УДК 519.68; 005.334; 007.51/.52
DOI: 10.20535/SRIT.2308-8893.2020.2.09

АНАЛІЗ РИЗИКІВ ПРОЕКТУ ЗА ДОПОМОГОЮ ТЕКСТОВОГО ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ КОМЕНТАРІВ У СИСТЕМІ УПРАВЛІННЯ ПРОЕКТАМИ JIRA

А.А. ЛЄДНІКОВА, Д.В. ШИПІК, П.І. БІДЮК

Анотація. У ході дослідження розроблено методологію та створено програмний продукт для визначення ризиків проекту на базі комунікації розробників, подано результати роботи програми на даних реального проекту CASSANDRA компанії Apache Software Foundation. Методологію реалізовано на основі вже відомих алгоритмів визначення емоційних складових у тексті VAD та матричних методів аналізу ризиків проекту з використанням власних розробок, що дозволять об'єднати ці різні підходи. Визначення назви потенційних ризиків визначається за допомогою моделі побудови тем LDA. Отримані результати допоможуть визначати важливість задачі відповідно до комунікацій та ранжувати їх у середині проекту за важливістю та потреби додаткової уваги, що в контексті продукту дасть змогу менеджерам проекту більш швидко розуміти та вирішувати проблеми.

Ключові слова: аналіз ризиків проекту, імовірнісний латентний семантичний аналіз, модель латентного розподілу Диріхле, оброблення природної мови, аналіз тональності тексту.

ВСТУП

Сфера інформаційних технологій (IT) розвивається дуже швидко і стрімко, так само як і її представники. Компанії динамічно зростають і дедалі більше стають децентралізованими. Але чим більші компанія та команда, тим більше потреб у її менеджменті. Agile Software Development (ASD) стає найпоширенішою технікою управління проектами: організації шукають способи бути більш гнучкими, тоді як 71% організацій уже повідомляють про використання цих підходів для своїх проектів [1].

В управлінні проектами є декілька складових, одна з яких найбільш важлива і водночас найбільш трудомістка, — це аналіз ризиків, метою якого є забезпечення виконання завдань проекту в певний час і за наявності певної кількості ресурсів. Такий тип управління повинен здійснюватися протягом усього часу існування проекту.

Для моніторингу поточного стану ризиків можна визначити показники ризиків. Ці показники можуть бути кількісними (імовірність витрати на контрольні заходи) або якісні (оцінка мотивації персоналу проекту). Інший метод моніторингу полягає у використанні тригерів як порогових значень для показників, які запускають заходи, коли їх досягнуто.

Найпопулярнішим інструментом за кількістю користувачів (понад 50 мільйонів користувачів) для управління IT проектами натепер є Jira. Це система для відстеження помилок і проблем, яка надає спільне середовище, де члени команди можуть подавати і обговорювати питання, потребувати поради і ділитися думками, корисними для заходів з підтримання або дизай-

нерських рішень [2, 3]. Зазвичай проекти приватні, але деякі компанії, які надають безкоштовні програмні продукти, мають відкриті репозиторії, завдяки яким кожен користувач має змогу відстежувати стан проекту, створювати завдання, коментувати та допомагати у їх розробленні. Прикладом такої компанії є Apache Software Foundation [4].

Згідно з Atlassian, найбільшою проблемою, з якою стикаються команди сьогодні, — це спілкування [5]. Коли роботу в команді виконано правильно, переваги очевидні:

- 50% більш мотивовані успіхом команди, ніж компанії (27%), або індивідуальним успіхом (23%);
- 43% вважають, що вони мають великий особистий вплив на місію своєї основної команди проти 33% на місію компанії в цілому;
- 56% почувають впевненіше працювати в команді, ніж індивідуально.

Оскільки інженерія програмного забезпечення є інтенсивною діяльністю людського капіталу, важливість управління емоціями у професії програмного забезпечення очевидна. Емоції є ключовими проблемами в поведінці людей [6]. Чим більше методологія бере до уваги людські фактори, тим успішнішою вона стає в реальному світі. Це тому, що людські та соціальні фактори справляють сильний вплив на успіх розроблення програмного забезпечення та остаточної системи [7].

