Analysis and Detection of Information Types of Open Source Software Issue Discussions

Deeksha Arya*, Wenting Wang*, Jin L.C. Guo*, Jinghui Cheng†

*School of Computer Science, McGill University, Montreal, Canada

†Department of Computer and Software Engineering, Polytechnique Montreal, Montreal, Canada

2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)

Abstract（1/2）

- Issue Tracking System（ITS）は開発者やユーザ，ステークホルダーにとってソフトウェアプロジェクトについての豊富な情報源である

- 例えば…
  - エンドユーザは，彼らのプロジェクトの障害を取り除くために一時的でも早急な解決策を求めるかもしれない

Abstract（2/2）

- しかし，スレッドから情報を発見したり検索することは大変である
- この論文では，OSSの議論中に存在している情報のタイプを識別し，議論中の文をそれらの情報タイプに自動的に分類する技術の使用による有用性を調査する

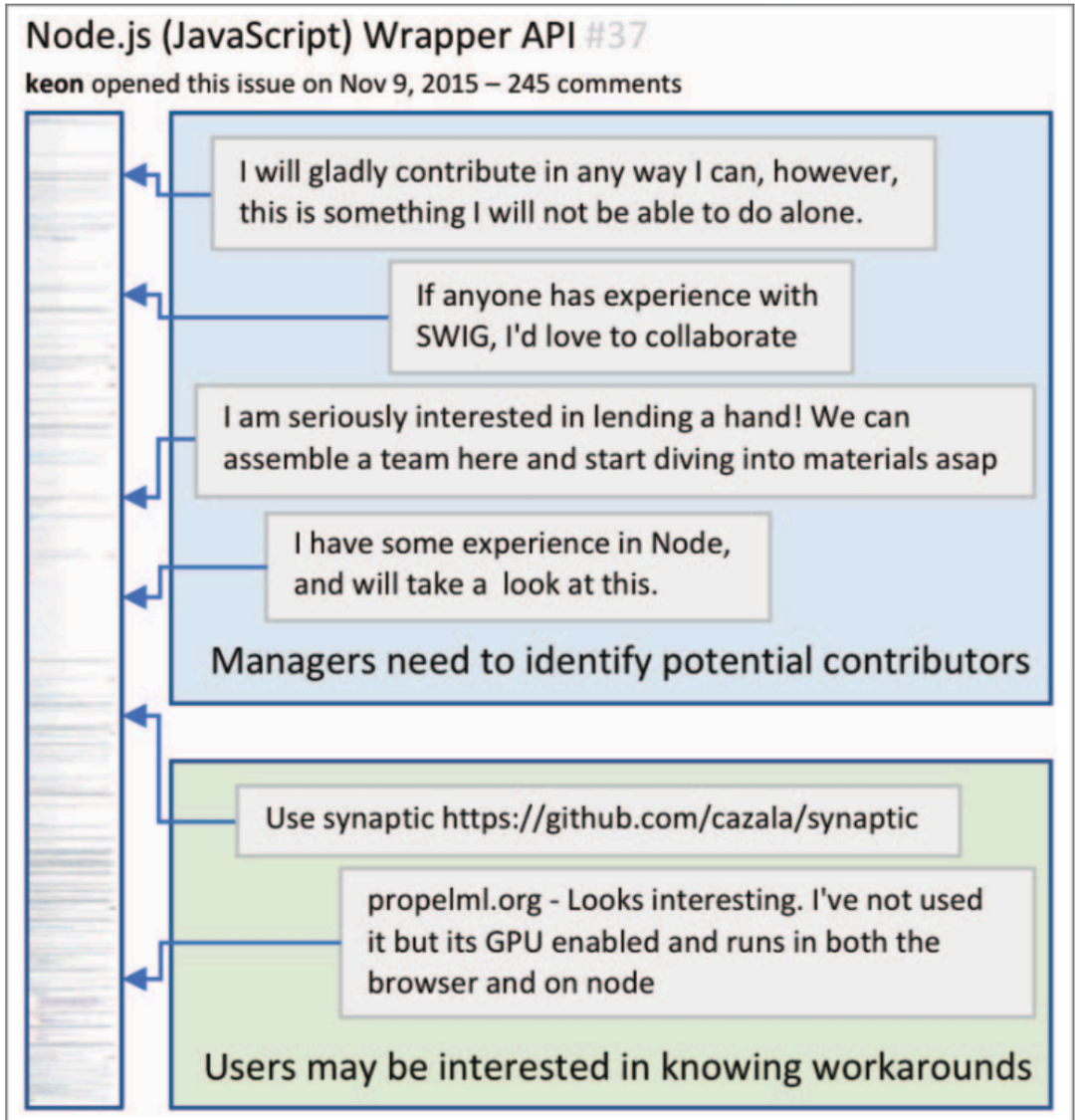


Fig. 1: Example issue discussion thread *TensorFlow #37* (compressed to the left), where useful information such as potential contributors and temporary workarounds is hidden in the 245 comments.

Research Question

RQ1：さまざまなステークホルダーのニーズを潜在的に満たすことができる，issueの議論で提示されている主な情報タイプは何か？

RQ2：自動分類手法は問題のコメントの情報タイプをどの程度まで識別できるか？

Approach to RQ1

- データ準備
  - GitHubにホストしているAIライブラリであるTensorFlow，scikit-learn, spaCyを調査
  - それぞれのプロジェクトで，最もコメントされている5つの未解決問題を選択
    →15件の議論スレッドから合計4656文を取得

- Qualitative Content Analysis Process[18]
  - issue title，議論スレッド，議論スレッド中の全てのコメントのそれぞれの文の情報タイプを明確にする

[18] S. B. Merriam and E. J. Tisdell. Qualitative Research: A Guide to Design and Implementation. Jossey-Bass, 4th edition, 2015.

