Content Categorization *of* API Discussions

Daqing Hou, Lingfeng Mo

Clarkson University
Potsdam NY, USA

Motivation

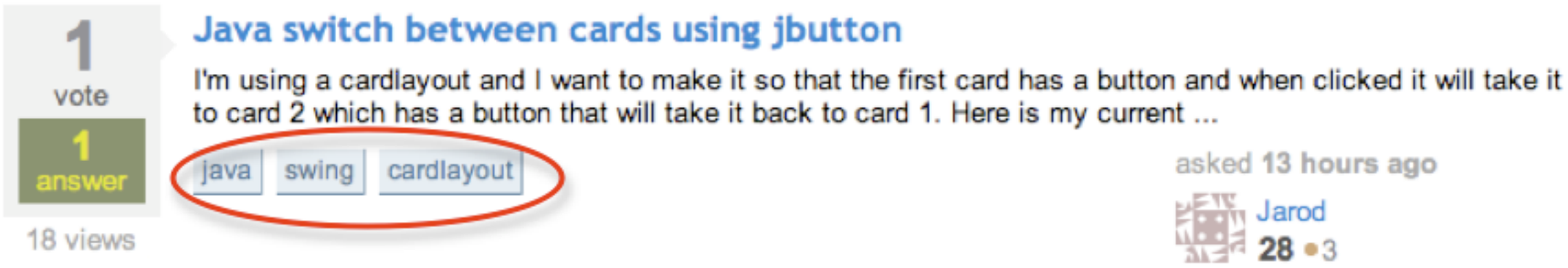
- APIs are fundamental building blocks of modern Software Engineering
  - *e.g. SDK, GUI, Web, DB, ...*
- Online forums important for use of APIs
  - *Every popular API has a dedicated forum*
  - *plus general Q&A sites such as Stack Overflow*
- Better management needed to help unleash full potential of large volumes of forum data

Motivation

- Text classification is critical for organizing large data, making their search and browsing more efficient, but

- Many forums do NOT classify discussions

- Stack Overflow relies on *manual* labeling



Motivation

- Text classification is critical for organizing large data, making their search and browsing more efficient, but

- Many forums do NOT classify discussions
- Stack Overflow relies on *manual* labeling
  - *time consuming*
  - *prone to user errors*
  - *with multiple labels, not always clear what the* central topic *of discussion is*

Research Questions

- Study designed to explore categorization of central topic of API discussion
  - **RQ1** *How well will existing learning algorithms such as Naive Bayes work for categorizing API forum discussions?*
  - **RQ2** *What are the major factors that impact classification accuracy?*

Study Procedure

- Data collection
  - *Java Swing; done iteratively, 3 datasets of increasing size*
  - *required "deep" analysis of content of API discussions*
- Data preprocessing & feature selection
  - *stop words removal*
  - *identifier splitting*
  - *code removal*
- Mainly Naive Bayes, compared with SVM in the end
  - *Mallet and SVM light*

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

10

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

17

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

8

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

TABLE I: Three versions of training data collected from Java Swing Forum and their breakdown by categories

Data Version	Categories/Labels	#Documents	Total
V1.0	border	4	46
	dispose	3	
	drawing	3	
	focus	3	
	layout	8	
	action	10	
	icon	5	
	rendererEditor	4	
	title	3	
	others	3	
V2.0	borderAndMargin	13	158
	dispose	9	
	drawing	8	
	focus	9	
	layout	9	
	action	8	
	loadingIcons	8	
	rendererEditor	13	
	titleBar	6	
	titleBarFont	10	
	textIconPosition	10	
	dynamicHierarchy	10	
	defaultButton	11	
	mouseMotionPosition	12	
	threading	8	
	OOP	7	
	social	7	
V3.0	borderAndMargin	82	833
	dispose	103	
	drawing	108	
	focus	104	
	layout	125	
	titleBar	106	
	textIconPosition	96	
	dynamicHierarchy	109	

All labels are major topics

Brief Review of NB

$$C = arg_{C_i} max P(C_i \mid D) \qquad (1)$$

Brief Review of NB

$$C = arg_{C_i} max P(C_i \mid D) \tag{1}$$

$$P(A \mid B) * P(B) = P(B \mid A) * P(A) \tag{2}$$

Brief Review of NB

$$C = arg_{C_i} max P(C_i \mid D) \tag{1}$$

$$P(A \mid B) * P(B) = P(B \mid A) * P(A) \tag{2}$$

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \tag{3}$$

Brief Review of NB

$$C = arg_{C_i} max P(C_i \mid D) \tag{1}$$

$$P(A \mid B) * P(B) = P(B \mid A) * P(A) \tag{2}$$

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \checkmark \tag{3}$$

Brief Review of NB

$$C = arg_{C_i} max P(C_i \mid D) \tag{1}$$

$$P(A \mid B) * P(B) = P(B \mid A) * P(A) \tag{2}$$

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \tag{3}$$

Brief Review of NB

$$C = arg_{C_i} max P(C_i \mid D) \tag{1}$$

$$P(A \mid B) * P(B) = P(B \mid A) * P(A) \tag{2}$$

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \tag{3}$$

Brief Review of NB

$$C = arg_{C_i} max P(C_i \mid D) \tag{1}$$

$$P(A \mid B) * P(B) = P(B \mid A) * P(A) \tag{2}$$

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \tag{3}$$

$$\bar{P}(D \mid C_i) = \prod_{1 \leq j \leq n} P(T_j \mid C_i)^{\#(T_j, D)} \tag{4}$$

Brief Review of NB

$$C = arg_{C_i} max P(C_i \mid D) \tag{1}$$

$$P(A \mid B) * P(B) = P(B \mid A) * P(A) \tag{2}$$

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \tag{3}$$

$$\bar{P}(D \mid C_i) = \prod_{1 \le j \le n} P(T_j \mid C_i)^{\#(T_j, D)} \tag{4}$$

Brief Review of NB

$$C = arg_{C_i} max P(C_i \mid D) \tag{1}$$

$$P(A \mid B) * P(B) = P(B \mid A) * P(A) \tag{2}$$

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \tag{3}$$

$$\bar{P}(D \mid C_i) = \prod_{1 \le j \le n} P(T_j \mid C_i)^{\#(T_j, D)} \tag{4}$$

Metrics

- 90%-10% split of data for training-testing
- 10,000 trials

- **Test Accuracy** for each trial (#correct / #classified)
- **Test Accuracy Mean** and **stdev** for all 10,000 trials

RQ1: NB works very well

Training Data	RAW	SWR	WS	SWR + WS
v1.0 - original	27.5, 18.1	37.0, 21.3	35.8, 20.0	46.3, 21.6
v1.0 - code	33.7, 20.5	34.9, 20.6	38.2, 20.3	41.7, 21.1
v1.0 - text	34.5, 20.3	**57.3, 21.3**	33.8, 20.3	56.3, 21.1
v2.0 - original	49.5, 12.3	60.4, 12.0	55.8, 12.3	**62.7, 11.7**
v2.0 - code	56.2, 11.8	57.1, 12.0	61.9, 11.7	61.4, 11.6
v2.0 - text	60.0, 12.6	62.2, 11.7	62.1, 11.6	62.2, 11.6
v3.0 - original	92.1, 3.0	**93.6, 2.6**	91.3, 3.0	91.8, 3.0
v3.0 - code	91.5, 3.0	92.1, 2.9	89.7, 3.2	89.6, 3.2
v3.0 - text	92.1, 2.9	93.0, 2.8	92.0. 3.0	92.7, 2.8

