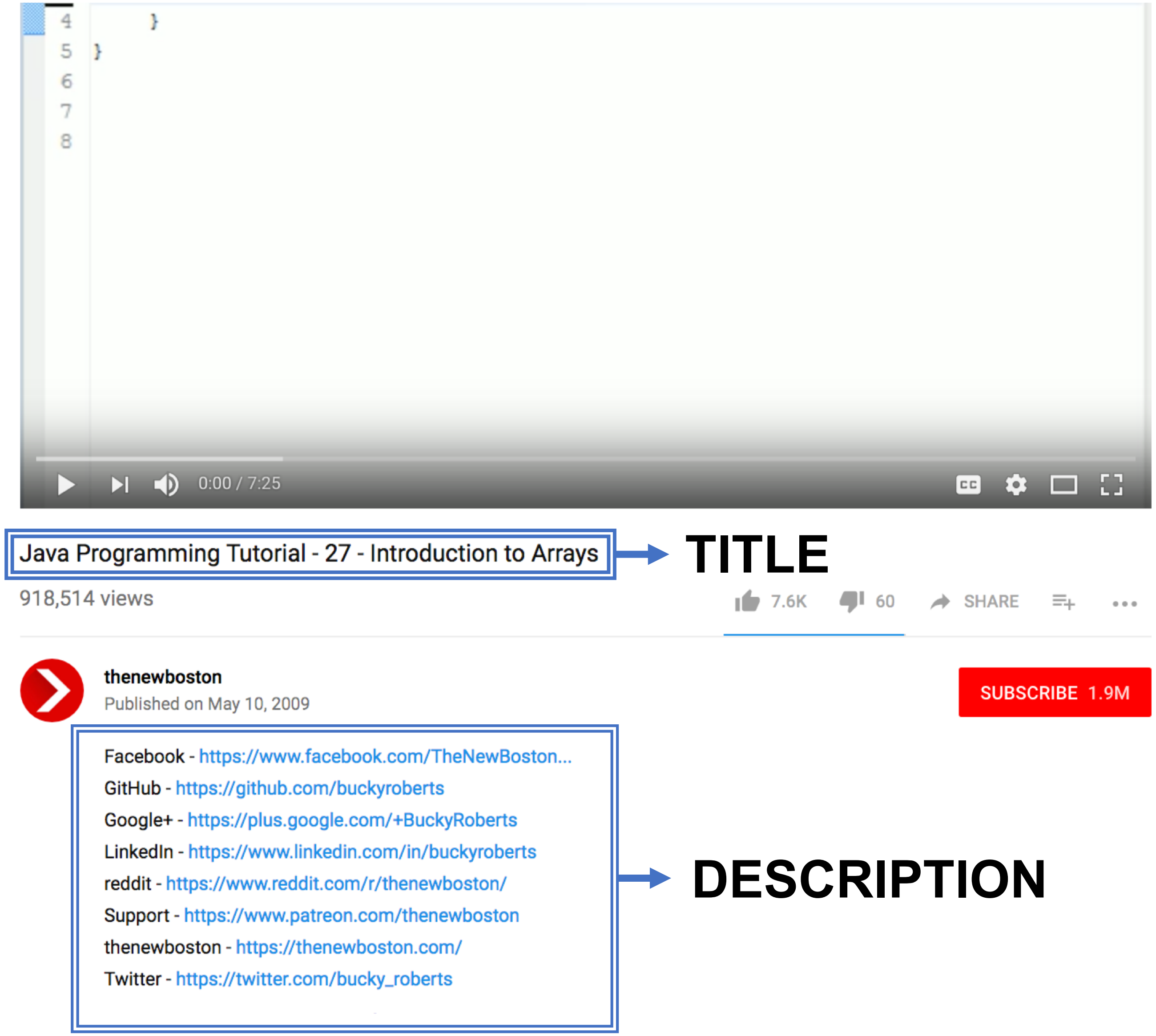






Problem

- Challenging to quickly determine whether a video is relevant
- Description not always concise and informative