У контексті системи Jira дані коментарів та журналів завдань є цінною інформацією для визначення поточного стану розроблення. Аналізуючи їх на предмет висловлюваних емоцій або певних ключових слів, можливо створити матрицю ризиків для виявлення проблем, пов’язаних з людським фактором та комунікаціями [2]. Побудова тематичних моделей дає змогу дати швидке та чітке уявлення про сутність проблеми. Використання такого підходу до історичних даних допоможе отримати цінні уроки минулого для більш ефективної роботи у майбутньому.

Одним з перших досліджень у напрямі визначення емоцій розробників, а не їх поведінки, було проведено A. Murgia, P. Tourani, B. Adams, M. Ortu, яке порушило питання про відсутність досліджень у цій галузі [2]. Автори позначили коментарі як «одне повідомлення – одна емоція», використовуючи рамки Парротта (любов, радість, здивування, гнів, смуток, страх), щоб виміряти людську згоду щодо їх присутності у звітах про проблеми.

Дослідження, проведене M. Ortu, G. Destefanis, B. Adams та ін., також показало, що «коментарі розробників містять не тільки технічну інформацію, але й цінну інформацію про почуття та емоції» [8]. R. Jongeling [9] використовував цей набір даних, щоб перевірити, чи інструменти для навчання машин для емоцій, отримані від соціальних даних, узгоджуються з даними, позначеними вручну. Пізніше наявність цього сховища спонукала до подальших досліджень та експериментів з емоцій розробників: виявлення вигорання та продуктивності [10, 11], вимірювання афективності та ефективності [10], моделювання напряму емоції гніву та аналізу ввічливості.

M. Mäntylä, B. Adams, G. Destefanis та ін. [10] використовували цей набір даних у зв’язку з VAD-лексиконом (Valence, Arousal, Dominance) з 13 915 англійських слів [12] для аналізу VAD у звітах про завдання, оскільки вони вважають, що «використання вимірного підходу більш вигідне ніж використання дискретного підходу, оскільки розмір може залежати від вигорання і продуктивності».

Тематичне моделювання є одним з найпопулярніших імовірнісних алгоритмів кластеризації, який останнім часом набуває дедалі більшої уваги. Основною ідеєю моделювання тематики є створення імовірнісної генеративної моделі для корпусу текстових документів. У тематичних моделях документи являють собою сукупність (суміш) тем, де тема — розподіл імовірностей над словами.

Дві основні моделі теми — імовірнісний латентний семантичний аналіз (pLSA) [13] і модель латентного розподілу Діріхле (LDA) [14]. T. Hofmann [13] увів pLSA для моделювання документів, але вона не забезпечує жодної ймовірнісної моделі на рівні документа, що ускладнює її узагальнення для моделювання нових невідомих документів. D. Blei, A. Ng, M. Jordan розширили її, увівши розподіл Діріхле як ваг суміші тем для кожного документа і назвали її моделлю латентного розподілу Діріхле (LDA) [14].

Модель латентного розподілу Діріхле є сучасною технікою «без учителя» для вилучення тематичної інформації (тем) зі збірника документів. Основна ідея полягає в тому, що документи подані у вигляді випадкової суміші прихованих тем, де кожна тема є розподілом імовірностей над словами.

E. Guzman у своїй праці [15] описала прототип візуалізації, який подає огляд емоційного клімату проекту на основі текстової інформації, як-от пошти та артефакти. Вона складається з двох основних частин: вилучення емоцій із Senti Strength і моделювання тематики з LDA. Перший виражений в кольорах (зелений — позитивний, жовтий — нейтральний і пурпуровий — негативний) і розмір кола, а другий — у хмарах слів.

Дослідження присвячено вивченню емоцій, виражених у системі відстеження проблем Apache Jira для їх перетворення у ризики та надання автоматизованих інструментів для подальшого можливого розроблення системи підтримання прийняття рішень для управління проектним ризиком, пов’язаних з людськими і соціальними факторами.

ПОСТАНОВКА ЗАВДАННЯ

Мета дослідження:

1) дослідження ризиків проектів сфери IT та методів їх виявлення;

2) дослідження існуючих методів та алгоритмів для інтелектуального аналізу тексту на предмет тригерів ризиків;

3) розроблення методології використання інтелектуального аналізу тексту для ідентифікації та аналізу ризиків проекту;

4) розроблення програмного забезпечення для проведення експериментів за даною методологією;

5) аналіз результатів та рекомендації щодо подальших досліджень.