RQ2: Increase in accuracy plateaus as training set gets sufficiently large

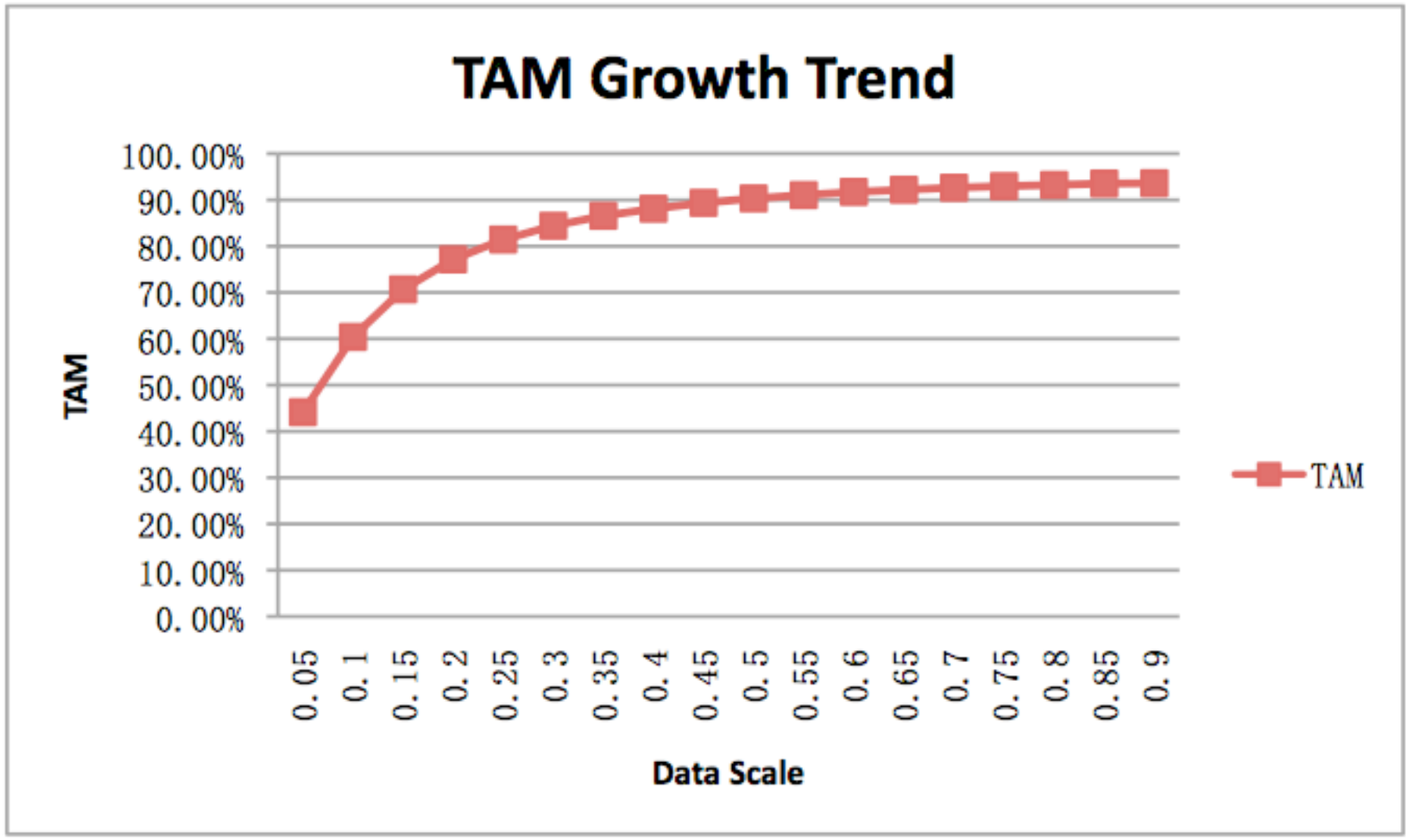


Fig. 2: Average Test Accuracy (TAM) as a function of training set size

RQ2: Increase in accuracy plateaus as training set gets sufficiently large

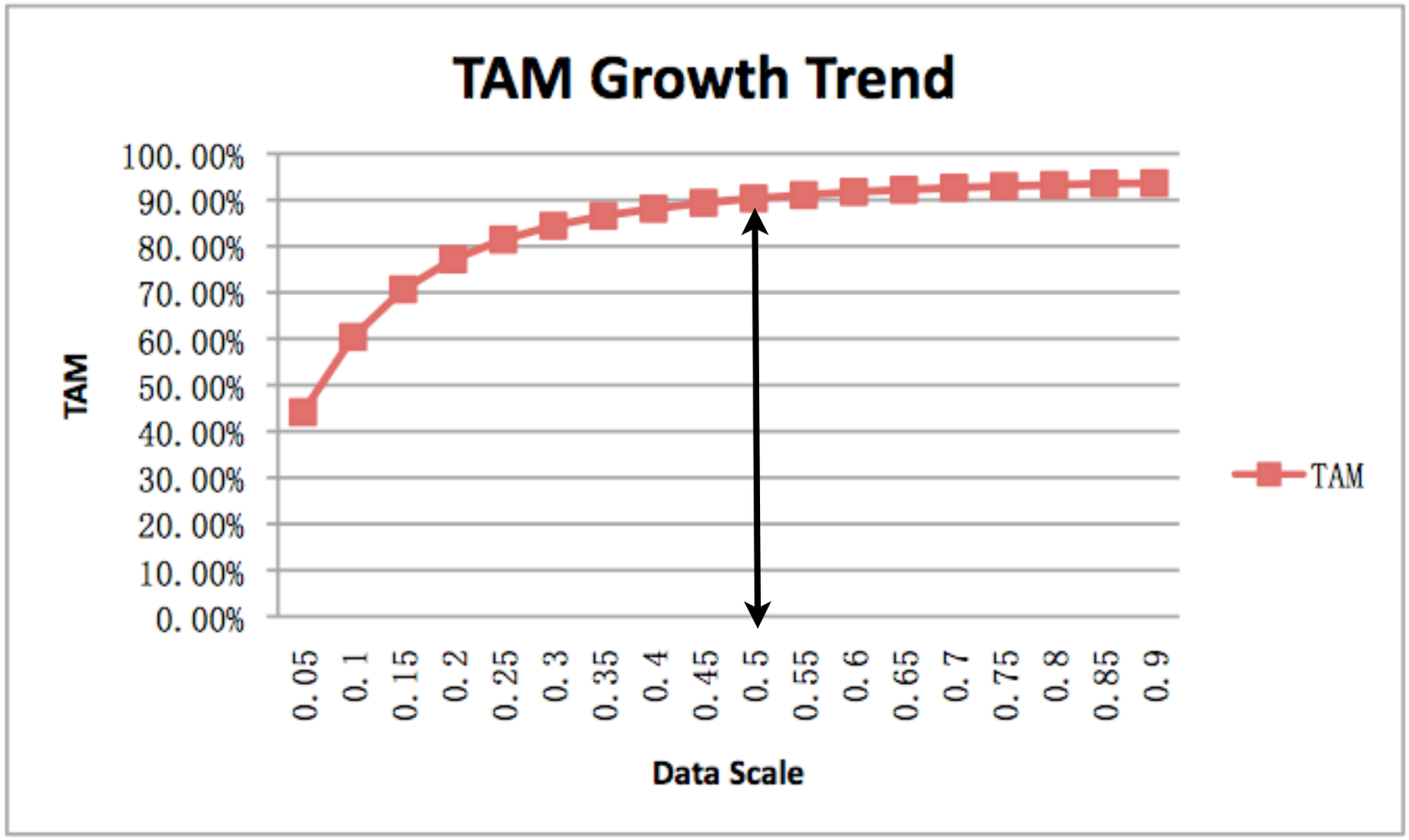


Fig. 2: Average Test Accuracy (TAM) as a function of training set size

RQ2: Increase in accuracy plateaus as training set gets sufficiently large