Qualitative Content Analysis Process

A. 4人のアノテーターが，選択した議論スレッドを文レベルで独自に符号化する

B. その4人のデータをKJ法によってグループ化し，異なるアノテーターによる異なる関連符号にリンクする準備用の符号ブックを作成

C. 符号ブックを用いて，二人のアノテーターでそれぞれのプロジェクトから1つずつ無作為に選ばれた3つの議論スレッドの二回目の符号化を行う
   - このステップで符号化された1090文の全体的なコーエンのカッパ係数は0.71で，二つのデータでかなりの一致が見られた[8][17]

D. 最後に，それら二人のアノテーターは別々に最終的な符号ブックを用いて残りの議論スレッドの再符号化を行う

[8]J. Cohen. A coefficient of agreement for nominal scales. Educational and psychological measurement, 20(1):37–46, 1960.
[17]J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. Biometrics, 33(1):159–174, 1977.

Result RQ1

- 議論スレッド中の54種類の情報タイプを明らかにした
  - それらをさらに16種類の主なタイプにグループ分けした

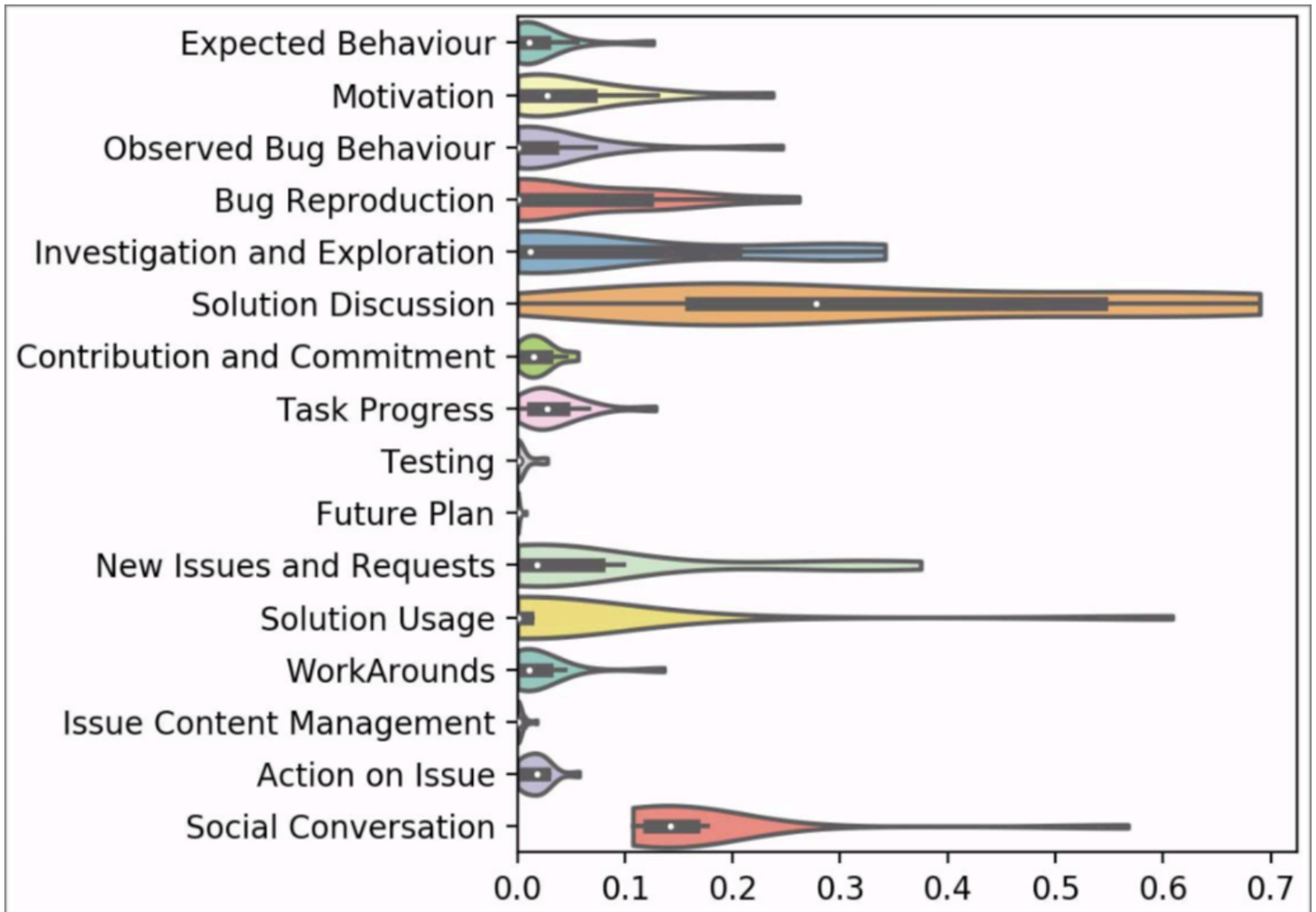
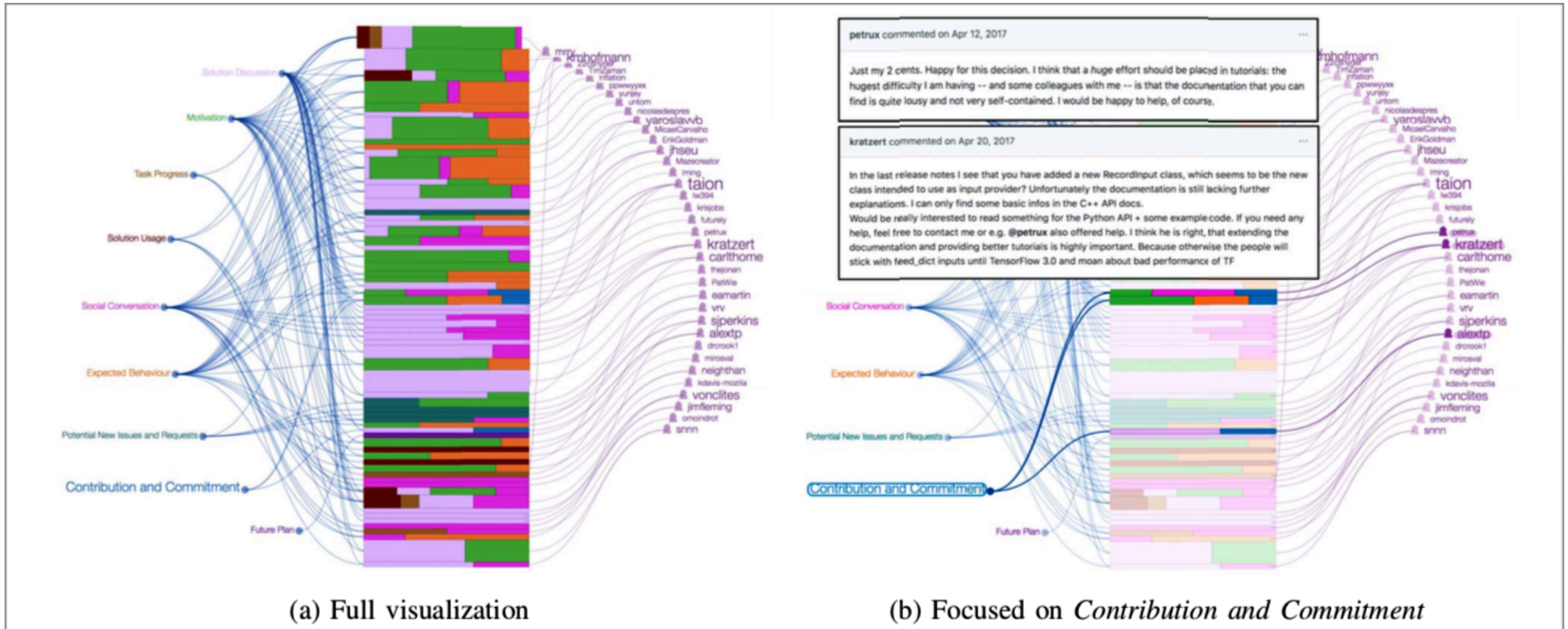