Automatic Tag Recommendation for Software Development Video Tutorials

Esteban Parra, Javier Escobar-Avila, Sonia Haiduc

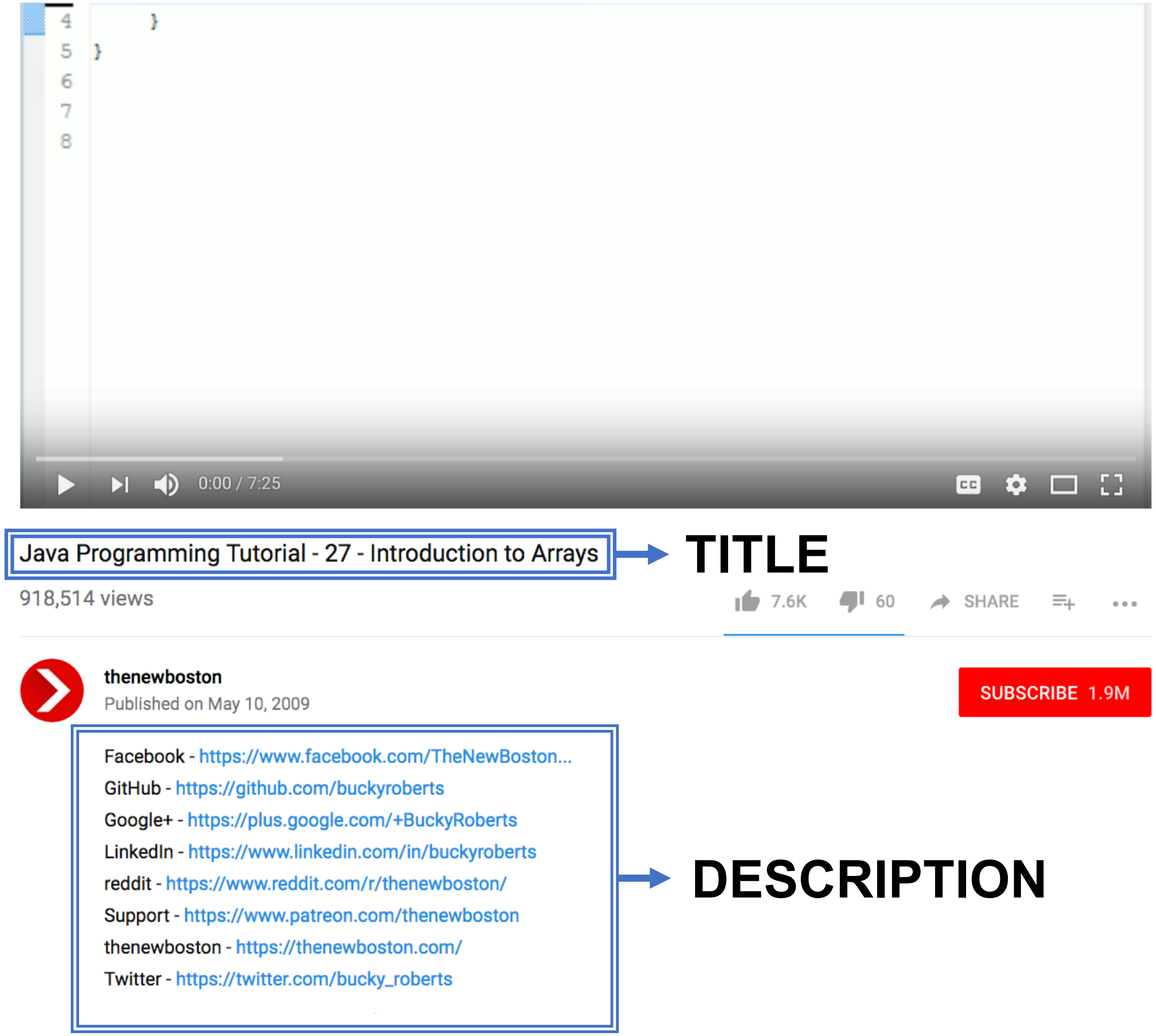
Florida State University

Tallahassee, Florida

parrarod@cs.fsu.edu, escobara@cs.fsu.edu, shaiduc@cs.fsu.edu

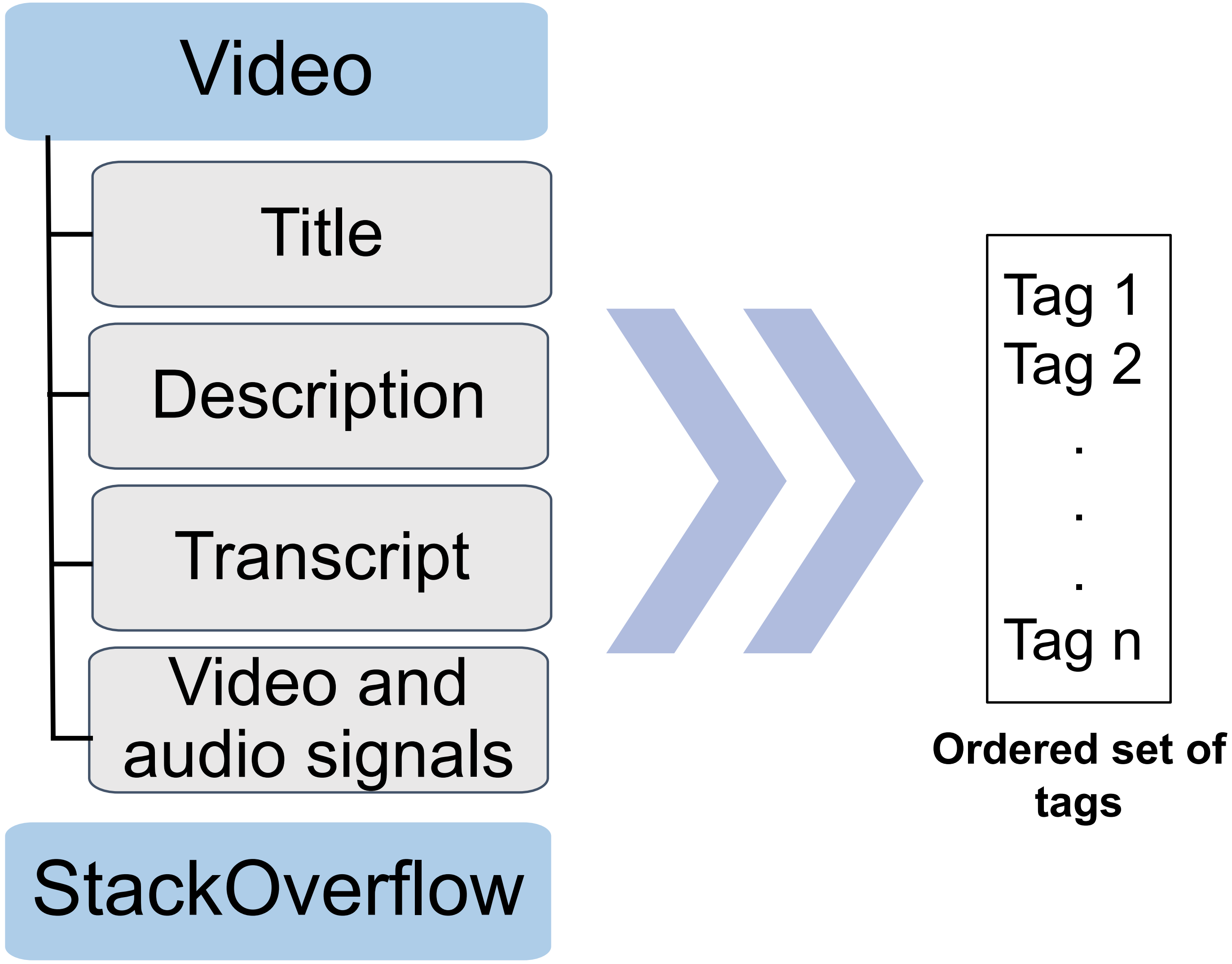
Problem

- Challenging to quickly determine whether a video is relevant
- Description not always concise and informative



Automatic Tag Recommendation

We aim to provide relevant tags describing the content of the video



Automatic Tagging Approaches

Information Retrieval-based	StackOverflow-based	Closed-Source
• TF-IDF • LDA • BM25F	• IR SO-based Tagging • TagMulRec	• Cortical.io • Google Cloud Video Intelligence

Automatic Tagging Approaches

Information Retrieval-based	StackOverflow-based	Closed-Source
• TF-IDF • LDA • BM25F	• IR SO-based Tagging • TagMulRec	• Cortical.io • Google Cloud Video Intelligence

Extractive   Abstractive

Research Questions

RQ1 - What is the quality of the automatically produced tags?

- RQ1.1 Which approach produces the best tags?
- RQ1.2 Do *singularization* and *stemming* impact the performance of the approach?

RQ2 - Where do the relevant tags come from?

- Within the video (Extractive) – title, description, transcript
- Outside of the video (Abstractive)