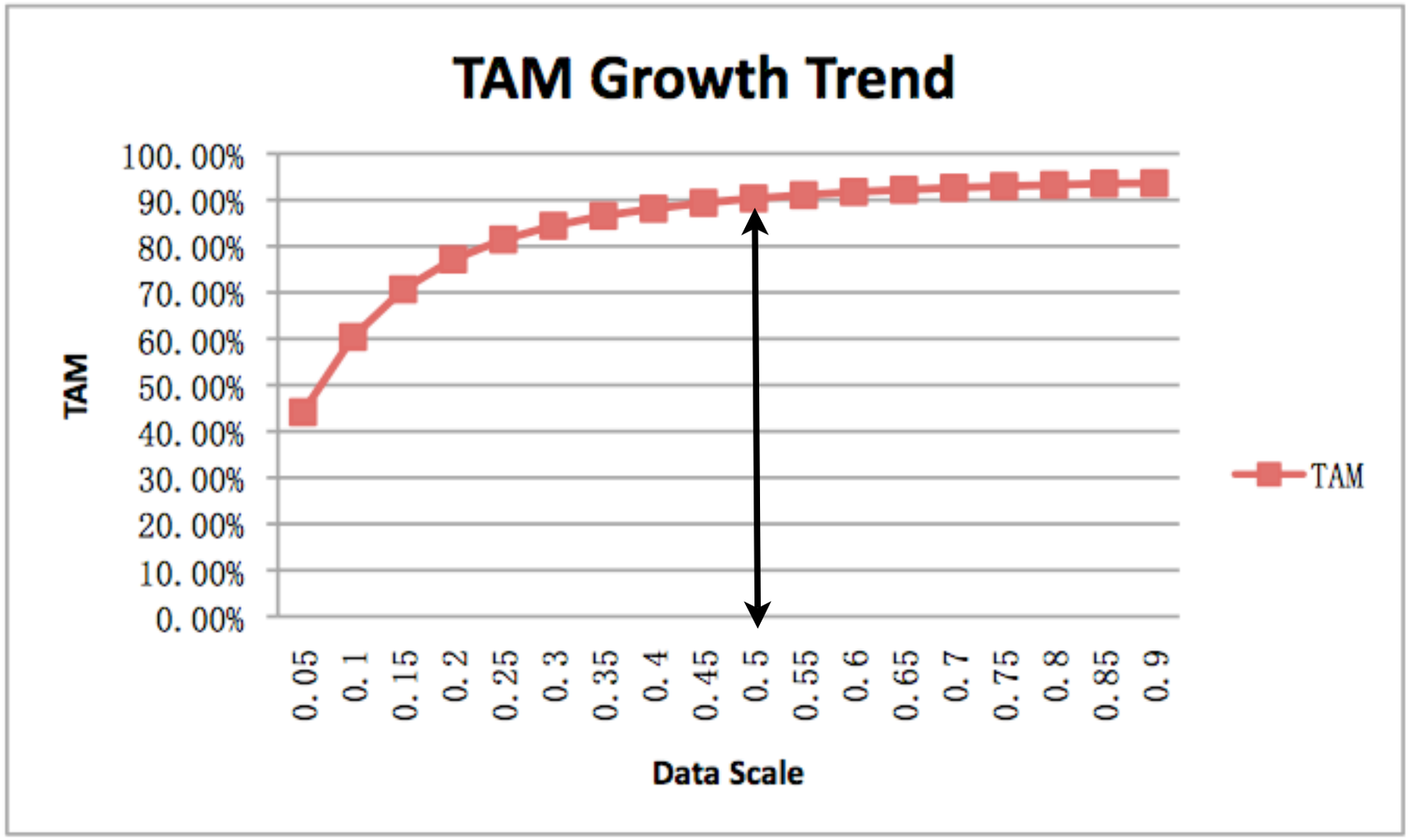


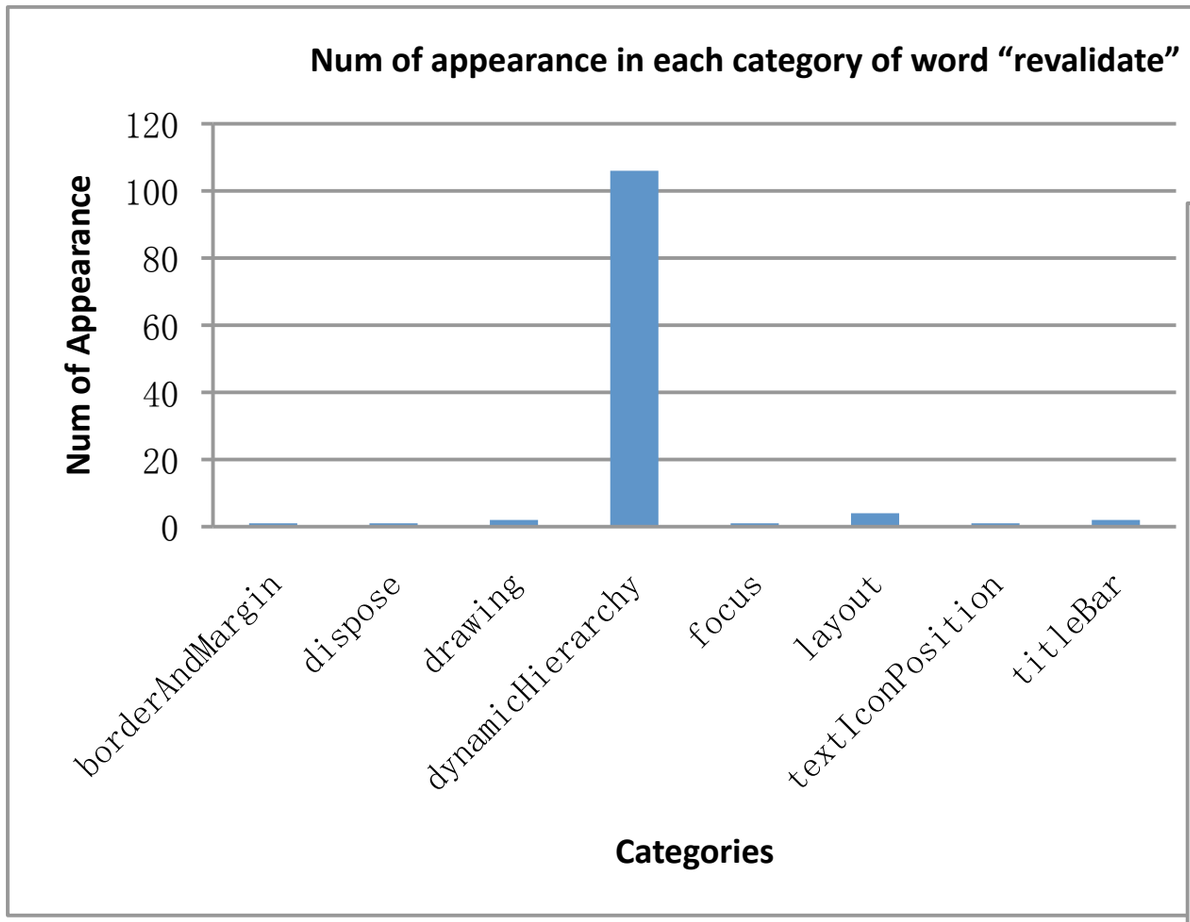
Fig. 2: Average Test Accuracy (TAM) as a function of training set size

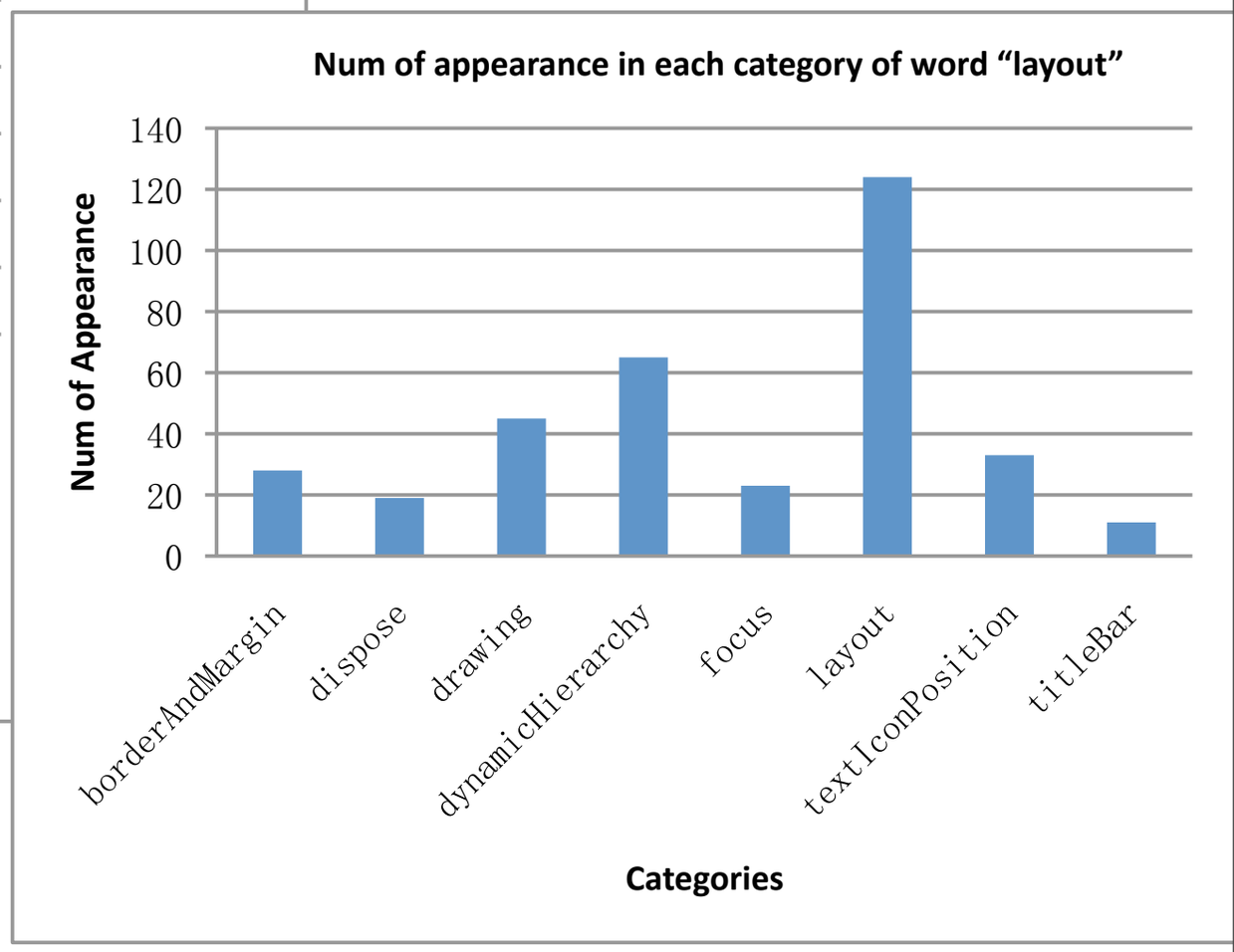
833 * 0.5 / 8 = 52 documents per category

RQ2: Multi-labels

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \quad (3)$$

$$\bar{P}(D \mid C_i) = \prod_{1 \le j \le n} P(T_j \mid C_i)^{\#(T_j, D)} \quad (4)$$