Fig. 2: Distribution of the percentage of the identified information types in the selected issues.

Use Case



(a) Full visualization

(b) Focused on *Contribution and Commitment*

Fig. 3: Visualization of a discussion thread *TensorFlow #7951* from beginning to May 20, 2017 with information types

- MotivationやExpected Behavioursといった情報タイプから，入力パイプライン再設計の動機やその後の動きなどの情報を示すコメントを素早く特定できる
- issueの議論中で，自分のニーズと一致しているかもしれない他の貢献について示唆しているコメントを見つけることができる

Automated Information Type Detection（1/3）

1. 文の前処理
   - ソースコードブロック，引用文，URL，ユーザへのメンションなどを全てトークンに置き換え，句読点を削除

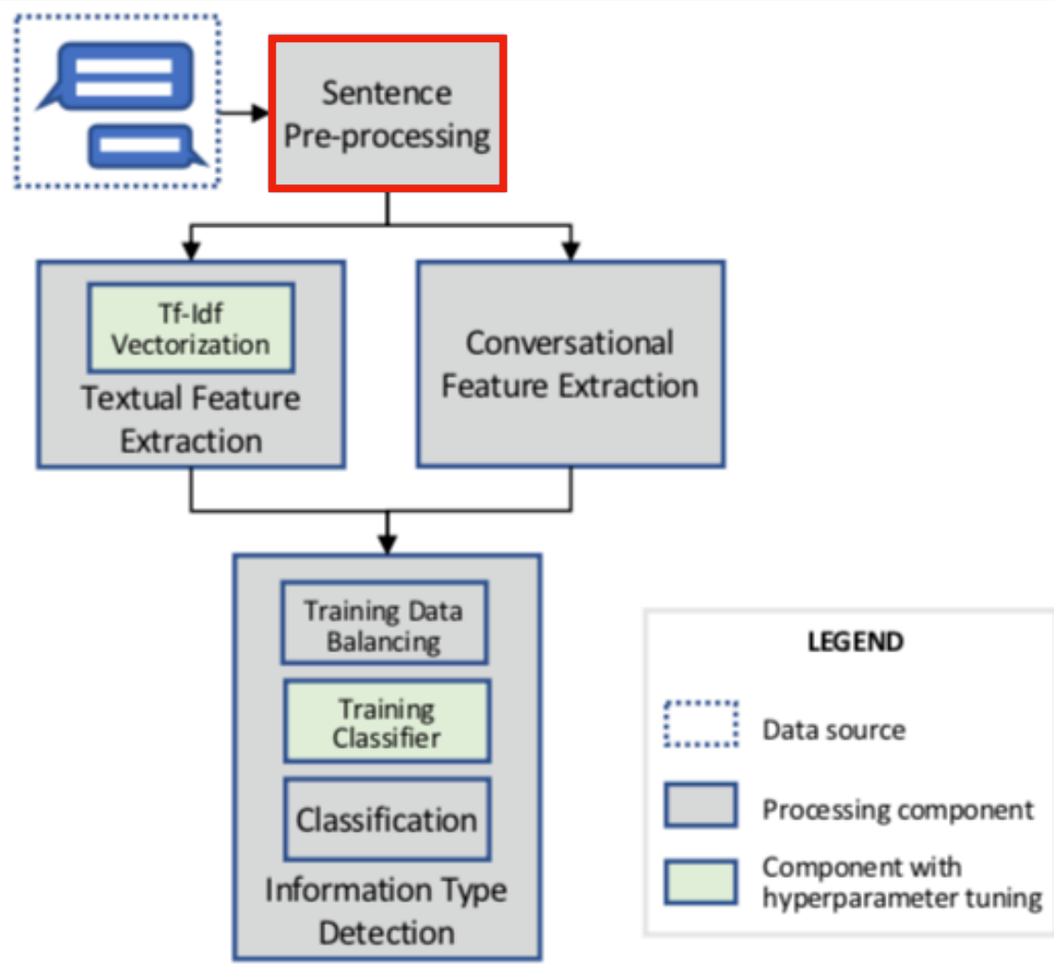


Fig. 4: The process of supervised information type detection

Automated Information Type Detection（2/3）

2. 特徴抽出
   - Textual Features
   - Conversation Features

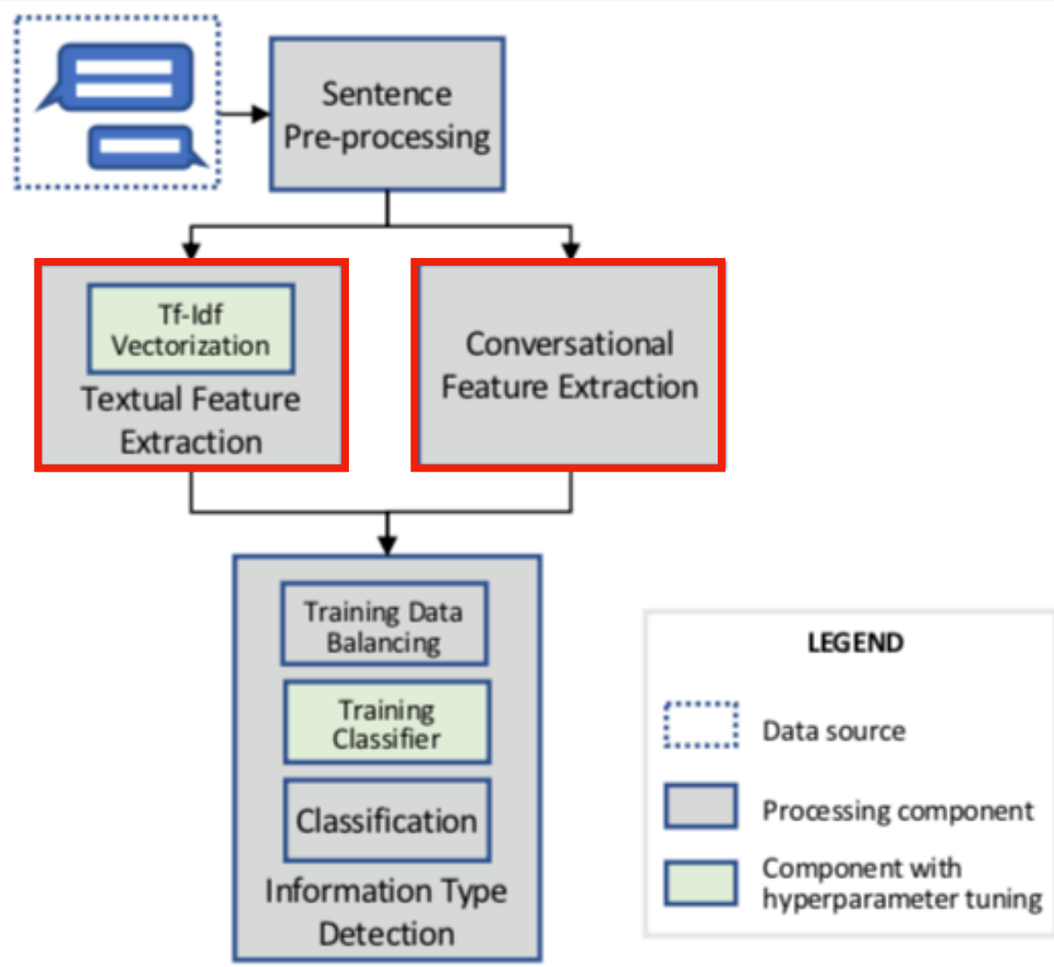


Fig. 4: The process of supervised information type detection

TABLE II: List of conversational features

Feature Type	Feature Name	Description	Value Range
Participant	AA	Author's association with repository.	{OWNER, CL, MBR, OTHER}
	BEGAUTH	Flag of whether the comment author also posted the original issue.	{True, False}
Length	LEN	Length of the sentence in terms of character count.	{Positive Numbers}
	TLEN	Count of words in sentence divided by that of the longest sentence in thread	(0, 1]
	CLEN	Count of words in sentence divided by that of the longest sentence in comment.	(0, 1]
Structural	TLOC	Position of sentence in comment divided by the number of sentences in comment.	(0, 1]
	CLOC	Position of sentence in conversation divided by the number of sentences in thread	(0, 1]
	FIRST_TURN	Flag of whether if this is in the first comment.	{True, False}
	LAST_TURN	Flag of whether this is the last comment or not	{True, False}
Temporal	TPOS1	Time from beginning of conversation to comment divided by the total time of thread.	[0, 1]
	TPOS2	Time from comment to end of conversation divided by the total time of thread.	[0, 1]
	PPAU	Time from previous comment to current comment (normalized).	[0, 1]
	NPAU	Time from current comment to next comment (normalized).	[0, 1]
Code	HAS_CODE	Flag to indicate whether the comment contains a code snippet.	{True, False}

