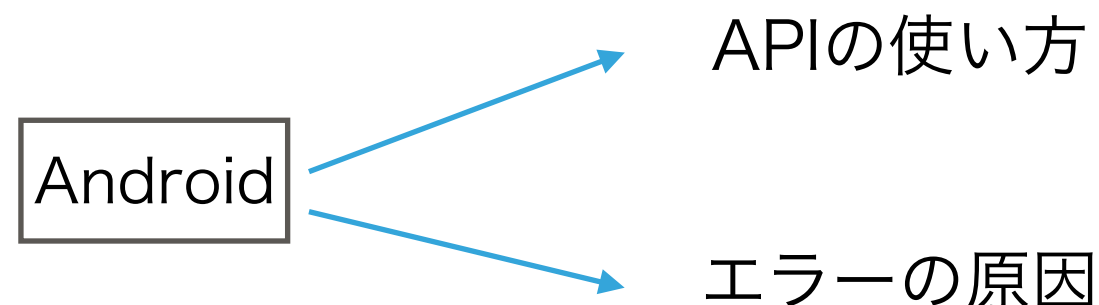


Automatically Classifying Posts into Question Categories on Stack Overflow

**Stefanie Beyer,
Christian Macho,
Martin Pinzger ,
Massimiliano Di Penta**

背景

- StackOverflowに質問をする開発者の理由は様々である
- 質問を分類するためにタグが導入されている
 - 開発者が求めている内容の質問を見つけやすくなる
- StackOverflowの質問された理由について分類する質問が広く研究されている
 - ・ 理由：APIの使い方を知りたい, エラーの原因を知りたい
 - 理由のカテゴリー分けができることで, 質問を探しやすくなる



既存研究と問題点

- 既存研究ではいくつかの投稿を分類するアプローチが取られている
 - 手作業による分類
 - ・ ラベルのないより大きなデータには適用できない
 - トピックモデル
 - ・ 教師なしのためラベルがないため、既存手法として比較が難しい
 - k-NN
 - ・ precision 41.33%とかなり低い
- 既存研究により分類分けされているカテゴリー異なる

目的とアプローチ

- 目的
 - 質問を質問内容に応じて自動分類する手法を構築する
- アプローチ
 - 質問のカテゴリの分類を行う
 - 自動分類を行うために手作業でラベル付を行う
 - カテゴリ分けを自動で行うため分類器を比較し、最適な分類器と設定を分析する
 - ・ 今回はAndroidに関する質問のみを対象とする

質問のカテゴリーの分類

- 5つの既存研究のカテゴリー分類から共通部分をまとめ新たに7つのカテゴリー分類を行う
 - 一部のカテゴリーは削除(ハードウェア, デバイスなど)
 - 各研究のカテゴリー分類の定義を確認し同様のものをマージする
 - 3名で議論し実施

質問のカテゴリーの分類

7つのカテゴリーに分類

Table 1: Our 7 question categories harmonized from the five prior approaches [1, 4, 5, 27, 31].