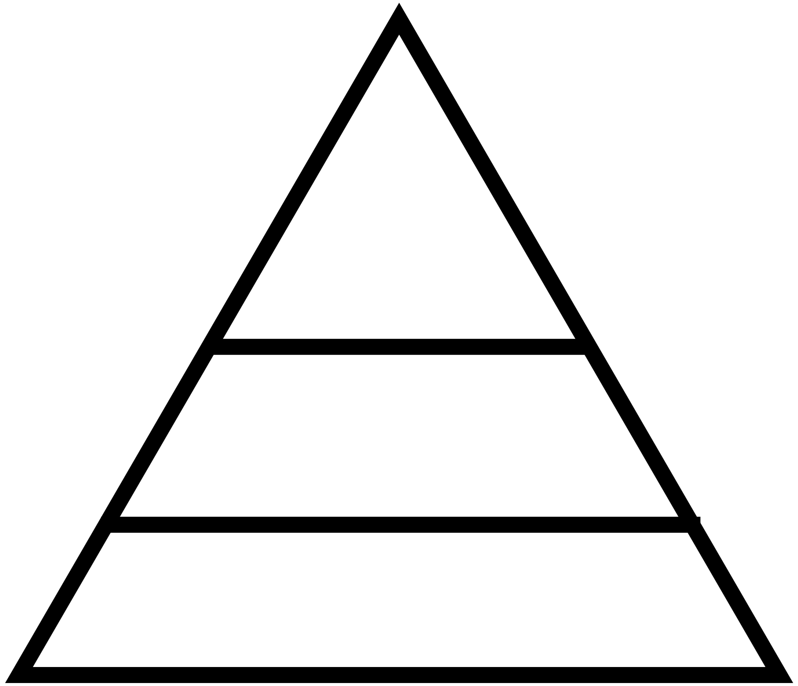
Building a Ground Truth

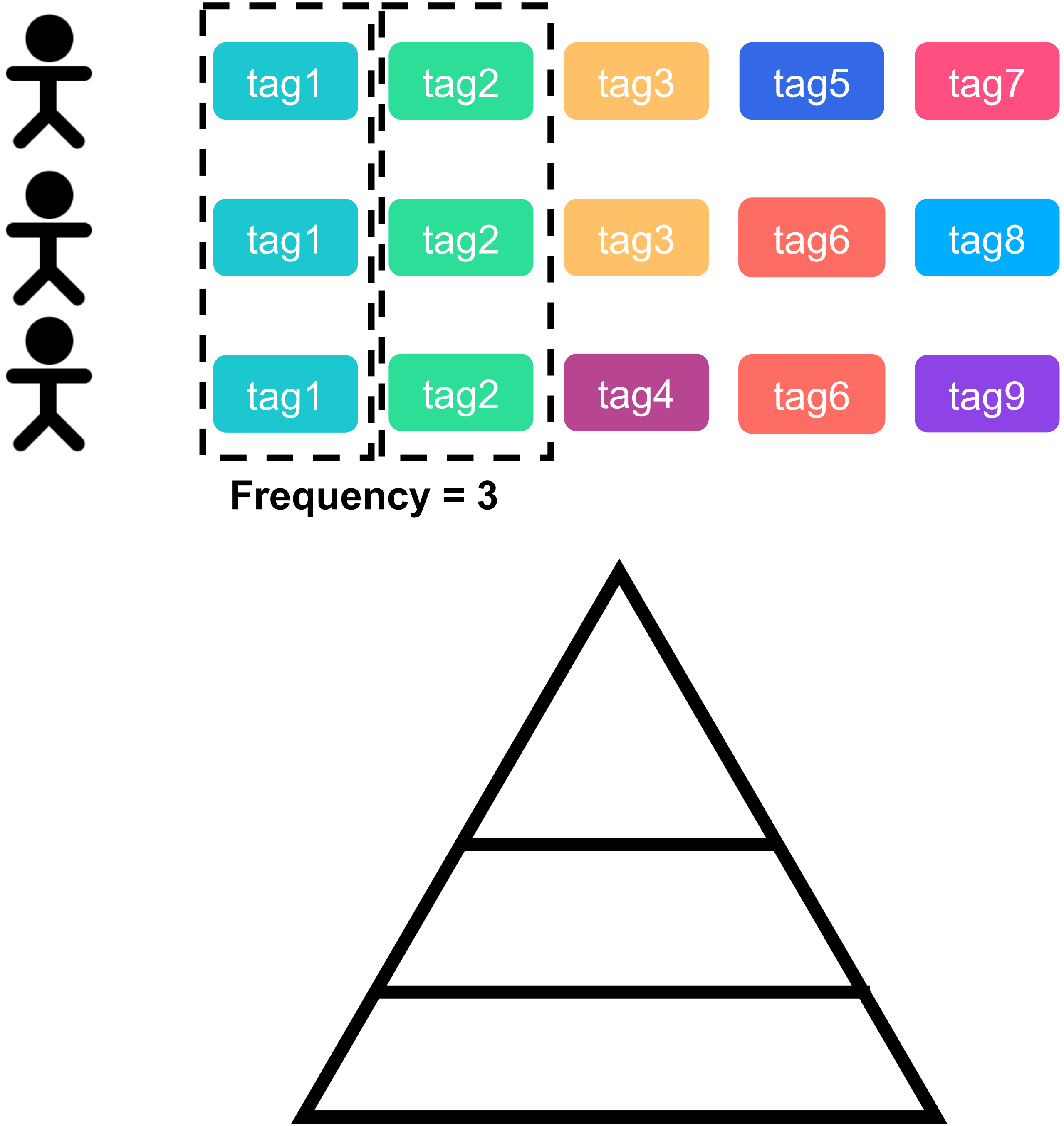
- 57 participants
  - At least 6 months of experience with Java
    - 15 undergraduate students
    - 39 graduate students
    - 2 professional developers
    - 1 faculty member
- 75 Java programming videos
  - Covering various topics
  - Created by various content creators
- Each participant annotated 3 to 5 videos

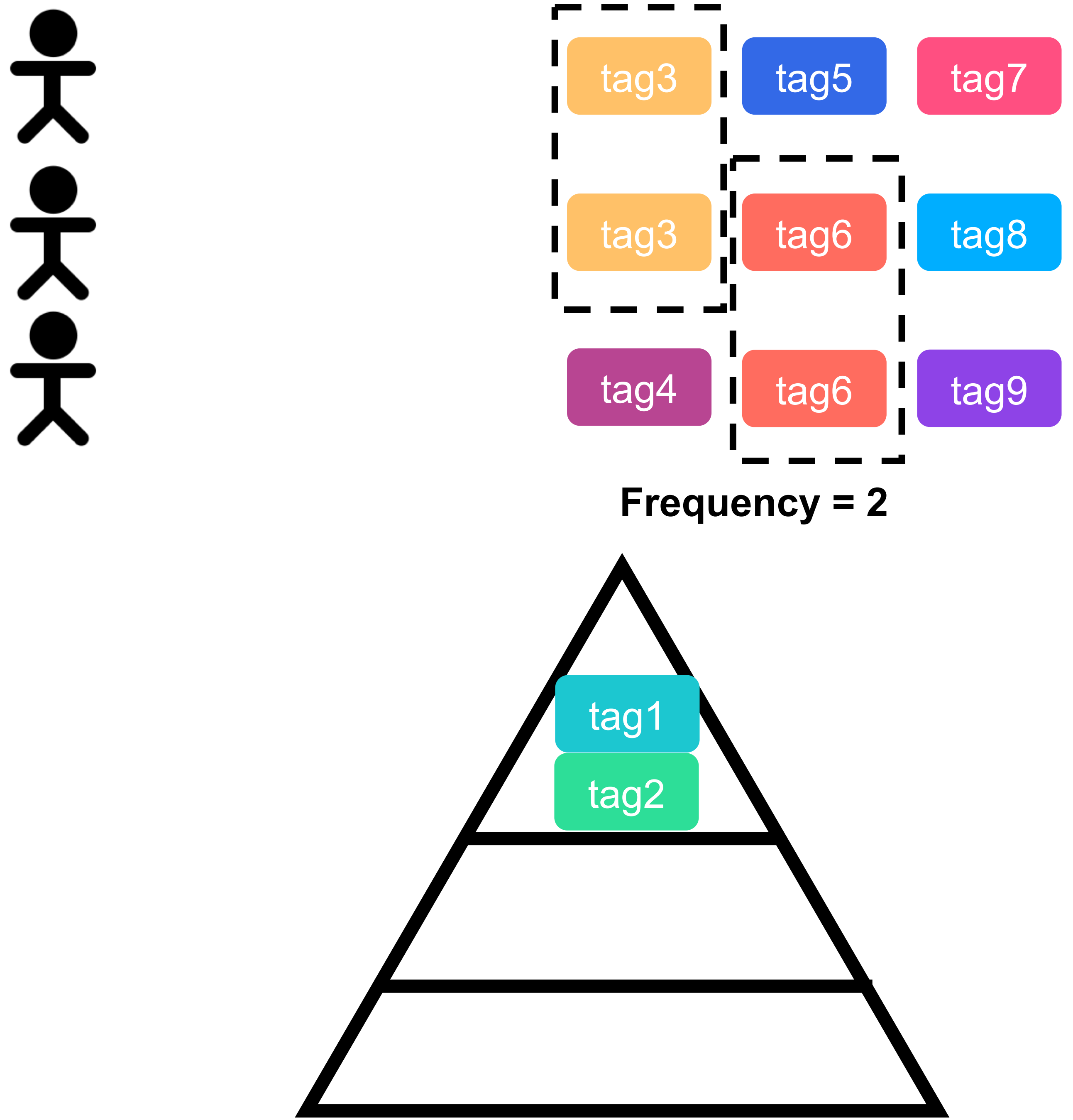
Pyramid Method

- Summarization assessment methodology
- Relies on **multiple annotations**
- Provides a reliable assessment of content selection quality in tagging
- Measures the **informativeness** a set of tags are based on the content of human annotations