Note: CL - Collaborator, MBR - Member

Conversational Featuresの分類リスト

Automated Information Type Detection（3/3）

3. 情報タイプ検出
   - 教師あり学習により分類
     - ロジスティック回帰
     - ランダムフォレスト
   - 情報タイプの分布がかなり偏っているため，訓練用データのバランスを取る必要がある
     - Adjusting class weight
     - SMOTE

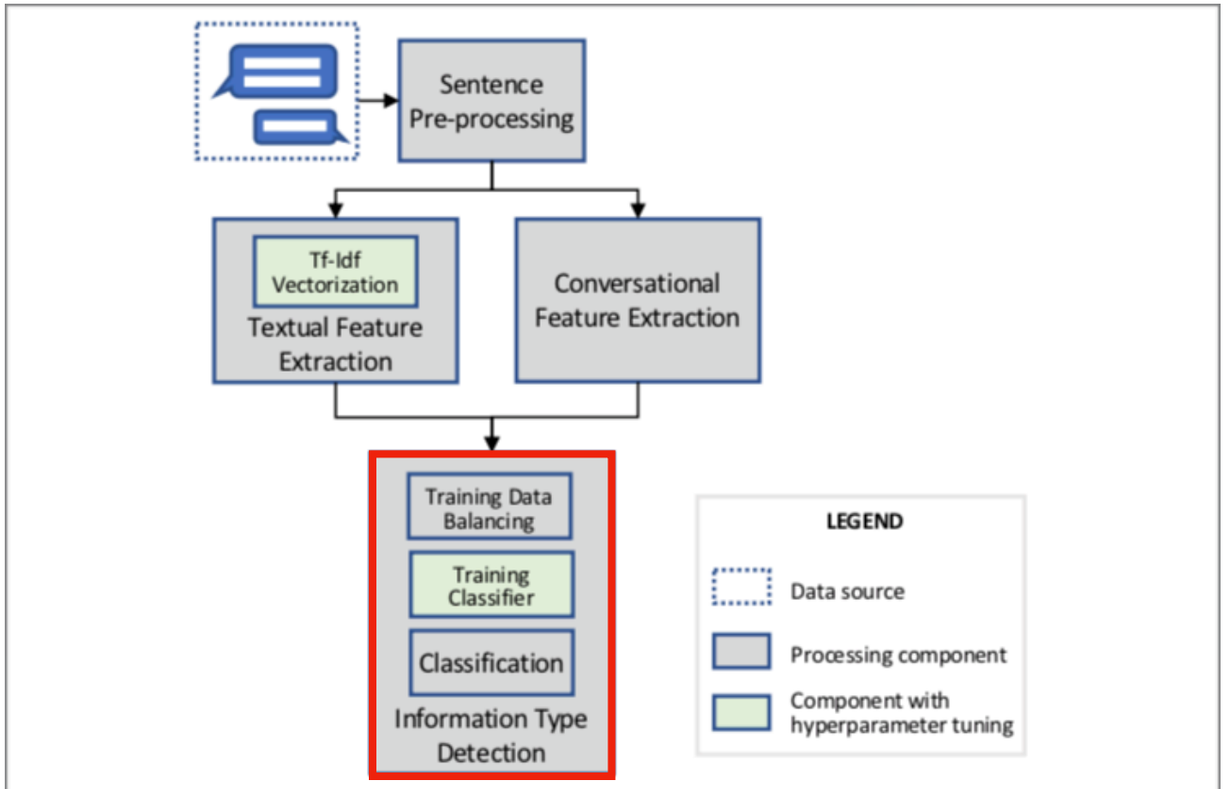


Fig. 4: The process of supervised information type detection

Evaluation of Automated Detection Methods（1/2）

- 2つのシナリオで評価
  1. 議論中の文はすでに部分的にラベル付けされており，ユーザは残りのラベル付けされた文も検索したい
  2. 他のスレッドについての知識から，新しい議論スレッド内のコメントを分類する必要がある

- シナリオ毎に適した，分類精度を評価するための検証方法を設定
  - 1. Stratified 5-fold cross validation（5分割交差検証）
  - 2. Leave-One-Issue-Out cross validation

Evaluation of Automated Detection Methods（2/2）

- 3つの独立変数からすべての可能な検出方法の組み合わせ
  1. 二つの分類モデル：LR，RF
  2. 3種類の特徴セット: Textual Features，Conversational Features, Textual + Conversational Features
  3. 2つの不均衡なデータ処理技術：Class Weight，SMOTE

TABLE III: All configurations for detecting information types

ID	Model	Feature Set	Imbalance Handling
LTC	Logistic Regression	Textual	Class Weight
LTS	Logistic Regression	Textual	SMOTE
LCC	Logistic Regression	Conversational	Class Weight
LCS	Logistic Regression	Conversational	SMOTE
LBC	Logistic Regression	Both	Class Weight
LBS	Logistic Regression	Both	SMOTE
RTC	Random Forest	Textual	Class Weight
RTS	Random Forest	Textual	SMOTE
RCC	Random Forest	Conversational	Class Weight
RCS	Random Forest	Conversational	SMOTE
RBC	Random Forest	Both	Class Weight
RBS	Random Forest	Both	SMOTE