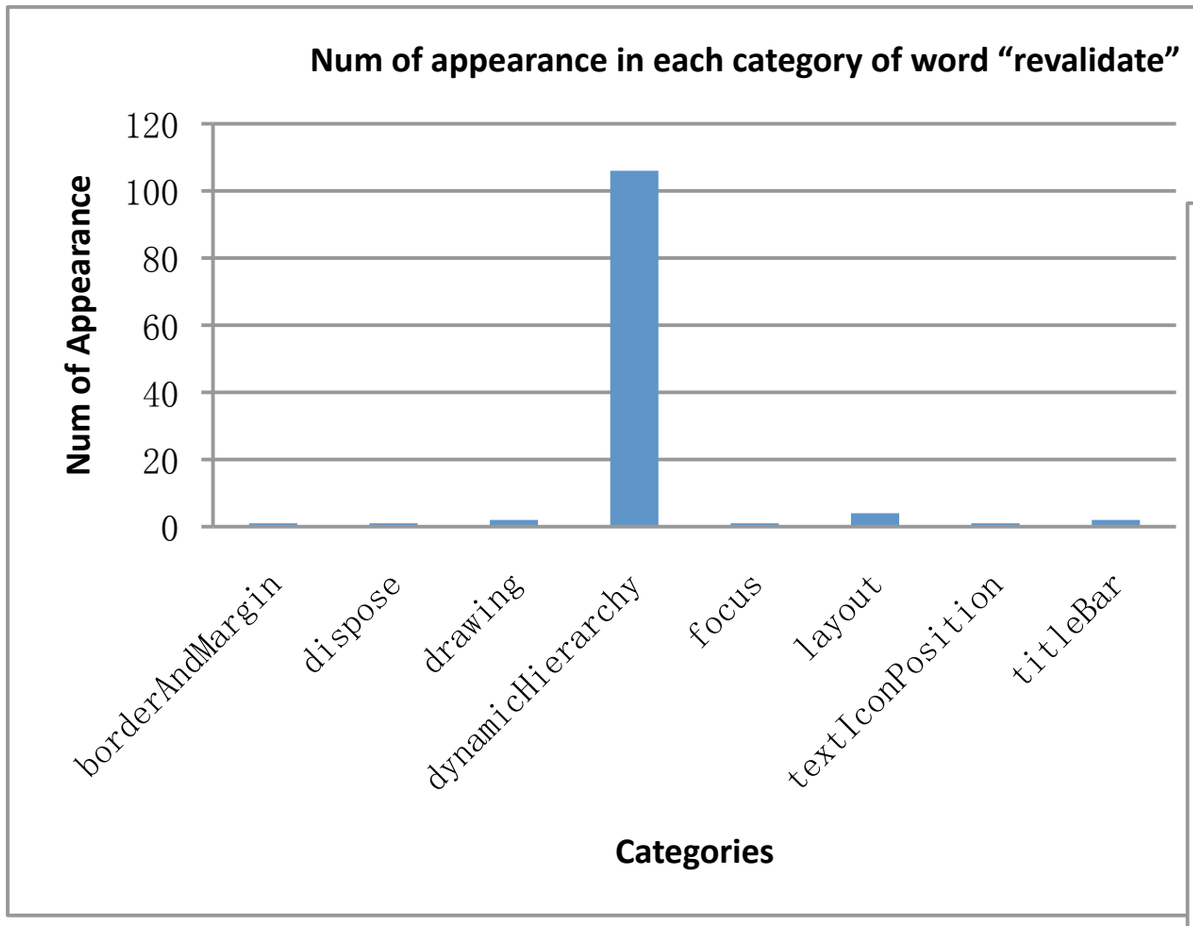


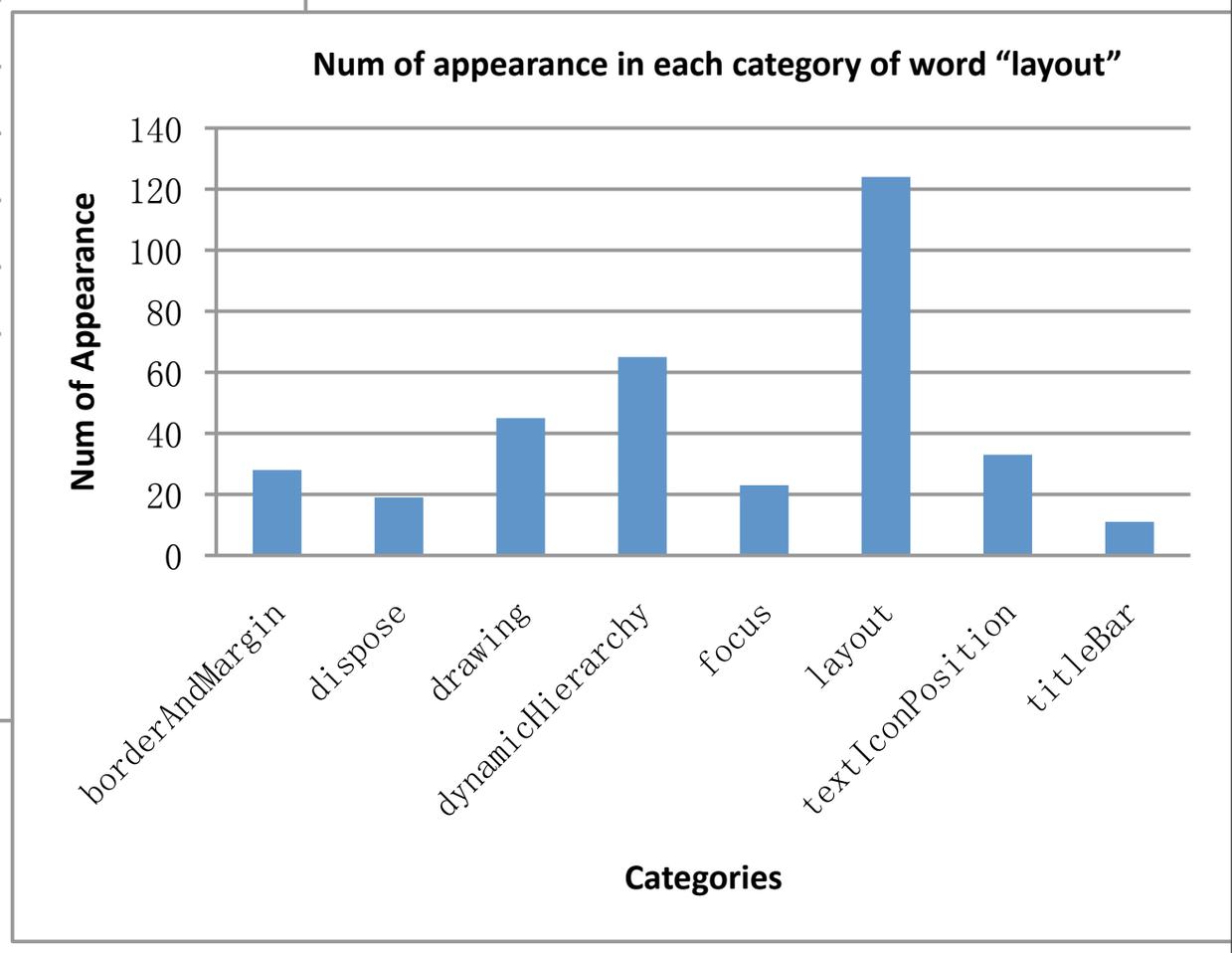


RQ2: Multi-labels

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \quad (3)$$

$$\bar{P}(D \mid C_i) = \prod_{1 \le j \le n} P(T_j \mid C_i)^{\#(T_j, D)} \quad (4)$$