?	Rosen <i>et al.</i> [27]	Allamanis <i>et al.</i> [1]	Treude <i>et al.</i> [31]	Beyer <i>et al.</i> [5]	Beyer <i>et al.</i> [4]
API USAGE	<u>How</u> : A how type of questions asks for ways to achieve a goal. These questions can ask for instructions on how to do something programmatically to how to setup an environment. A sample how question asks: How can I disable landscape mode for some of the views in my Android app?	How to implement something: create, to create, is creating, call, can create, add, want to create	<u>How-to</u> : Questions that ask for instructions, e.g. "How to crop image by 160 degrees from center in asp.net".	<u>How-to</u> : the questioner does not know how to implement it. The questioner often asks how to integrate a given solution into her own code or asks for examples.	
		Way of using something: to use, can use, to do, want to use, to get, can do, instead of			<u>Interaction of API Classes</u> : Furthermore, several posts discuss the interaction of API classes, such as Activity, AsyncTask, and Intents.
DISCREPANCY	<u>Why</u> : why type of questions are used to ask the reason, cause, or purpose for something. They typically involve questions clarifying why an error has happened or why their code is not doing what they expect. An example why question is: I don't understand why it randomly occurs?	<u>Do not work</u> : doesn't work, work, try, didn't, won't, isn't, wrong, run, happen, cause, occur, fail, work, check, to see, fine, due	<u>Discrepancy</u> : Some unexpected behavior that the person asking the question wants explained, e.g. "iphone - Coremotion acceleration always zero".	<u>What is the Problem</u> : problems where the questioner has an idea how to solve it, but was not able to implement it correctly. The posts often contain How to...? questions, for which there is no working solution.	
ERRORS			<u>Error</u> : Questions that include a specific error message, e.g. C# Obscure error: file "could not be refactored"	<u>Error</u> : describe the occurrence of errors, exceptions, crashes or even compiler errors. All posts in this category contain a stack trace, error message, or warning.	<u>Exception Handling</u> : 17 posts discuss problems with handling exceptions.
REVIEW	<u>What</u> : A what type of question asks for information about something. They can be more abstract and conceptual in nature, ask for help in making a decision, or ask about non-functional requirements. For example questions about specific information about a programming concept: Explain to me what is a setter and getter. What are setters and getters? couldn't find it on wikipedia and in other places.		<u>Decision Help</u> : Asking for an opinion, e.g., Should a business object know about its corresponding contract object.	<u>Better Solution</u> : contain questions for better solutions or best practice solutions. Typically, the questioner already has an unsatisfactory solution for the problem.	
CONCEPTUAL		<u>How/Why something works</u> : hope, make, understand, give, to make, work, read, explain, check	<u>Review</u> : Questions that are either implicitly or explicitly asking for a code review, e.g. "Simple file download via HTTP - is this sufficient?".		
			<u>Conceptual</u> : Questions that are abstract and do not have a concrete use case, e.g. "Concept of xml sitemaps".	<u>Why</u> : focus on obtaining background information on a component or life cycle. The questioner asks for explanation and understanding.	
				<u>Is it possible</u> : contain questions to get more information about the possibilities and limitations of Android apps or several APIs.	
API CHANGE				<u>Version</u> : deal with problems that occur when changing the API level. Furthermore, this category contains posts that deal with the compatibility of API versions.	<u>API Changes</u> : Further 3 posts discuss how to implement features for newer or older versions of the API. In 2 of the 100 posts the problem relates to deprecated methods in the API classes. 3 posts discuss bugs in the Android API and restrictions of Android versions to access Micro SD cards.
LEARNING			<u>Learning a Language/Technology</u> : learn, to learn, start, read, understand, recommend, find, good		<u>Tutorials/Docu</u> : In 10 posts, the developers mention tutorials and documentation should cover parts of the Android API in more detail.

カテゴリー

- API USAGE
 - APIの使い方についての質問
- Discrepancy
 - 正しく動作しない, 動かないについての質問
- Errors
 - エラーについての質問
- Review
 - アドバイスを貰いたい質問
 - ・ 作成したコードに欠陥がないか, どのAPIを選択すべきか
- Conceptual
 - 概念についての質問
 - ・ デザインパターンやライブラリの設計の内容など

カテゴリー

- API CHAGE
 - APIのバージョンについての質問
- LEARNING
 - 言語の学び方やチュートリアルについての質問

手作業によるラベル付

- データセット
 - StackOverflowのAndroidタグの付けられた質問
 - ・ 1,052,568件からランダムで500件抽出
- 手順
 - 作業員2名が作業を行う。最初の50件は同様の質問にラベル付を行い結果のカッパ係数を算出
 - 残りの450件は各作業員が別々に分担して、ラベル付けを行う（マルチラベル）
 - ラベル付の決め手となったフレーズにはチェックを入れる