Result RQ2（1/4）

シナリオ１：議論中の文はすでに部分的にラベル付けされており，
ユーザは残りのラベル付けされた文も検索したい

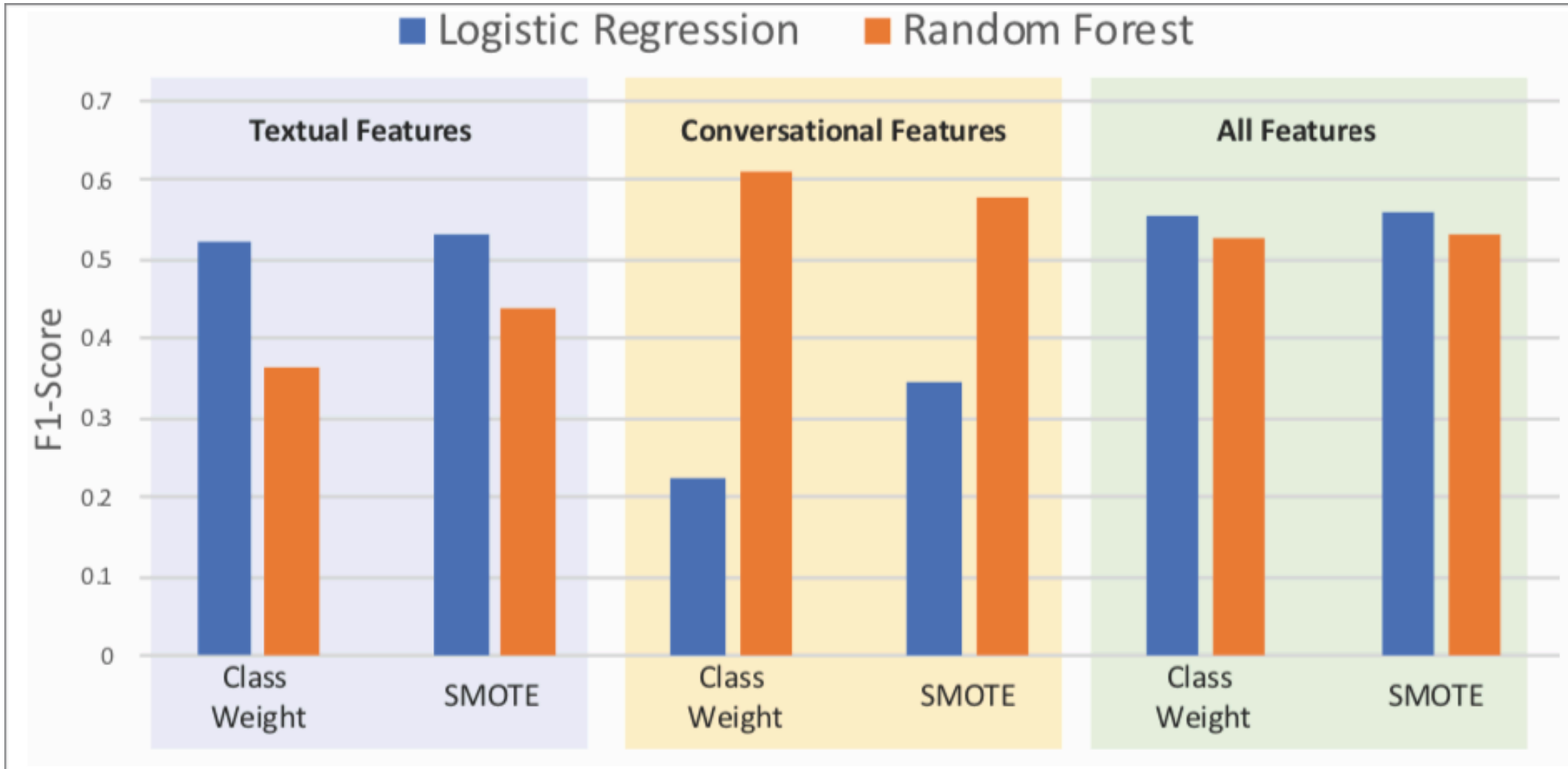


Fig. 5: Comparison of F1-scores in Scenario 1 (Stratified 5-fold cross validation)

Result RQ2（2/4）

シナリオ１：議論中の文はすでに部分的にラベル付けされており，
ユーザは残りのラベル付けされた文も検索したい

TABLE V: Detailed results for each information type in Scenario 1 with configuration RCC (Random Forest using Conversational features with class weight adjustment)

Label	Precision	Recall	F1-Score	Support
Expected Behaviour	0.42	0.28	0.33	124
Motivation	0.56	0.53	0.54	288
Observed Bug Behaviour	0.56	0.70	0.62	131
Bug Reproduction	0.53	0.47	0.50	245
Investigation and Exploration	0.60	0.65	0.62	377
Solution Discussion	0.68	0.71	0.69	1411
Contribution and Commitment	0.25	0.19	0.21	83
Task Progress	0.27	0.14	0.18	125
Potential New Issues and Requests	0.67	0.66	0.66	230
Solution Usage	0.65	0.82	0.73	368
Workarounds	0.58	0.45	0.49	89
Action on Issue	0.45	0.39	0.42	61
Social Conversation	0.63	0.62	0.63	798
Weighted average/Total	0.61	0.62	0.61	4330