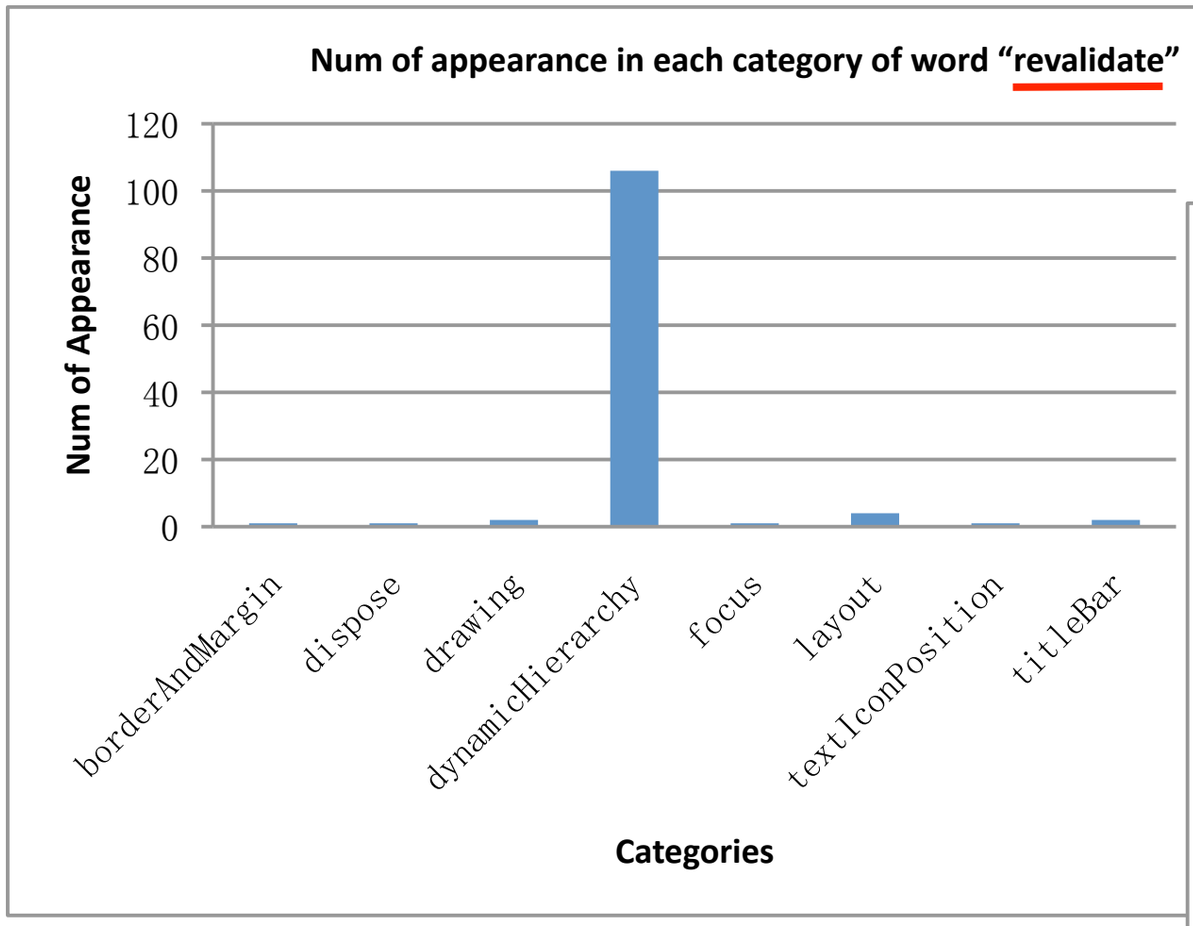


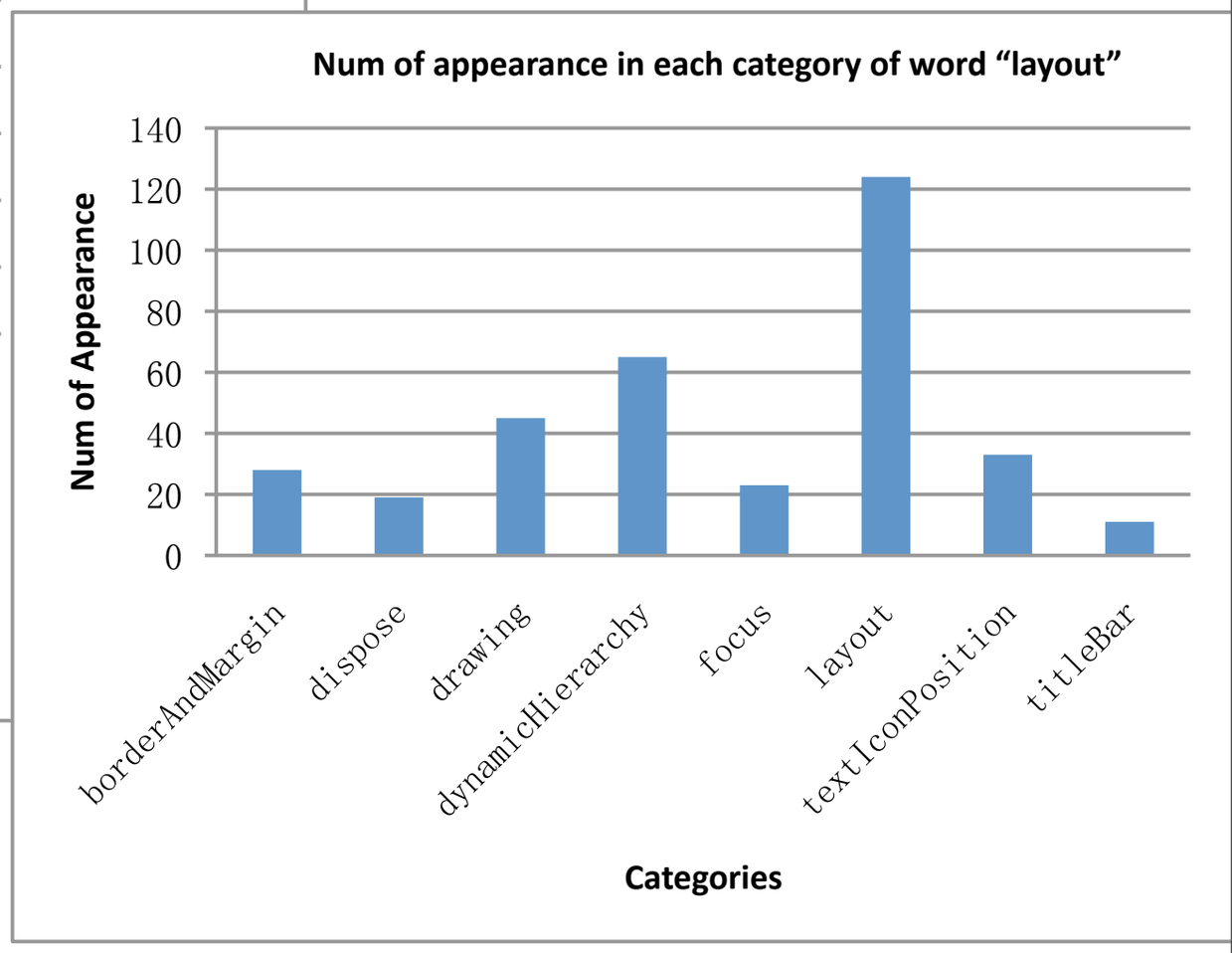


RQ2: Multi-labels

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \quad (3)$$

$$\bar{P}(D \mid C_i) = \prod_{1 \le j \le n} P(T_j \mid C_i)^{\#(T_j, D)} \quad (4)$$

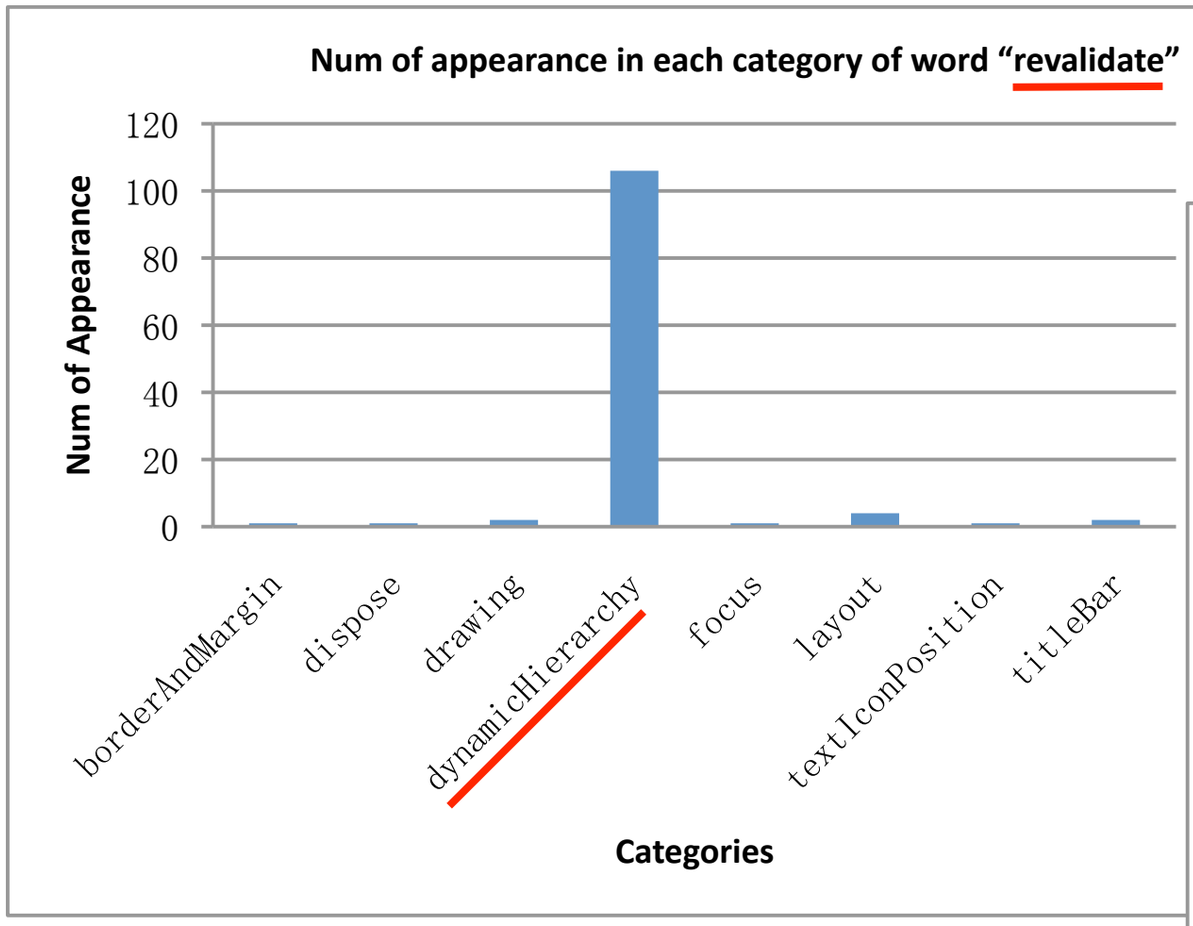


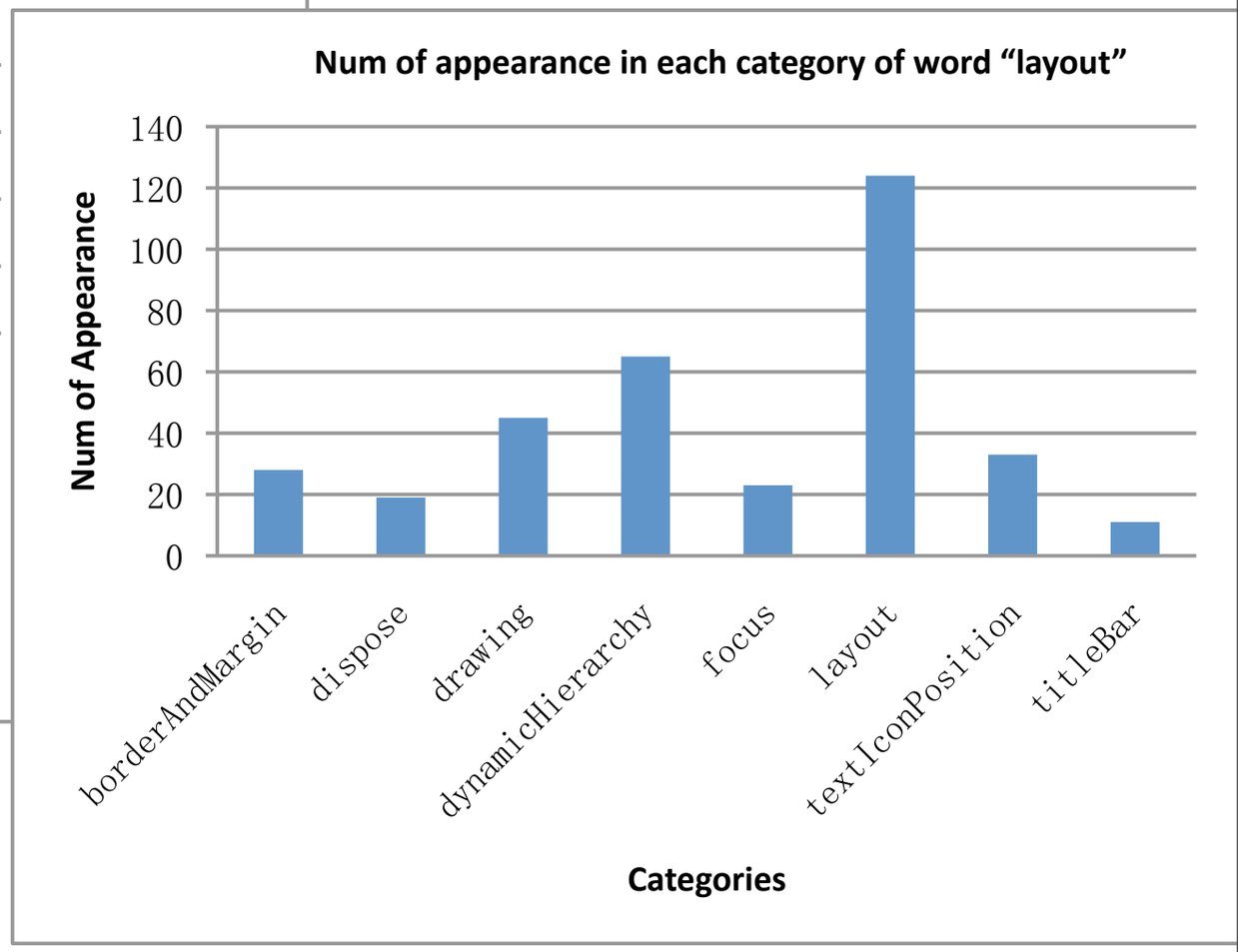


RQ2: Multi-labels

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \quad (3)$$

$$\bar{P}(D \mid C_i) = \prod_{1 \le j \le n} P(T_j \mid C_i)^{\#(T_j, D)} \quad (4)$$