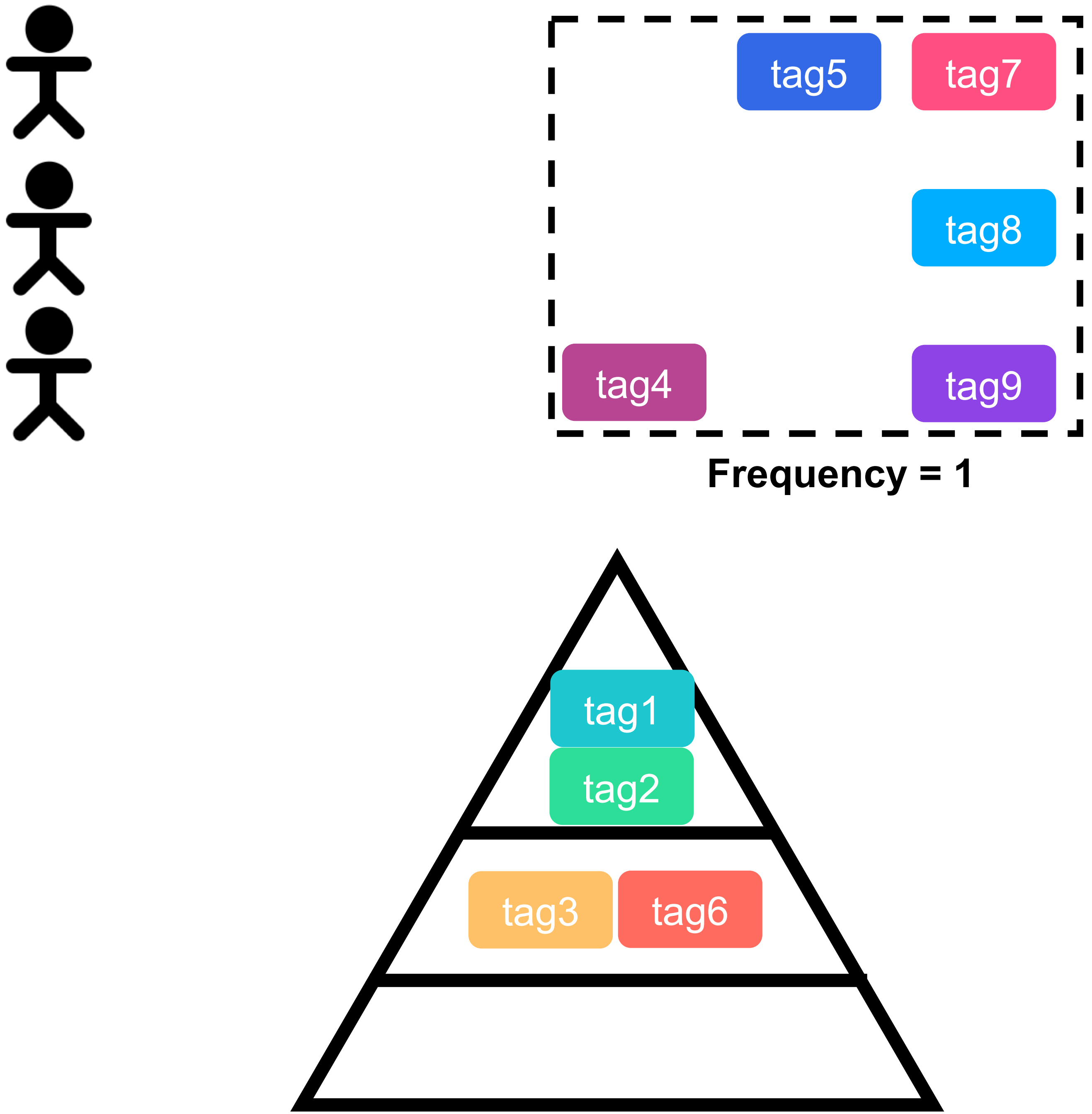

tag1
tag2
tag3
tag5
tag7
tag1
tag2
tag3
tag6
tag8
tag1
tag2
tag4
tag6
tag9

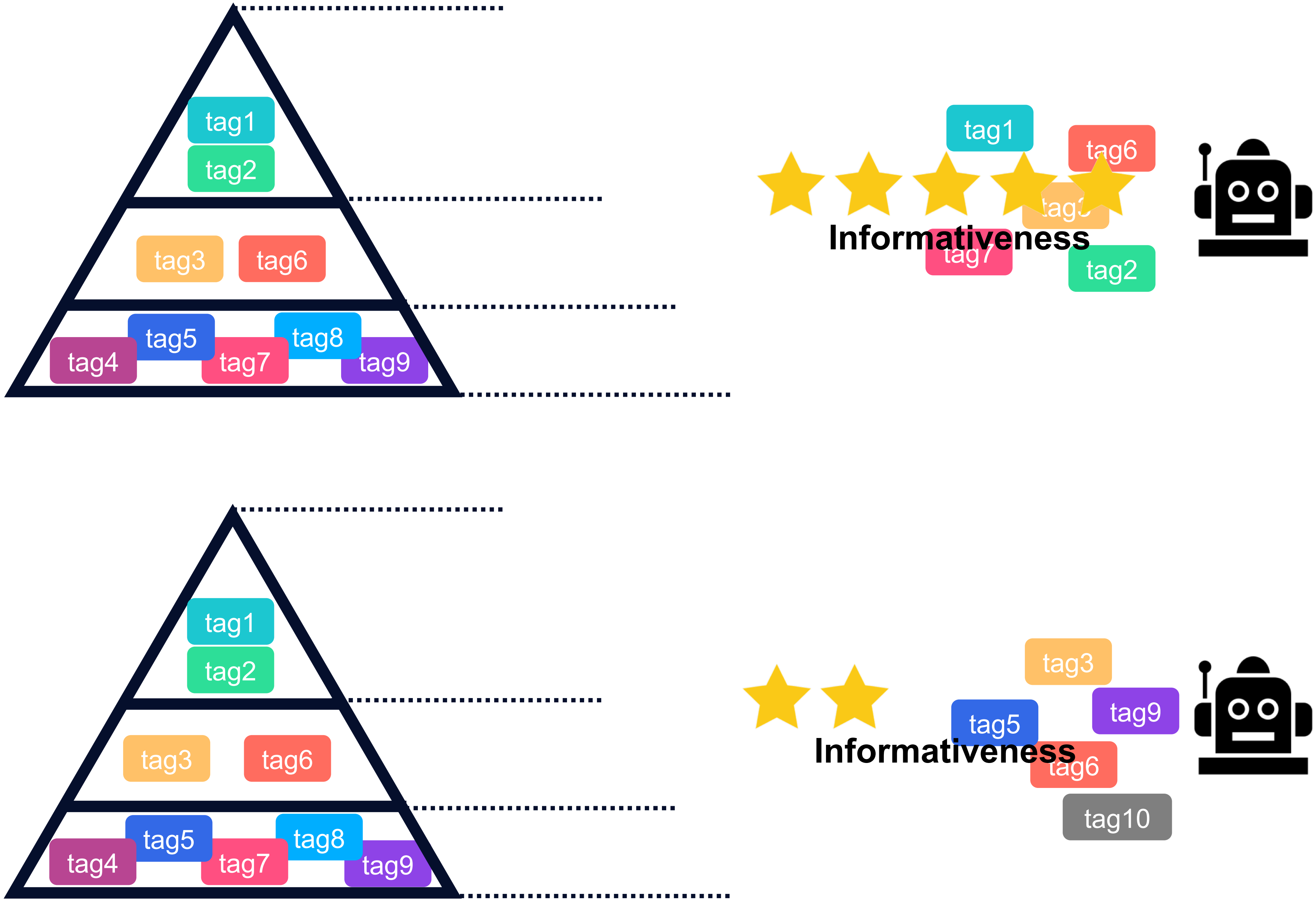



tag1
tag2
tag3
tag5
tag7
tag1
tag2
tag3
tag6
tag8
tag1
tag2
tag4
tag6
tag9
Frequency = 3



tag3
tag5
tag7
tag3
tag6
tag8
tag4
tag6
tag9
Frequency = 2
tag1
tag2


tag5
tag7
tag8
tag4
tag9
Frequency = 1
tag1
tag2
tag3
tag6


tag1
tag2
tag3
tag6
tag5
tag8
tag4
tag7
tag9
tag1
tag6
Informativeness
tag7
tag2
tag1
tag2
tag3
tag6
tag5
tag8
tag4
tag7
tag9
tag3
tag9
tag5
Informativeness
tag6
tag10