Result RQ2（3/4）

シナリオ２：他のスレッドについての情報から,
新しい議論スレッド内のコメントを分類したい

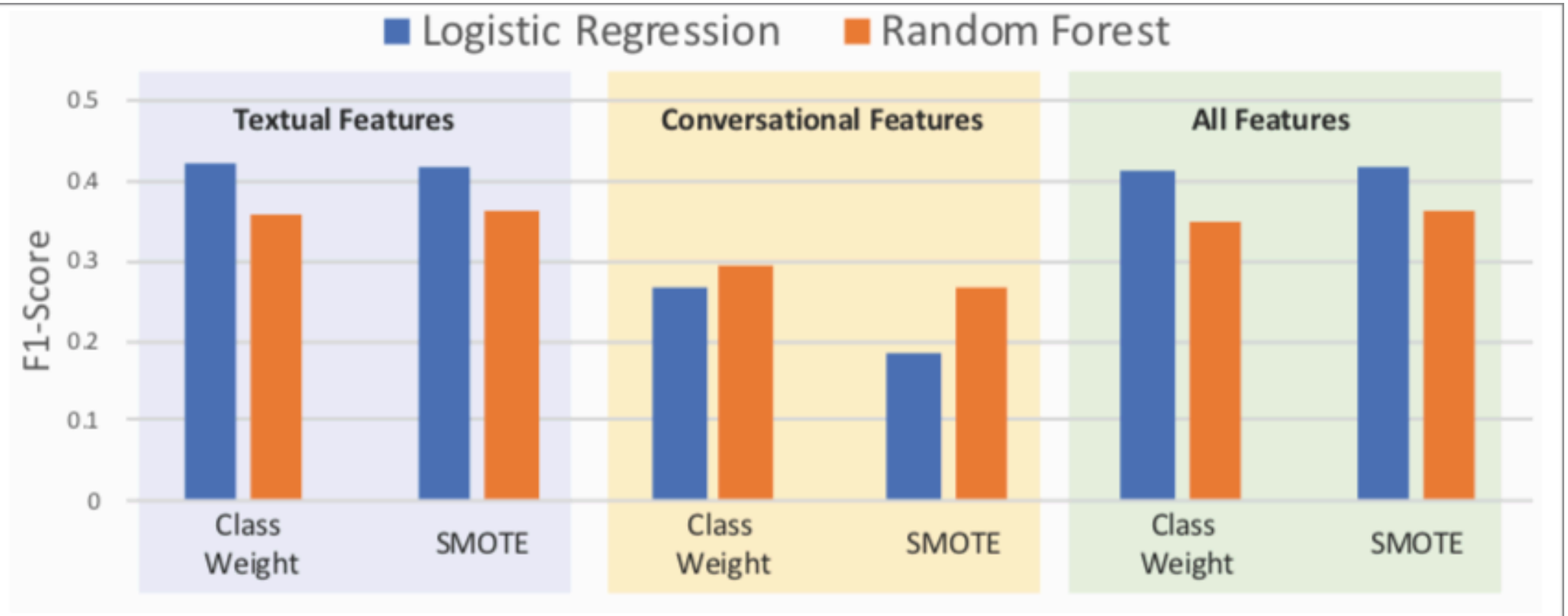


Fig. 6: Comparison of F1-scores in Scenario 2 (Leave-One-Issue-Out cross validation)

Result RQ2（4/4）

シナリオ２：他のスレッドについての情報から,
新しい議論スレッド内のコメントを分類したい

TABLE VI: Detailed results for Scenario 2 with configuration LTC (Logistic Regression using textual features with class weight adjustment)

Label	Precision	Recall	F1-Score	Support
Expected Behaviour	0.71	0.1	0.15	124
Motivation	0.44	0.1	0.13	288
Observed Bug Behaviour	0.23	0.03	0.04	131
Bug Reproduction	0.53	0.36	0.42	245
Investigation and Exploration	0.47	0.24	0.31	377
Solution Discussion	0.59	0.65	0.58	1411
Contribution and Commitment	0.51	0.31	0.37	83
Task Progress	0.35	0.26	0.29	125
Potential New Issues and Requests	0.1	0.03	0.03	230
Solution Usage	0.57	0.08	0.12	368
Work-Arounds	0.51	0.06	0.09	89
Action on Issue	0.78	0.49	0.58	61
Social Conversation	0.74	0.69	0.70	798
Weighted average/Total	0.55	0.42	0.42	4330

まとめ

- OSSのissueの議論に含まれる情報タイプを調査した
  - 16種類の情報タイプを識別した

- issueの議論の文をその情報タイプに自動的に分類する技術の使用の有効性を調査した
  - Random Forestでは，issueについての事前の情報があればConversational Featuresのみを使用してほとんどの文の種類を効果的に検出できることがわかった
  - ロジスティック回帰では，新しいissueから文を分類するときにTextual Featuresを使うと満足のいく精度を得られることがわかった

所感

- シナリオ２「新しい議論スレッド内のコメントを分類したい」場合の精度があまり高くないと思い，まだ実用性はなさそうだと思った（実現すればかなり便利？）

- 訓練データの学習方法やモデルの精度の検証方法などがたくさん出てきてそれらの知識も必要だと思った

おしまいにゃあ゛ あ゛ あ゛ あ゛ あ゛
あ゛ あ゛ あ゛ あ゛

16種類の主な情報タイプ(1/6)