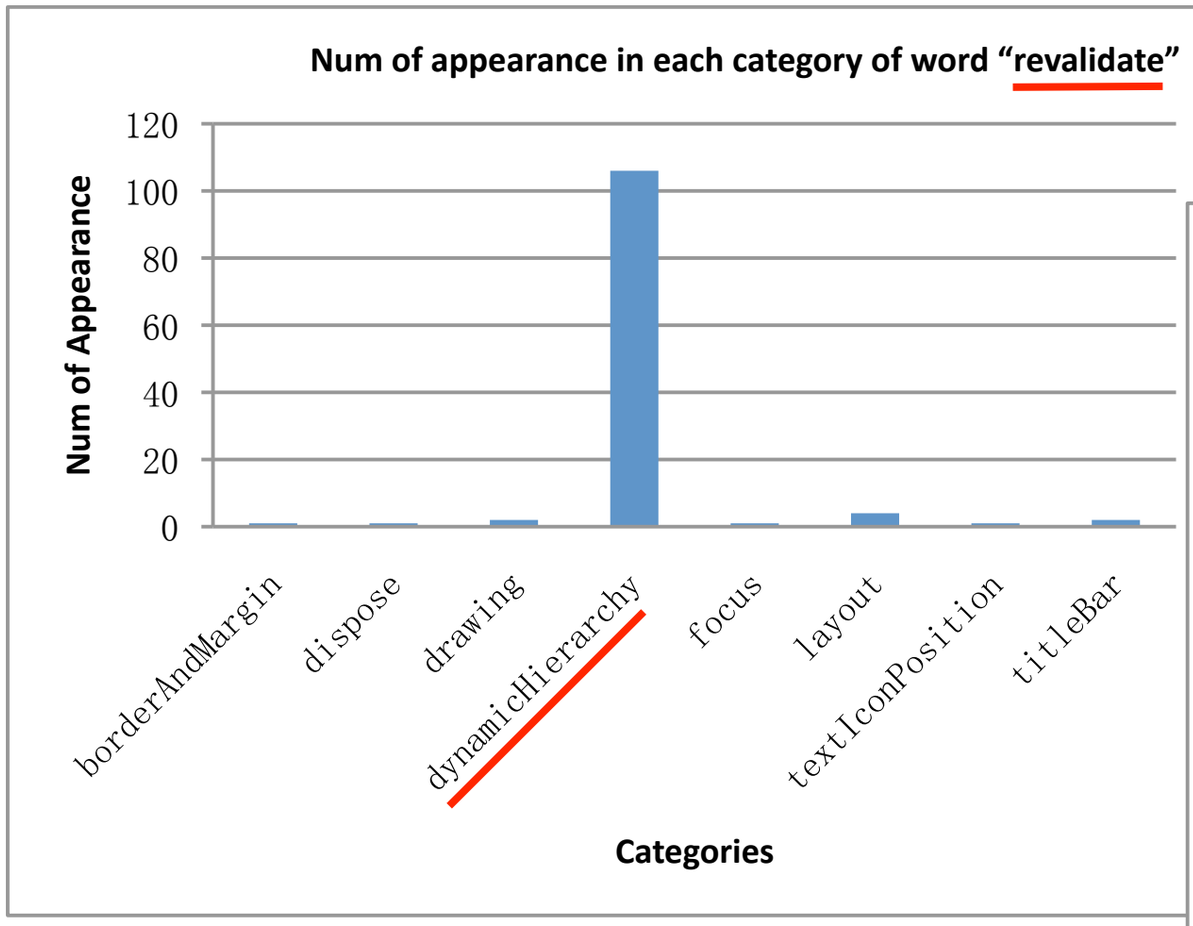


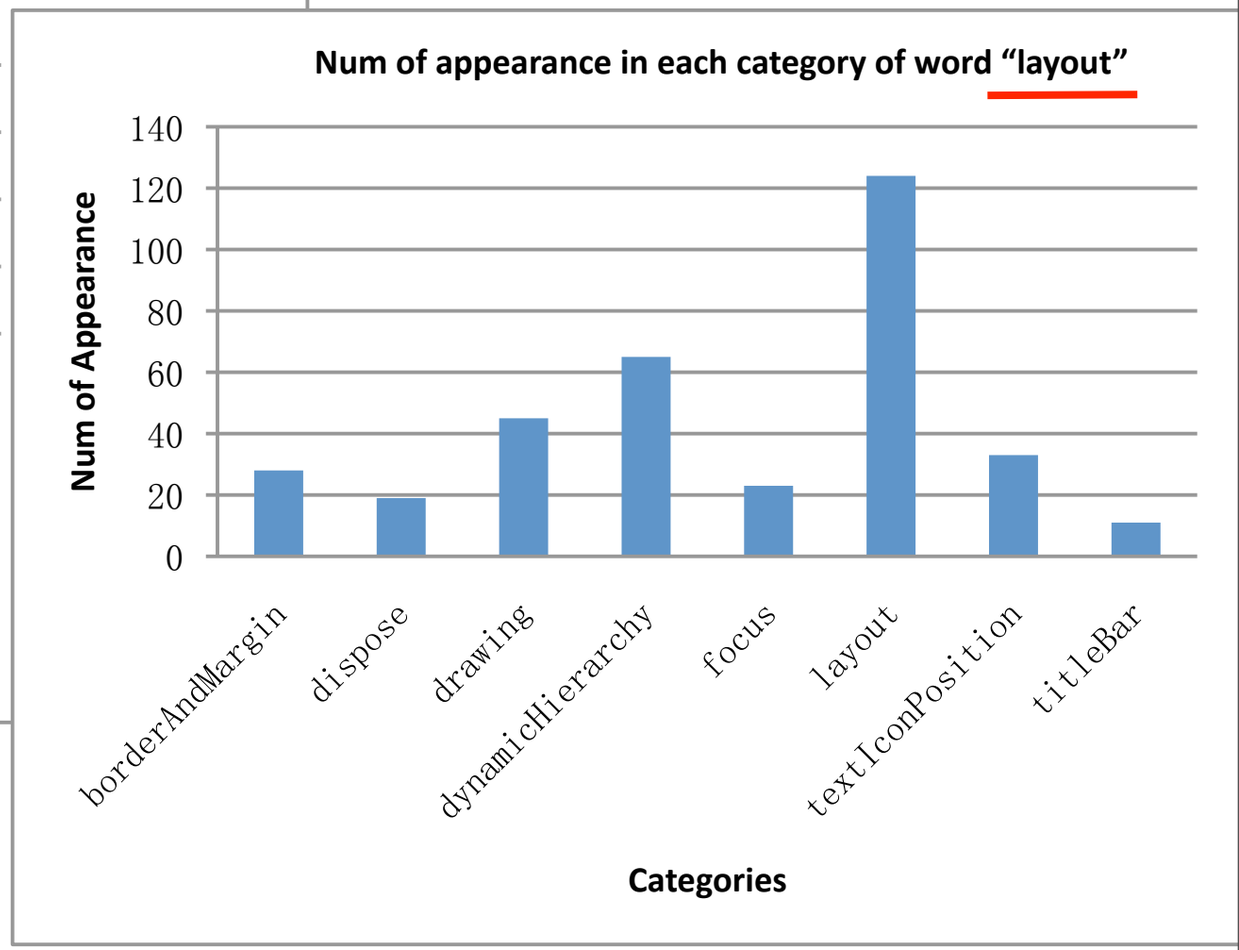


RQ2: Multi-labels

$$P(C_i \mid D) = \frac{P(D \mid C_i) * P(C_i)}{P(D)} \quad (3)$$

$$\bar{P}(D \mid C_i) = \prod_{1 \le j \le n} P(T_j \mid C_i)^{\#(T_j, D)} \quad (4)$$





Dealing with Multi-labels

TABLE III: Test Accuracy Mean (TAM) and the associated stddev for the N-label Naïve Bayes Classifier