カッパ係数：2名のラベル付の一致度
0~1の範囲を取る

< 0	No agreement	一致していない
0.00 – 0.20	Slight	わずかに一致
0.21 – 0.40	Fair	おおむね一致
0.41 – 0.60	Moderate	適度に一致
0.61 – 0.80	Substantial	かなり一致
0.81 – 1.00	Almost perfect	ほとんど一致

手作業によるラベル付

- REVIEWにラベル付けされた質問
 - 分類のキーとなったフレーズにはチェックを入れておき分類器を作成する際に利用

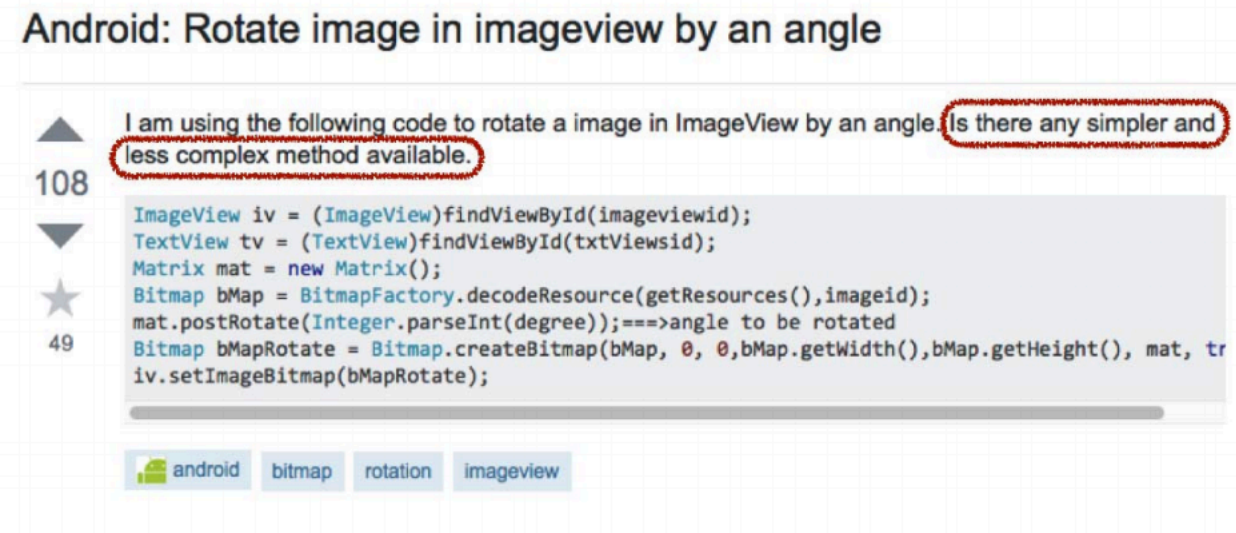


Figure 1: Question 8981845 from SO with the phrase marked in red that is indicating the question category REVIEW.

手作業によるラベル付の結果

Table 2: Number of posts per question category and most frequently used phrases to identify each question category.

Category	# of posts	# of phrases	most frequently used phrases (count)
API USAGE	206	293	how to (145), how can/could I (75), how do I (28)
CONCEPTUAL	145	211	is there a/any way to (36), what is the difference between/the use of/the purpose of (26), can I use (25), is it possible to (21)
DISCREPANCY	129	206	i try/tried to (60), do/does not work (45), what is/am i doing wrong (26), solve/fix/I have the problem (24)
ERRORS	93	246	(fatal/uncaught/throwing) exception (130), get/getting/got (an) error(s) (34)
REVIEW	79	101	is there a better/best/proper/correct/more efficient/simpler way to (32), (what) should I use/switch/do (13), is this/my understandings right/wrong (8)
API CHANGE	30	54	before/after (the) update/upgrade (to API/version/level) (14), work above/below/with API level/android/version x.x (but) (6)
LEARNING	30	36	suggest/give me/find (links to) tutorial(s) (21)