- Expected Behavior（予期される動作）
  - SHはユーザ視点からissueによる影響を受けた時の予測や理想の状況を議論する
  - この議論は時に個人の考えやOSSの参加者からの意見を参考にする．例えば，「自分の提案は，ストリーミングデータ処理を行うためにスペースを残したいユーザーが，メモリ要件の変更について心配する必要がないように，近いうちに語彙を読み取り専用にするオプションを持つことです」という参加者のコメント
- Motivation（動機）
  - SHはなぜissueの修正や機能の追加が必要かを詳しく調べる
  - 議論を補強するために彼らは大抵，要求された機能を含むユースケースを説明したり，要求された機能を実装した参加者を引用した．例えば，TensorFlowの入力パイプラインの再設計のサポートをする参加者の一人：「残念なことに今の所，この方法は私のGPUを圧迫している．しかし，他のほとんどの深層学習フレームワークはどうにかしてこの方法をはるかに高性能にしている」

16種類の主な情報タイプ(2/6)

- Observed Bug Behaviour（観察されたバグの振る舞い）
  - バグ報告でのみ見られる．観察されたバグの振る舞いの描写に焦点を当てる
  - 例）「’pipe()’メソッドを使っているおかしな動作を見つけた」
    →この動作についての描写をする
- Bug Reproduction（バグの再現）
  - バグ報告で見られ，全てのレポート・リクエスト・バグの再現に関する質問に焦点を当てる
  - 例えば，バグが複製されたという一人の参加者のコメント：「ここにW10のドイツ語のテキストで起こっている同様の問題がある」
- Investigation and Exploration（研究と調査）
  - OSSのSHは，issueを起こしてきたと思われる問題についての考えの調査について議論する
  - 時々，参加者はどのように何を調査するのかを提案する．例えば，「この結果は私の仮説と一致するが，メモリの増加がそれほど重要でないということも示している．しかしそれでもライブラリの設計での潜在的な欠陥を示している」

16種類の主な情報タイプ(3/6)

- Solution Disussion（ソリューションディスカッション）
  - 開発者の視点から解決スペース周辺で形成され，その参加者は設計のアイデアを議論し，提案，内容，変更，挑戦やそのようなトピックに関する便利な参考文献同様に，実行の詳細を議論する.
  - 例）「これへの複数のアプローチ方法があることを知っているが，パフォーマンスのためにnode-gypを強く勧める」
- Contribution and Commitment（貢献とコミットメント）
  - 参加者は貢献者に呼びかけ，issueの解決に貢献するために意欲的または非意欲的かの意思表示を要求する
  - 例）一人の潜在的な貢献者「私はできる限りの貢献をしたい．しかしこれは一人では解決できない問題だ．他に2,3人興味のある人がいればすごくいいが…」
- Task Progress（タスクの進捗）
  - SHはissueの解決に向けたタスクやサブタスクの進捗の要求や報告をする
  - 参加者は時々彼らの活動計画を述べる
  - 例）私は最初の挑戦をする．それは文字列バージョンをnodejsに取得するコンセプトの証明である．swigインターフェースの追加に取り掛かかるだろう

16種類の主な情報タイプ(4/6)

- Testing（テスト）
  - 参加者はテストに関するシステムの環境，コード，データやフィードバック同様，テストの手続きについて議論する
  - 例）「'0.101'と'master'でテストされたものについて - issueはサンプル文書だけでなく，コーパス全体として'master'で修正されているようだ」
- Future Plan（未来の計画）
  - 参加者はissueに関する長期間の計画について議論する：それらは大抵，最新のissueを解決するためには要求されない作業や考えに関係している
  - 例）「将来のため，この方向性に関する何かのプログラムを試作しているためしばらくお待ちください」
- Potential New Issue and Requests（潜在的な新たなissueと要求）
  - 参加者は，最新のissueを調査したり対処している一方で，新たなバグや必要な機能を明確にしたり議論している．それらは最新のissueの議論の範囲外だが，新たなissue報告に誘導するかもしれない
  - 例）process hangingを起こすscikit-learnの並列処理のバグについて議論している時，一人の参加者が言う：「この0.14にはないがサイドポイントとして，マスター上により多くのjoblib並列処理があるようだ」

16種類の主な情報タイプ(5/6)

- Solution Usage（ソリューションユーセッジ）
  - 通常，問題の完全または部分的な解決策が一度公開され，SHが新しい解決策の更新でライブラリを使用する方法について質問をしたり，提案を提供したときに議論される
  - 例）「[新しいリリースで]モデルの学習を続ける方法を教えてください」
- Workarounds（ワークアラウンド（一時的な解決方法））
  - 公式の修正や改良がリリースされるまで，issueに対処するための一時的または代わりとなる解決策について議論する.
  - 例）ストリーミングデータのメモリ増加に関する議論での一時的な解決策：「現在のところ，nlpオブジェクトの再ロード/収集に関する回避策は，本番環境では問題なく動作します」
- Issue Content Management（問題コンテンツ管理）
  - 議論の方向性の変更やissueを中心としたコメントの質の管理に焦点を当てる
  - 例）「この議論をこのリンク先の議論へと移動させたい」

16種類の主な情報タイプ(6/6)

- Action on Issue（issueに対する行動）
  - 参加者は，issue自体に対して実行する適切な行動についてコメントする
  - 例）「このissueは古く，情報の大部分も古くなっているのでこの問題を解決します」
- Social Conversation（社会的な会話）
  - 賞賛，失望，困惑や後悔といった感情を表したりちょっとした会話をする
  - 例）「私はこれが多くの考えと注意を引きつけたことを嬉しく思います！」