#labels	TAM, stddev
N = 2	97.8, 1.6
N = 3	99.3, 0.9
N = 4	99.7, 0.5

Removing code (from layout category) also helps reduce multi-label errors

Naive Bayes vs SVM

Training Data	RAW	SWR	WS	SWR + WS
v1.0 - original	27.5, 18.1	37.0, 21.3	35.8, 20.0	46.3, 21.6
v1.0 - code	33.7, 20.5	34.9, 20.6	38.2, 20.3	41.7, 21.1
v1.0 - text	34.5, 20.3	**57.3, 21.3**	33.8, 20.3	56.3, 21.1
v2.0 - original	49.5, 12.3	60.4, 12.0	55.8, 12.3	**62.7, 11.7**
v2.0 - code	56.2, 11.8	57.1, 12.0	61.9, 11.7	61.4, 11.6
v2.0 - text	60.0, 12.6	62.2, 11.7	62.1, 11.6	62.2, 11.6
v3.0 - original	92.1, 3.0	**93.6, 2.6**	91.3, 3.0	91.8, 3.0
v3.0 - code	91.5, 3.0	92.1, 2.9	89.7, 3.2	89.6, 3.2
v3.0 - text	92.1, 2.9	93.0, 2.8	92.0. 3.0	92.7, 2.8

Classifiers	Imbalanced	Under-sampling	Over-sampling
borderAndMargin	93.41% / 75.00% / 60.00%	87.91% / 47.06% / 80.00%	98.99% / 100.0% / 90.91%
dispose	95.60% / 100.0% / 60.00%	86.81% / 45.45% / 100.0%	96.97% / 100.0% / 72.73%
drawing	98.90% / 100.0% / 90.91%	97.80% / 90.91% / 90.91%	99.07% / 100.0% / 90.91%
focus	96.70% / 100.0% / 72.73%	96.70% / 90.00% / 81.82%	96.97% / 90.00% / 81.82%
layout	93.41% / 85.71% / 54.55%	92.31% / 66.67% / 72.73%	96.00% / 76.92% / 90.91%
titleBar	93.41% / 72.73% / 72.73%	93.41% / 72.73% / 72.73%	94.05% / 75.00% / 81.82%
textIconPosition	96.70% / 100.0% / 72.73%	100.0% / 100.0% / 100.0%	98.98% / 100.0% / 90.91%
dynamicHierarchy	95.60% / 100.0% / 63.64%	81.32% / 39.29% / 100.0%	98.98% / 100.0% / 90.91%
average	**95.47**% / 91.68% / 68.41%	**92.03**% / 69.01% / 87.27%	**97.50**% / 92.74% / 86.37%

Naive Bayes vs SVM

Training Data	RAW	SWR	WS	SWR + WS
v1.0 - original	27.5, 18.1	37.0, 21.3	35.8, 20.0	46.3, 21.6
v1.0 - code	33.7, 20.5	34.9, 20.6	38.2, 20.3	41.7, 21.1
v1.0 - text	34.5, 20.3	**57.3, 21.3**	33.8, 20.3	56.3, 21.1
v2.0 - original	49.5, 12.3	60.4, 12.0	55.8, 12.3	**62.7, 11.7**
v2.0 - code	56.2, 11.8	57.1, 12.0	61.9, 11.7	61.4, 11.6
v2.0 - text	60.0, 12.6	62.2, 11.7	62.1, 11.6	62.2, 11.6
v3.0 - original	92.1, 3.0	**93.6, 2.6**	91.3, 3.0	91.8, 3.0
v3.0 - code	91.5, 3.0	92.1, 2.9	89.7, 3.2	89.6, 3.2
v3.0 - text	92.1, 2.9	93.0, 2.8	92.0. 3.0	92.7, 2.8

Classifiers	Imbalanced	Under-sampling	Over-sampling
borderAndMargin	93.41% / 75.00% / 60.00%	87.91% / 47.06% / 80.00%	98.99% / 100.0% / 90.91%
dispose	95.60% / 100.0% / 60.00%	86.81% / 45.45% / 100.0%	96.97% / 100.0% / 72.73%
drawing	98.90% / 100.0% / 90.91%	97.80% / 90.91% / 90.91%	99.07% / 100.0% / 90.91%
focus	96.70% / 100.0% / 72.73%	96.70% / 90.00% / 81.82%	96.97% / 90.00% / 81.82%
layout	93.41% / 85.71% / 54.55%	92.31% / 66.67% / 72.73%	96.00% / 76.92% / 90.91%
titleBar	93.41% / 72.73% / 72.73%	93.41% / 72.73% / 72.73%	94.05% / 75.00% / 81.82%
textIconPosition	96.70% / 100.0% / 72.73%	100.0% / 100.0% / 100.0%	98.98% / 100.0% / 90.91%
dynamicHierarchy	95.60% / 100.0% / 63.64%	81.32% / 39.29% / 100.0%	98.98% / 100.0% / 90.91%
average	**95.47**% / 91.68% / 68.41%	**92.03**% / 69.01% / 87.27%	**97.50**% / 92.74% / 86.37%

Naive Bayes vs SVM

Training Data	RAW	SWR	WS	SWR + WS
v1.0 - original	27.5, 18.1	37.0, 21.3	35.8, 20.0	46.3, 21.6
v1.0 - code	33.7, 20.5	34.9, 20.6	38.2, 20.3	41.7, 21.1
v1.0 - text	34.5, 20.3	**57.3, 21.3**	33.8, 20.3	56.3, 21.1
v2.0 - original	49.5, 12.3	60.4, 12.0	55.8, 12.3	**62.7, 11.7**
v2.0 - code	56.2, 11.8	57.1, 12.0	61.9, 11.7	61.4, 11.6
v2.0 - text	60.0, 12.6	62.2, 11.7	62.1, 11.6	62.2, 11.6
v3.0 - original	92.1, 3.0	**93.6, 2.6**	91.3, 3.0	91.8, 3.0
v3.0 - code	91.5, 3.0	92.1, 2.9	89.7, 3.2	89.6, 3.2
v3.0 - text	92.1, 2.9	93.0, 2.8	92.0. 3.0	92.7, 2.8

Classifiers	Imbalanced	Under-sampling	Over-sampling
borderAndMargin	93.41% / 75.00% / 60.00%	87.91% / 47.06% / 80.00%	98.99% / 100.0% / 90.91%
dispose	95.60% / 100.0% / 60.00%	86.81% / 45.45% / 100.0%	96.97% / 100.0% / 72.73%
drawing	98.90% / 100.0% / 90.91%	97.80% / 90.91% / 90.91%	99.07% / 100.0% / 90.91%
focus	96.70% / 100.0% / 72.73%	96.70% / 90.00% / 81.82%	96.97% / 90.00% / 81.82%
layout	93.41% / 85.71% / 54.55%	92.31% / 66.67% / 72.73%	96.00% / 76.92% / 90.91%
titleBar	93.41% / 72.73% / 72.73%	93.41% / 72.73% / 72.73%	94.05% / 75.00% / 81.82%
textIconPosition	96.70% / 100.0% / 72.73%	100.0% / 100.0% / 100.0%	98.98% / 100.0% / 90.91%
dynamicHierarchy	95.60% / 100.0% / 63.64%	81.32% / 39.29% / 100.0%	98.98% / 100.0% / 90.91%
average	**95.47**% / 91.68% / 68.41%	**92.03**% / 69.01% / 87.27%	**97.50**% / 92.74% / 86.37%

Naive Bayes vs SVM

Training Data	RAW	SWR	WS	SWR + WS
v1.0 - original	27.5, 18.1	37.0, 21.3	35.8, 20.0	46.3, 21.6
v1.0 - code	33.7, 20.5	34.9, 20.6	38.2, 20.3	41.7, 21.1
v1.0 - text	34.5, 20.3	**57.3, 21.3**	33.8, 20.3	56.3, 21.1
v2.0 - original	49.5, 12.3	60.4, 12.0	55.8, 12.3	**62.7, 11.7**
v2.0 - code	56.2, 11.8	57.1, 12.0	61.9, 11.7	61.4, 11.6
v2.0 - text	60.0, 12.6	62.2, 11.7	62.1, 11.6	62.2, 11.6
v3.0 - original	92.1, 3.0	**93.6, 2.6**	91.3, 3.0	91.8, 3.0
v3.0 - code	91.5, 3.0	92.1, 2.9	89.7, 3.2	89.6, 3.2
v3.0 - text	92.1, 2.9	93.0, 2.8	92.0. 3.0	92.7, 2.8

Classifiers	Imbalanced	Under-sampling	Over-sampling
borderAndMargin	93.41% / 75.00% / 60.00%	87.91% / 47.06% / 80.00%	98.99% / 100.0% / 90.91%
dispose	95.60% / 100.0% / 60.00%	86.81% / 45.45% / 100.0%	96.97% / 100.0% / 72.73%
drawing	98.90% / 100.0% / 90.91%	97.80% / 90.91% / 90.91%	99.07% / 100.0% / 90.91%
focus	96.70% / 100.0% / 72.73%	96.70% / 90.00% / 81.82%	96.97% / 90.00% / 81.82%
layout	93.41% / 85.71% / 54.55%	92.31% / 66.67% / 72.73%	96.00% / 76.92% / 90.91%
titleBar	93.41% / 72.73% / 72.73%	93.41% / 72.73% / 72.73%	94.05% / 75.00% / 81.82%
textIconPosition	96.70% / 100.0% / 72.73%	100.0% / 100.0% / 100.0%	98.98% / 100.0% / 90.91%
dynamicHierarchy	95.60% / 100.0% / 63.64%	81.32% / 39.29% / 100.0%	98.98% / 100.0% / 90.91%
average	**95.47**% / 91.68% / 68.41%	**92.03**% / 69.01% / 87.27%	**97.50**% / 92.74% / 86.37%