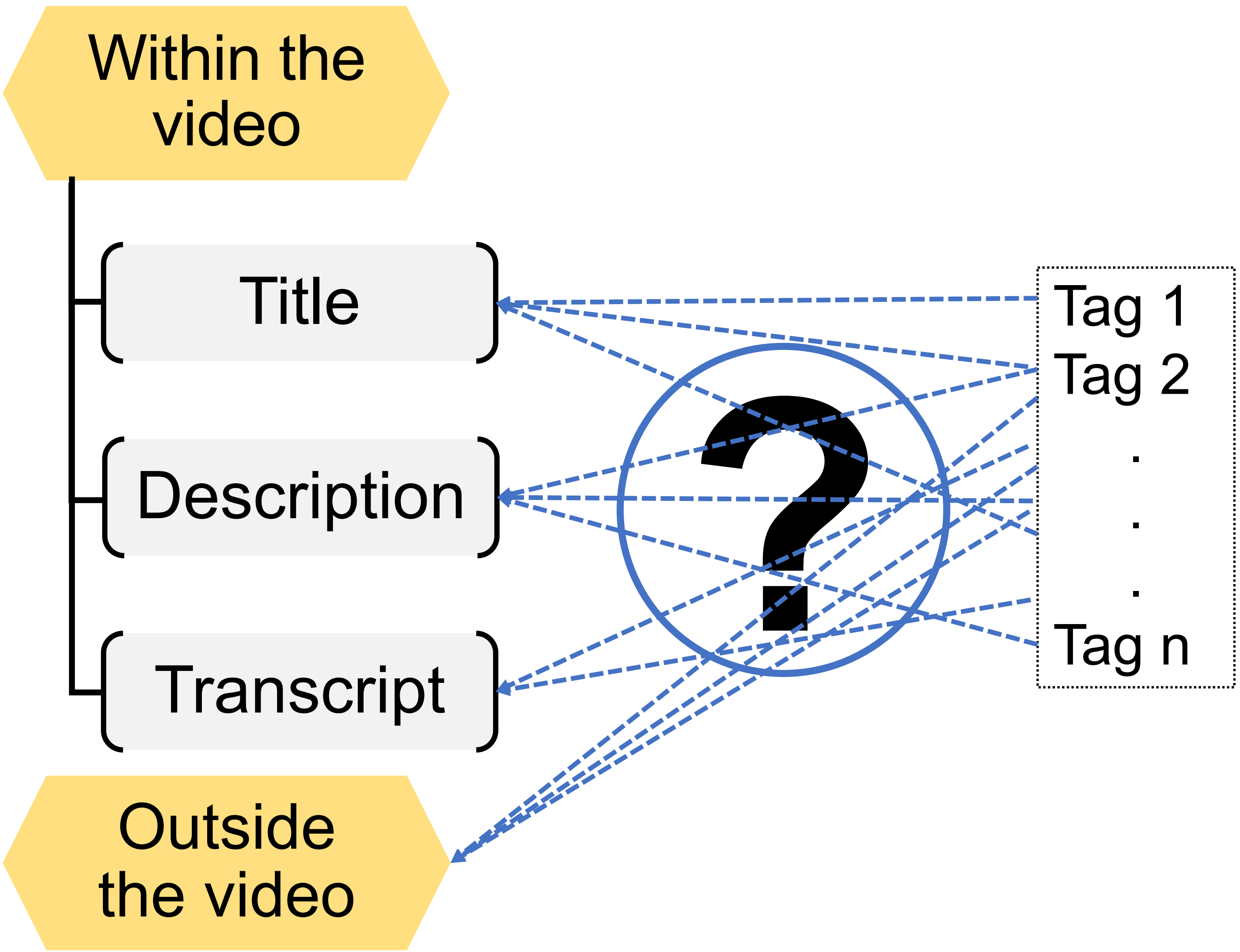
Results RQ1.1 – Best Approach

- All the configurations of BM25F outperform the other tagging approaches
- Configurations of BM25F that **leverage the transcript** achieve the better performance
- The most informative tags are recommended **within the first five tags**
- The tags produced by Google Video Intelligence are the least informative
- TagMulRec performs better than other abstractive approaches

Results RQ1.2 - Preprocessing

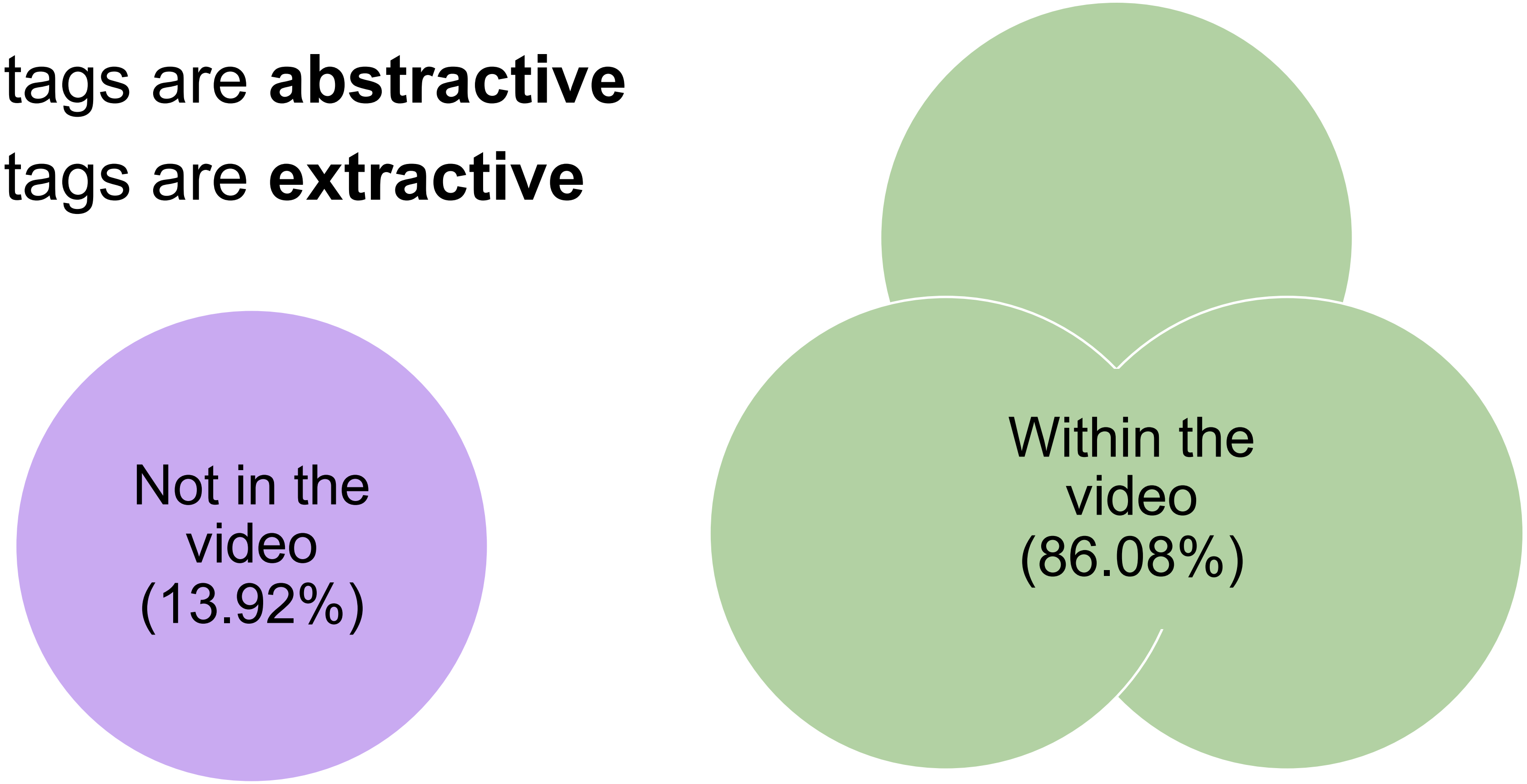
- Both **stemming** and **singularization** improve the performance of the tagging approaches
- **Singularization** leads to a higher performance improvement than **stemming**
- The best tagging results are achieved by **using BM25F(4,2,1) with singularization**

RQ2. Provenance: Where do relevant tags come from?