- 500件の分類と1147件の分類フレーズを抽出

自動分類の分類器

- 分類器

- 1.SVM

- 2.RandomForest

- 前処理

- 1.ストップワードの除去

- 2.N-grams

- ・ $n = 1 \sim 3$ で実施

- 3.Pruning

- ・ 頻出の単語を削除する

- ・ 80%以上の投稿で使われる単語を除去

- 4.Re-sampling

- ・ SMOTEアルゴリズムを利用してデータの不均衡に対処

- ・ 少ないデータを人工的に作成し, 多いデータを減らす

自動分類の分類器

- 学習データへの入力データ
 - 1.投稿に付けられたタグのみを入力
 - 2.タイトル・本文すべてを入力とする場合
 - 1.タイトル・本文全て
 - 2.タイトル本文の中の分類フレーズのみ
 - 3.両方

実験の目的と設定

- 目的
 - 最も良いモデルの作成方法を調査する
- データの分割
 - 訓練データ 90% テストデータ 10%
 - ・ 各カテゴリの投稿の10%または3件以上の投稿が含まれるように
 - 20回試行
- 手法の組み合わせ
 - $7(\text{カテゴリー}) \times 82(\text{configurations}) \times 2(\text{RF or SVM}) \times 2(\text{full text or phrases}) = 2,296$ パターンを試行
- 評価指標
 - Accuracy, Precision, Recall, F-score, AUC

テキスト全てを入力とした場合の結果

- 各分類で最も良い組み合わせのみ

Table 3: Best configuration and performance over 20 runs using RF with the full text as input.

category	type	n-grams	stop words	prune	re-sample	acc	prec _{avg}	rec _{avg}	f _{avg}	auc
API CHANGE	pos	n=1	T	F	T	0.93	0.92	0.93	0.92	0.83
API USAGE	combi	n _{txt} =1 n _{pos} =2	F	T	T	0.89	0.89	0.89	0.89	0.95
CONCEPTUAL	pos	n=1	F	F	T	0.66	0.64	0.66	0.64	0.61
DISCREPANCY	pos	n=1	T	F	T	0.74	0.72	0.74	0.72	0.68
LEARNING	pos	n=1	F	F	T	0.94	0.91	0.94	0.92	0.65
ERRORS	combi	n _{txt} =1 n _{pos} =1	T	T	F	0.84	0.85	0.84	0.81	0.96
REVIEW	pos	n=2	T	T	T	0.85	0.76	0.85	0.78	0.71
average	-	-	-	-	-	0.83	0.81	0.83	0.81	0.77

Table 4: Best configuration and performance over 20 runs using SVM with the full text as input.

category	type	n-grams	stop words	prune	re-sample	acc	prec _{avg}	rec _{avg}	f _{avg}	auc
API CHANGE	combi	n _{txt} =1 n _{pos} =2	T	T	T	0.97	0.96	0.97	0.96	0.94
API USAGE	combi	n _{txt} =1 n _{pos} =2	F	T	T	0.74	0.75	0.74	0.72	0.84
CONCEPTUAL	combi	n _{txt} =1 n _{pos} =2	F	F	T	0.73	0.70	0.73	0.69	0.74
DISCREPANCY	combi	n _{txt} =1 n _{pos} =1	F	F	T	0.72	0.72	0.72	0.72	0.73
LEARNING	combi	n _{txt} =1 n _{pos} =3	T	F	T	0.95	0.92	0.95	0.93	0.84
ERRORS	combi	n _{txt} =1 n _{pos} =2	F	T	T	0.84	0.84	0.84	0.84	0.80
REVIEW	combi	n _{txt} =1 n _{pos} =3	T	F	T	0.85	0.83	0.85	0.82	0.70
average	-	-	-	-	-	0.83	0.82	0.83	0.81	0.80