Naive Bayes vs SVM

Training Data	RAW	SWR	WS	SWR + WS
v1.0 - original	27.5, 18.1	37.0, 21.3	35.8, 20.0	46.3, 21.6
v1.0 - code	33.7, 20.5	34.9, 20.6	38.2, 20.3	41.7, 21.1
v1.0 - text	34.5, 20.3	**57.3, 21.3**	33.8, 20.3	56.3, 21.1
v2.0 - original	49.5, 12.3	60.4, 12.0	55.8, 12.3	**62.7, 11.7**
v2.0 - code	56.2, 11.8	57.1, 12.0	61.9, 11.7	61.4, 11.6
v2.0 - text	60.0, 12.6	62.2, 11.7	62.1, 11.6	62.2, 11.6
v3.0 - original	92.1, 3.0	**93.6, 2.6**	91.3, 3.0	91.8, 3.0
v3.0 - code	91.5, 3.0	92.1, 2.9	89.7, 3.2	89.6, 3.2
v3.0 - text	92.1, 2.9	93.0, 2.8	92.0. 3.0	92.7, 2.8

Classifiers	Imbalanced	Under-sampling	Over-sampling
borderAndMargin	93.41% / 75.00% / 60.00%	87.91% / 47.06% / 80.00%	98.99% / 100.0% / 90.91%
dispose	95.60% / 100.0% / 60.00%	86.81% / 45.45% / 100.0%	96.97% / 100.0% / 72.73%
drawing	98.90% / 100.0% / 90.91%	97.80% / 90.91% / 90.91%	99.07% / 100.0% / 90.91%
focus	96.70% / 100.0% / 72.73%	96.70% / 90.00% / 81.82%	96.97% / 90.00% / 81.82%
layout	93.41% / 85.71% / 54.55%	92.31% / 66.67% / 72.73%	96.00% / 76.92% / 90.91%
titleBar	93.41% / 72.73% / 72.73%	93.41% / 72.73% / 72.73%	94.05% / 75.00% / 81.82%
textIconPosition	96.70% / 100.0% / 72.73%	100.0% / 100.0% / 100.0%	98.98% / 100.0% / 90.91%
dynamicHierarchy	95.60% / 100.0% / 63.64%	81.32% / 39.29% / 100.0%	98.98% / 100.0% / 90.91%
average	**95.47**% / 91.68% / 68.41%	**92.03**% / 69.01% / 87.27%	**97.50**% / 92.74% / 86.37%

Naive Bayes vs SVM

Training Data	RAW	SWR	WS	SWR + WS
v1.0 - original	27.5, 18.1	37.0, 21.3	35.8, 20.0	46.3, 21.6
v1.0 - code	33.7, 20.5	34.9, 20.6	38.2, 20.3	41.7, 21.1
v1.0 - text	34.5, 20.3	**57.3, 21.3**	33.8, 20.3	56.3, 21.1
v2.0 - original	49.5, 12.3	60.4, 12.0	55.8, 12.3	**62.7, 11.7**
v2.0 - code	56.2, 11.8	57.1, 12.0	61.9, 11.7	61.4, 11.6
v2.0 - text	60.0, 12.6	62.2, 11.7	62.1, 11.6	62.2, 11.6
v3.0 - original	92.1, 3.0	**93.6, 2.6**	91.3, 3.0	91.8, 3.0
v3.0 - code	91.5, 3.0	92.1, 2.9	89.7, 3.2	89.6, 3.2
v3.0 - text	92.1, 2.9	93.0, 2.8	92.0. 3.0	92.7, 2.8

Classifiers	Imbalanced	Under-sampling	Over-sampling
borderAndMargin	93.41% / 75.00% / 60.00%	87.91% / 47.06% / 80.00%	98.99% / 100.0% / 90.91%
dispose	95.60% / 100.0% / 60.00%	86.81% / 45.45% / 100.0%	96.97% / 100.0% / 72.73%
drawing	98.90% / 100.0% / 90.91%	97.80% / 90.91% / 90.91%	99.07% / 100.0% / 90.91%
focus	96.70% / 100.0% / 72.73%	96.70% / 90.00% / 81.82%	96.97% / 90.00% / 81.82%
layout	93.41% / 85.71% / 54.55%	92.31% / 66.67% / 72.73%	96.00% / 76.92% / 90.91%
titleBar	93.41% / 72.73% / 72.73%	93.41% / 72.73% / 72.73%	94.05% / 75.00% / 81.82%
textIconPosition	96.70% / 100.0% / 72.73%	100.0% / 100.0% / 100.0%	98.98% / 100.0% / 90.91%
dynamicHierarchy	95.60% / 100.0% / 63.64%	81.32% / 39.29% / 100.0%	98.98% / 100.0% / 90.91%
average	**95.47**% / 91.68% / 68.41%	**92.03**% / 69.01% / 87.27%	**97.50**% / 92.74% / 86.37%

Naive Bayes vs SVM

Training Data	RAW	SWR	WS	SWR + WS
v1.0 - original	27.5, 18.1	37.0, 21.3	35.8, 20.0	46.3, 21.6
v1.0 - code	33.7, 20.5	34.9, 20.6	38.2, 20.3	41.7, 21.1
v1.0 - text	34.5, 20.3	**57.3, 21.3**	33.8, 20.3	56.3, 21.1
v2.0 - original	49.5, 12.3	60.4, 12.0	55.8, 12.3	**62.7, 11.7**
v2.0 - code	56.2, 11.8	57.1, 12.0	61.9, 11.7	61.4, 11.6
v2.0 - text	60.0, 12.6	62.2, 11.7	62.1, 11.6	62.2, 11.6
v3.0 - original	92.1, 3.0	**93.6, 2.6**	91.3, 3.0	91.8, 3.0
v3.0 - code	91.5, 3.0	92.1, 2.9	89.7, 3.2	89.6, 3.2
v3.0 - text	92.1, 2.9	93.0, 2.8	92.0. 3.0	92.7, 2.8

Classifiers	Imbalanced	Under-sampling	Over-sampling
borderAndMargin	93.41% / 75.00% / 60.00%	87.91% / 47.06% / 80.00%	98.99% / 100.0% / 90.91%
dispose	95.60% / 100.0% / 60.00%	86.81% / 45.45% / 100.0%	96.97% / 100.0% / 72.73%
drawing	98.90% / 100.0% / 90.91%	97.80% / 90.91% / 90.91%	99.07% / 100.0% / 90.91%
focus	96.70% / 100.0% / 72.73%	96.70% / 90.00% / 81.82%	96.97% / 90.00% / 81.82%
layout	93.41% / 85.71% / 54.55%	92.31% / 66.67% / 72.73%	96.00% / 76.92% / 90.91%
titleBar	93.41% / 72.73% / 72.73%	93.41% / 72.73% / 72.73%	94.05% / 75.00% / 81.82%
textIconPosition	96.70% / 100.0% / 72.73%	100.0% / 100.0% / 100.0%	98.98% / 100.0% / 90.91%
dynamicHierarchy	95.60% / 100.0% / 63.64%	81.32% / 39.29% / 100.0%	98.98% / 100.0% / 90.91%
average	**95.47**% / 91.68% / 68.41%	**92.03**% / 69.01% / 87.27%	**97.50**% / 92.74% / 86.37%

Naive Bayes vs SVM

Training Data	RAW	SWR	WS	SWR + WS
v1.0 - original	27.5, 18.1	37.0, 21.3	35.8, 20.0	46.3, 21.6
v1.0 - code	33.7, 20.5	34.9, 20.6	38.2, 20.3	41.7, 21.1
v1.0 - text	34.5, 20.3	**57.3, 21.3**	33.8, 20.3	56.3, 21.1
v2.0 - original	49.5, 12.3	60.4, 12.0	55.8, 12.3	**62.7, 11.7**
v2.0 - code	56.2, 11.8	57.1, 12.0	61.9, 11.7	61.4, 11.6
v2.0 - text	60.0, 12.6	62.2, 11.7	62.1, 11.6	62.2, 11.6
v3.0 - original	92.1, 3.0	**93.6, 2.6**	91.3, 3.0	91.8, 3.0
v3.0 - code	91.5, 3.0	92.1, 2.9	89.7, 3.2	89.6, 3.2
v3.0 - text	92.1, 2.9	93.0, 2.8	92.0. 3.0	92.7, 2.8

Classifiers	Imbalanced	Under-sampling	Over-sampling
borderAndMargin	93.41% / 75.00% / 60.00%	87.91% / 47.06% / 80.00%	98.99% / 100.0% / 90.91%
dispose	95.60% / 100.0% / 60.00%	86.81% / 45.45% / 100.0%	96.97% / 100.0% / 72.73%
drawing	98.90% / 100.0% / 90.91%	97.80% / 90.91% / 90.91%	99.07% / 100.0% / 90.91%
focus	96.70% / 100.0% / 72.73%	96.70% / 90.00% / 81.82%	96.97% / 90.00% / 81.82%
layout	93.41% / 85.71% / 54.55%	92.31% / 66.67% / 72.73%	96.00% / 76.92% / 90.91%
titleBar	93.41% / 72.73% / 72.73%	93.41% / 72.73% / 72.73%	94.05% / 75.00% / 81.82%
textIconPosition	96.70% / 100.0% / 72.73%	100.0% / 100.0% / 100.0%	98.98% / 100.0% / 90.91%
dynamicHierarchy	95.60% / 100.0% / 63.64%	81.32% / 39.29% / 100.0%	98.98% / 100.0% / 90.91%
average	**95.47**% / 91.68% / 68.41%	**92.03**% / 69.01% / 87.27%	**97.50**% / 92.74% / 86.37%

Conclusion and future work

- Goal: to classify **major topic** of API discussion
- Naive Bayes can achieve **average test accuracy** of 94.1%
- **Training dataset size** most important for increasing classification accuracy, but increase in accuracy plateaus as size passes some threshold
- **Multi-label** documents are major cause for classification errors
- **Data** are publicly available
- Future: to test more APIs, with more categories, and scalability