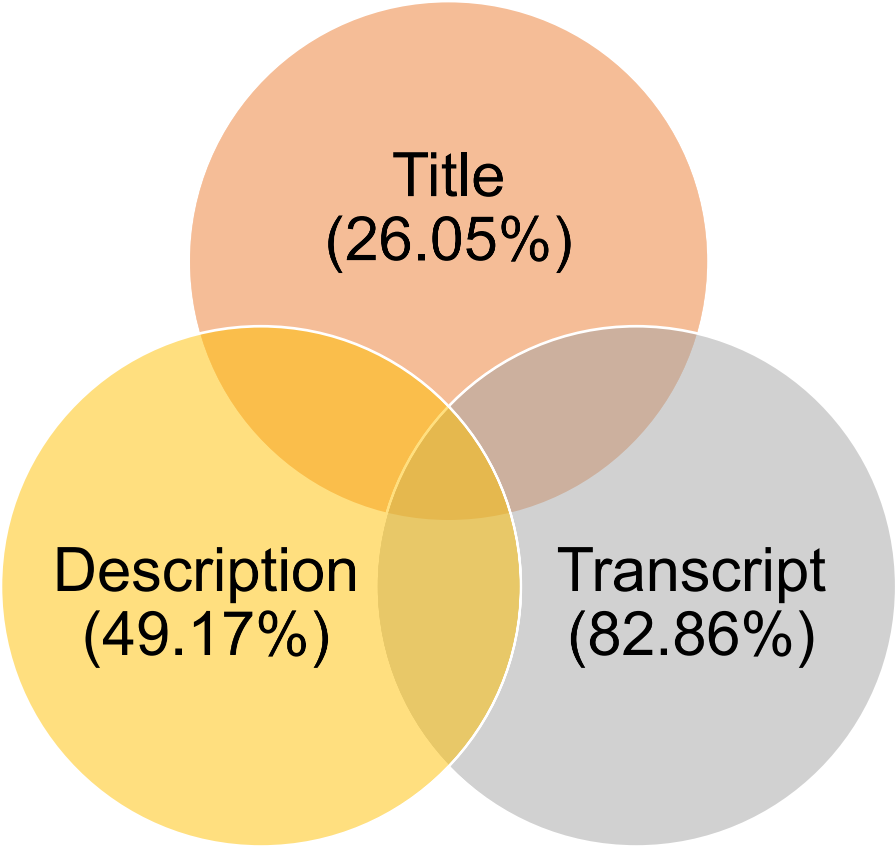
Provenance: Tags Recommended by Developers

- 13.92% of the tags are **abstractive**
- 86.08% of the tags are **extractive**



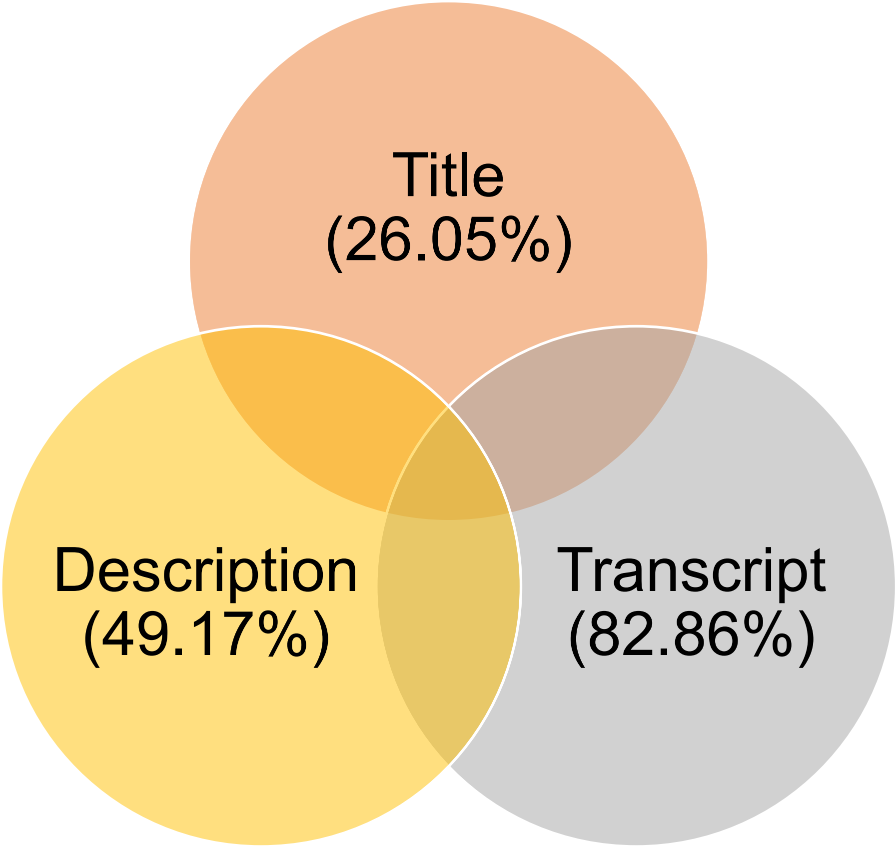
Provenance: Tags Recommended by Developers

- 82.86% of the tags are present in the **transcript** of the video
- 49.17% of the tags are present in the **description** of the video
- 26.05% of the tags are present in the **title** of the video



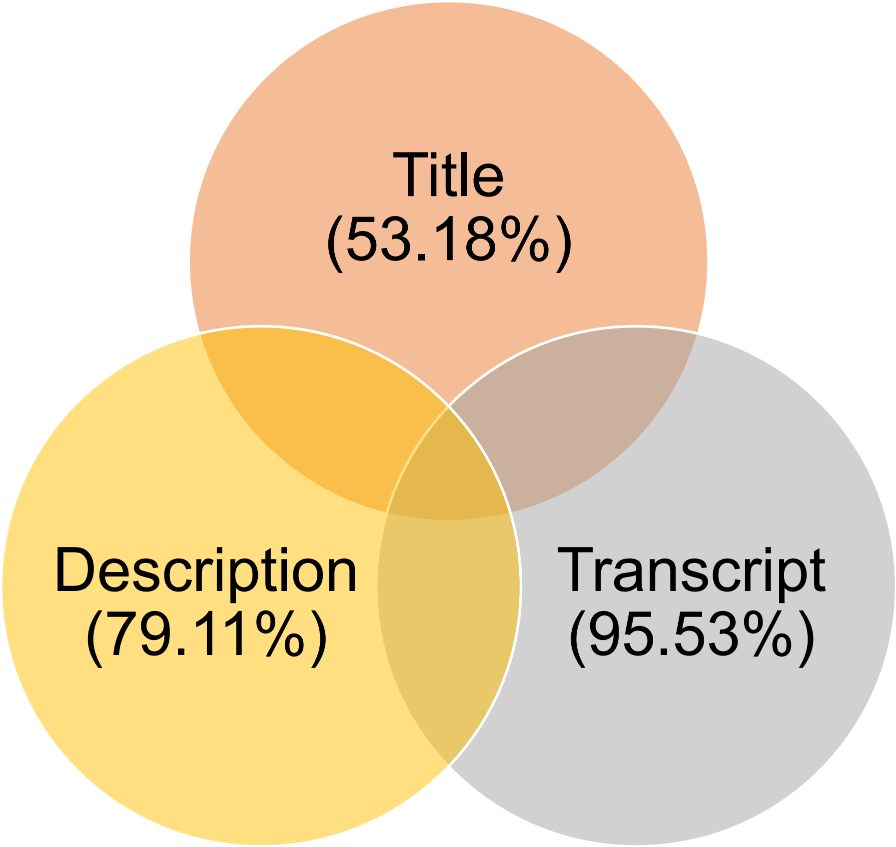
Provenance: Tags Recommended by Developers