フレーズを入力とした場合の結果

- 各分類で最も良い組み合わせのみ

Table 5: Best configuration and performance over 20 runs using RF with the phrases as input.

category	type	n-grams	stop words	prune	re-sample	acc	prec _{avg}	rec _{avg}	f _{avg}	auc
API CHANGE	combi	n _{txt} =1 n _{pos} =1	F	F	T	0.95	0.97	0.95	0.96	0.98
API USAGE	txt	n=2	F	T	F	0.85	0.86	0.85	0.85	0.92
CONCEPTUAL	combi	n _{txt} =2 n _{pos} =1	F	T	F	0.82	0.82	0.82	0.82	0.84
DISCREPANCY	combi	n _{txt} =2 n _{pos} =1	F	T	F	0.79	0.79	0.79	0.79	0.81
LEARNING	txt	n=1	F	F	T	0.95	0.96	0.95	0.95	0.91
ERRORS	txt	n=1	T	T	F	0.90	0.90	0.90	0.89	0.95
REVIEW	txt	n=3	F	F	T	0.90	0.91	0.90	0.88	0.82
average	-	-	-	-	-	0.88	0.89	0.88	0.88	0.89

Table 6: Best configuration and performance over 20 runs using SVM with the phrases as input.

category	type	n-grams	stop words	prune	re-sample	acc	prec _{avg}	rec _{avg}	f _{avg}	auc
API CHANGE	txt	n=2	T	F	T	0.95	0.91	0.95	0.93	0.76
API USAGE	combi	n _{txt} =1 n _{pos} =3	F	F	T	0.76	0.80	0.76	0.74	0.86
CONCEPTUAL	txt	n=3	F	F	T	0.75	0.78	0.75	0.69	0.71
DISCREPANCY	txt	n=2	T	T	T	0.78	0.76	0.78	0.72	0.70
LEARNING	txt	n=3	T	F	T	0.94	0.90	0.94	0.92	0.55
ERRORS	combi	n _{txt} =1 n _{pos} =3	T	F	T	0.82	0.77	0.82	0.75	0.67
REVIEW	txt	n=3	T	F	T	0.86	0.85	0.86	0.82	0.62
average	-	-	-	-	-	0.84	0.82	0.84	0.80	0.70

最も良いモデルのパフォーマンス調査

- 目的
 - 最も良いモデル作成の組み合わせが他のデータセットに対しても適用可能か調査する
- 分類器
 - Random Forest + テキストのフレーズのみ
 - 他の設定は前実験で最も良い組み合わせだったもの
- データセット
 - 学習データ
 - 500件のラベル付データ
 - テストデータ
 - 追加で100件, 手動でラベル付を行ったデータ
 - 試行回数
 - 100回

結果

Table 9: The performance of the classification on the test set of 100 SO posts using RF and phrases as input text.

category	acc	prec _{avg}	rec _{avg}	f _{avg}	auc
API CHANGE	0.92	0.97	0.92	0.94	0.90
API USAGE	0.82	0.82	0.82	0.82	0.88
CONCEPTUAL	0.79	0.78	0.79	0.78	0.84
DISCREPANCY	0.72	0.71	0.72	0.70	0.78
LEARNING	0.94	0.94	0.94	0.94	0.73
ERRORS	0.79	0.79	0.79	0.79	0.94
REVIEW	0.92	0.93	0.92	0.90	0.66
average	0.84	0.85	0.83	0.84	0.82

追加100件のデータに対しても，高い精度で分類できている

まとめ・所感

- まとめ

- 質問のカテゴリー分類の実施
- カテゴリーの自動分類手法の最適なモデル作成方法の調査
- 追加データに対して評価実験を行い手法の妥当性を評価

- 所感

- 背景とカテゴリー分けの部分はちょっとイマイチな気がする？
- 追加データに対して評価を行う部分が参考になった