- 82.86% of the tags are present in the **transcript** of the video
- 49.17% of the tags are present in the **description** of the video
- 26.05% of the tags are present in the **title** of the video



Provenance: Relevant Tags by BM25F(4,2,1)

- 95.53% of the tags are present in the **transcript** of the video
- 79.11% of the tags are present in the **description** of the video
- 53.18% of the tags are present in the **title** of the video



Conclusions

- Developers use **extractive** tags more than abstractive tags to describe software development videos
- Tags extracted using **BM25F** are more informative than tags recommended by other approaches
- The most informative tags are obtained **within the first five recommendations**
- The **transcript** of the video is the major contributor towards the extraction of informative tags

User Comment Classification for Software Development Videos Using Neural Networks

SERENE

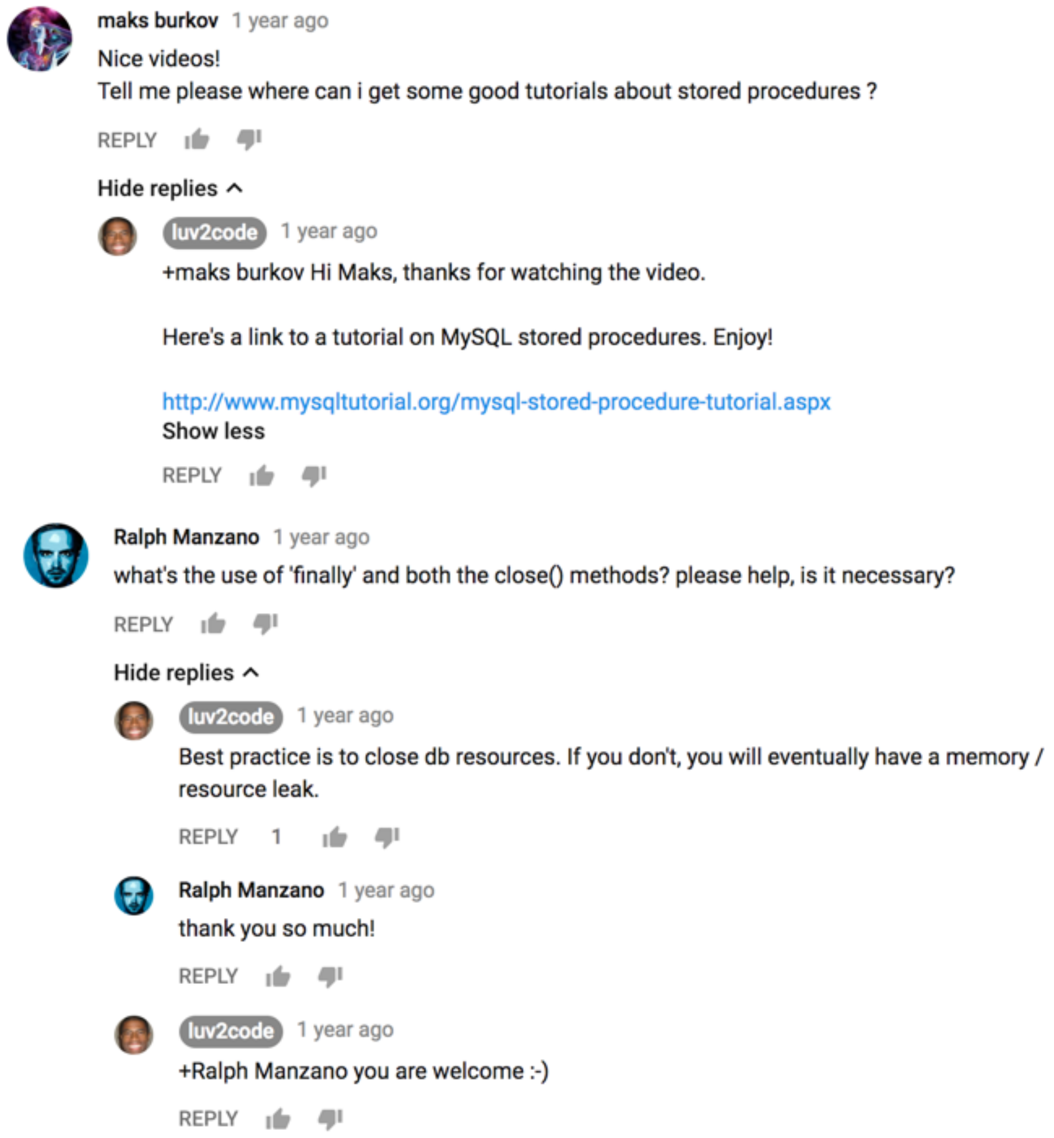
Esteban Parra, Javier Escobar-Avila, Jordan Snow, Jordan Ott, Sonia Haiduc, Shayok Chakraborty, and Erik Linstead

Florida State University & Chapman University

Use Comments

Often contain:

- Links to additional material
- Clarifications about the discussed concepts
- Solutions to errors that may occur in the tutorial
- Feedback



User comments are noisy

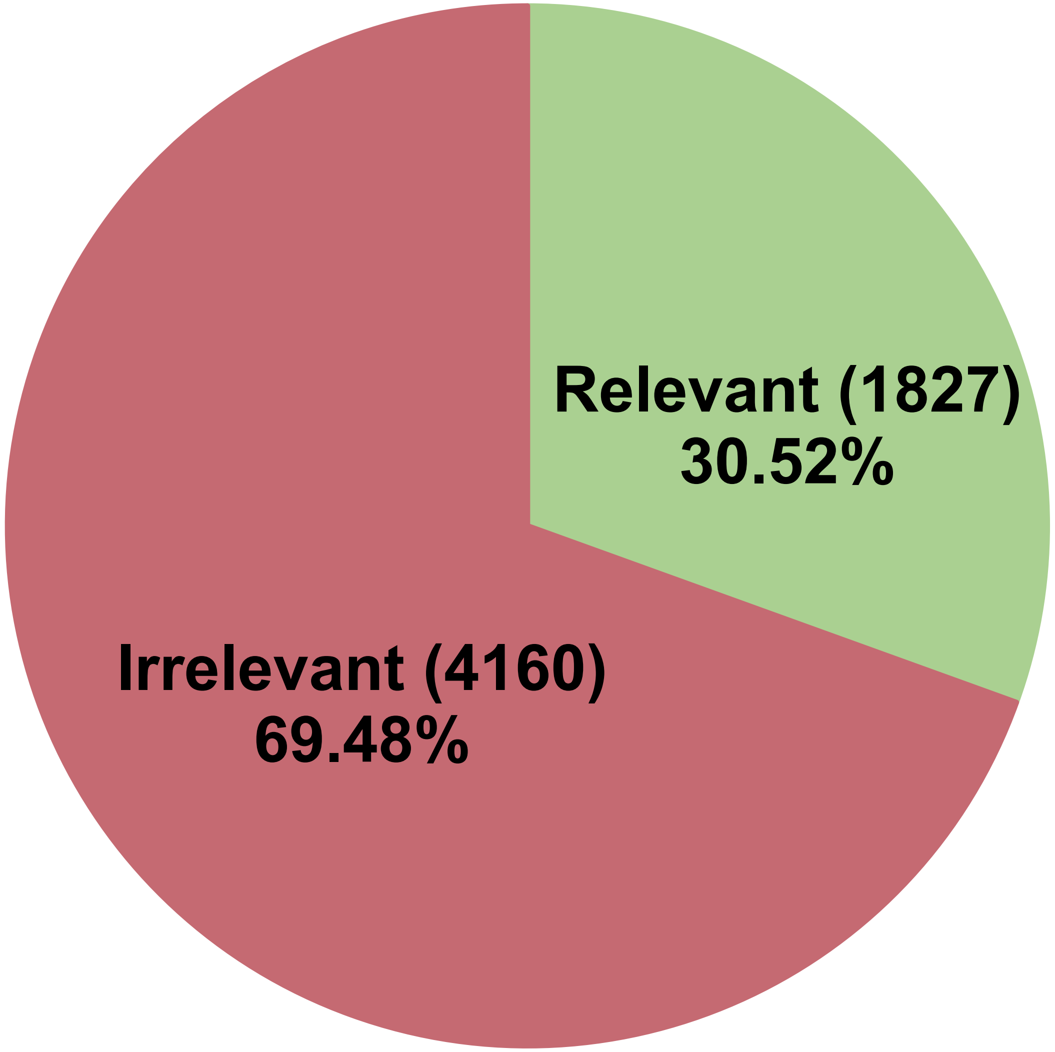
They contain a large number of irrelevant comments:

- Spam
- Insults
- Discussing unrelated topics

We aim to automatically identify the relevant comments

Dataset

- 12 popular software development tutorials
- 6000 comments
- Created by Poche et al.



Poche et. al. (2017) Analyzing User Comments on YouTube Coding Tutorial Videos. In: Proceedings of the 25th IEEE International Conference on Program Comprehension (ICPC'17), IEEE, Austin, TX, USA, pp 196{206

Automatic Classification Approaches

Traditional Classifiers	Neural Networks
• SVM • Naïve Bayes	• Feed Forward • RNN • CNN

Traditional Classifiers

SVM

- Separates two classes of samples by the maximal margin, in a high dimensional feature space

Naïve Bayes

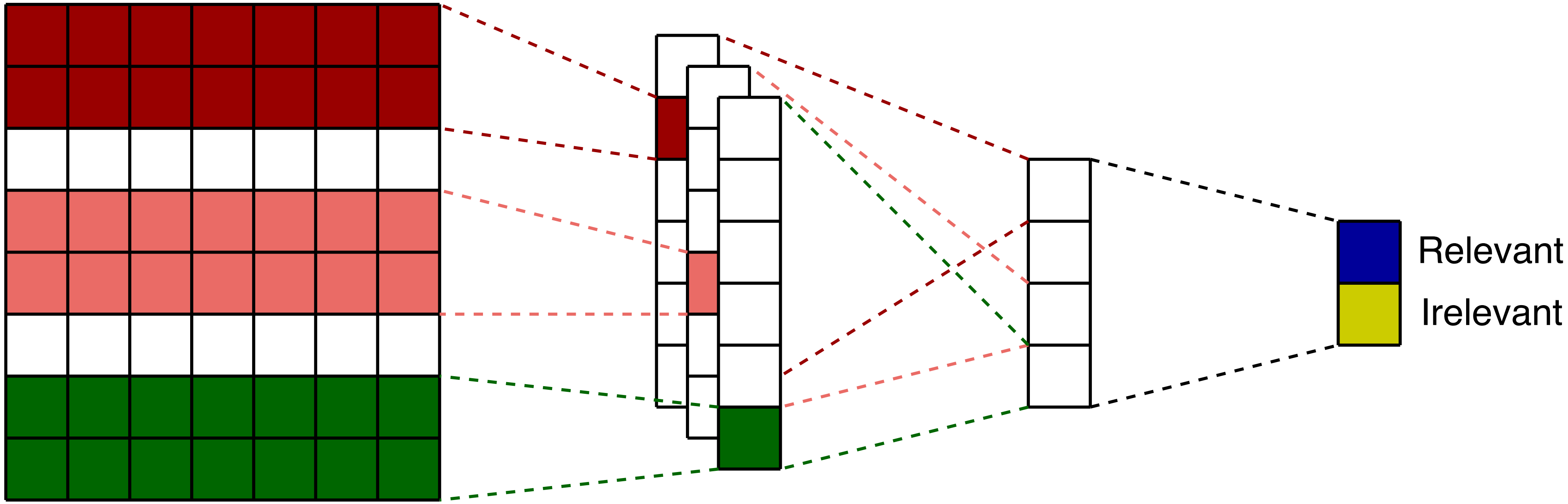
- Linear probabilistic classier that uses Bayes’ theorem to identify strong dependencies between features

Feature Schemes

Bag-of-Words (BoW)	Linguistic & Semantic (L&S)

• The features are the individual words of the text
• Represented by a word-document matrix
• The normalized term frequency of each word in each document
	• Measurable characteristics that distinguish irrelevant comments
	  • Number of white spaces
	  • Word duplication
	  • Number of stopwords
	  • Word similarity

Convolutional Neural Network (CNN)



1. Word embedding layer: Embeds words into low-dimensional vectors

2. Convolution layer with multiple filters and variable filter sizes

3. Max-over-time pooling layer

4. Fully connected layer with dropout and softmax output

Other NN architectures

Feed Forward (FF)	Recurrent Neural Network (RNN)
Composed of: • A trainable embedding layer • Five fully connected layers with dropout in between • A single fully connected output layer	• LSTM architecture Composed of: • A trainable embedding layer • Two LSTM layers • Two fully connected layers with dropout in between • A single fully connected output layer

Research Questions

RQ1 - Do L&S features provide a better classification than BoW features when using SVM and NB?

RQ2 - How do deep learning approaches perform compared to the classification approaches used by the state of the art?

Evaluation metrics

Capacity to correctly identify relevant comments

$$\boldsymbol{Precision} = \frac{correctly\ classified\ as\ relevant}{Total\ classified\ as\ relevant}$$

Ratio of correctly classified relevant comments from all the existing relevant comments

$$\boldsymbol{Recall} = \frac{Comments\ classified\ as\ relevant}{Total\ relevant\ comments\ in\ the\ dataset}$$

Effectiveness of a classifier as a combination of precision and recall

$$\mathbf{F}\boldsymbol{Score} = 2 \cdot \frac{(Precision\ \cdot Recall)}{(Precision + Recall)}$$

Replication State-of-the-art

- We achieved the best performance by NB when using the unprocessed text, with an F-score of 0.78.
- Traditional classifiers are negatively impacted by the use of stopword removal and stemming.

Classifiers	Poché et al (2017)			Our replication		
	Precision	Recall	F-Score	Precision	Recall	F-Score
SVM(Unprocessed)	0.79	0.75	0.77	0.89	0.29	0.43
SVM(Stem)	0.77	0.73	0.75	0.83	0.19	0.31
SVM(Stop&Stem)	0.75	0.65	0.70	0.81	0.31	0.45
NB(Unprocessed)	0.63	0.71	0.67	0.75	0.82	0.78
NB(Stem)	0.62	0.76	0.68	0.75	0.72	0.74
NB(Stop&Stem)	0.62	0.73	0.67	0.74	0.71	0.72

Results RQ1 – Feature Schemes

The classifiers using linguistic and semantic features underperform classifiers that use BoW features on software development video comments

Approach	Precision	Recall	FScore
SVM (BoW)	0.89	0.29	**0.43**
NB (BoW)	0.75	0.82	**0.78**
SVM (L&S)	0.61	0.28	0.38
NB (L&S)	0.49	0.56	0.52

Results RQ2 – Neural Networks

The neural network architectures significantly outperform the state of the art classifiers across all the performance metrics

Approach	Precision	Recall	FScore
NB (BoW)	0.75	0.82	0.78
FF	0.88	0.89	**0.88**
LSTM	0.88	0.87	**0.88**
CNN	0.87	0.89	**0.88**

Conclusions

- Linguistic and semantic features do not perform well on comments for software development videos.
- All the Neural Network outperform traditional classifiers in terms of precision, recall and F-score.