ВИКЛАДЕННЯ ТЕОРІЇ

Визначення емоційних показників. Спочатку кожний коментар позбавляється від пунктуації та чисел, а потім розбивається на токени, у результаті отримаємо подання у вигляді «мішка слів» (Bag-Of-Words). Далі лематизуємо кожний токен, тобто зводимо його до початкової форми.

Для кожного токену коментаря визначаємо показники валентності, збудження і домінування (ВЗД) за таблицею оцінок A. Warriner [12] з 13 915 англійських слів, залишаючи для подальшого розрахунку лише максимальні та мінімальні значення показників з усіх слів.

Перед прикладом розглянемо значення кожної зі складових [10].

Валентність (Valence) — це емоційний вимір, пов'язаний з привабливістю (або несприятливістю) події, об'єкта або ситуації. Термін означає напрям поведінкової активації до стимулу (апетитною мотивацією) або відхилення від нього (аверсивною мотивацією).

Збудження (Arousal) — це розмірність, що вказує рівень емоційної активації. Вона має різні фізіологічні та психологічні реакції, наприклад, підвищену частоту серцевих скорочень і настороженість до відповідей і сприймається як відчуття реактивності до подразників та психічного збудження. Збудження також посилює задоволення або невдоволення, що описується валентним виміром, наприклад, розчарування може змінити гнів, а мирне щастя може змінитися в заховлення, коли збудження збільшується.

Домінантність (Dominance) являє собою зміну відчуття контролю над стимулом (або ситуацією).

Розглянемо декілька прикладів з табл. 1. Кохання та радість мають більшу валентність як позитивні емоції, смуток — більш пасивну природу, тож отримує низькі показники збудження та домінування.

Таблиця 1. Подання слів у просторі ВЗД

Слово	Валентність	Збудження	Домінування
Anger / гнів	2,50	5,93	5,14
Joy / радість	8,21	5,55	7,00
Sadness / смуток	2,40	2,81	3,84
Love / кохання	8,00	5,36	5,92
Середнє	5,06	4,21	5,19

Для особливих випадків, коли максимум (max) нижчий від середнього значення або коли мінімум (min) вищий, установлюємо max або min до середнього значення всіх слів лексики. Далі використовуємо дані значення для розрахунку відносних показників за такою формулою:

$$Range(\overline{\omega}) = \begin{cases} \max(\overline{\omega}) - avg(\overline{W}), \ if \ \min(\overline{\omega}) > avg(\overline{W}), \\ avg(\overline{W}) - \min(\overline{\omega}), \ if \ \max(\overline{\omega}) < avg(\overline{W}), \\ \max(\overline{\omega}) - \min(\overline{\omega}), \ if \ \min(\overline{\omega}) \le avg(\overline{W}) < \max(\overline{\omega}). \end{cases} \quad (1)$$

Наприклад, якщо коментар буде містити всі слова, наведені в табл. 1, він отримує оцінку валентності 5,81 (8,21–2,40, третій випадок у формулі). Чим вище значення, тим більш екстремальні бали ВЗД.

Таким чином, дані показники визначають значущість наявності цих емоційних станів і ступінь їх відмінності від середніх значень.

Аналіз ризиків задачі. Задача може мати багато коментарів, включаючи й негативні, але якщо останні позитивні та вирішують проблему, то їх вага повинна бути більшою від попередніх.

Таким чином за точку відліку можна взяти час першого коментаря (0), а за верхню межу (1) — поточний час, нормування часу коментаря у цьому проміжку дає вагу актуальності коментаря.

Таким чином для кожного коментаря маємо: валентність, збудження, домінування, актуальність. Для задачі отримаємо зважені оцінки:

$$S = \frac{1}{n}\sum_{i=1}^{n} w_i s_i , \qquad (2)$$

де S — загальна оцінка задачі (валентність / збудження / домінування); S_i — оцінка (валентність / збудження / домінування) коментаря; w_i — актуальність коментаря.

Перехід до матриці ризиків (рис. 1). Таблиці оцінки ризиків дають змогу організаторам подій розподіляти рейтинги ризиків на всі небезпеки, щоб вони могли визначати пріоритети та систематично вирішувати їх.

	CONSEQUENCE				
LIKELIHOOD*	Insignificant 1	Minor 2	Moderate 3	Major 4	Catastrophe 5
A (Almost certain)	H	H	E	E	E
B (Likely)	M	H	H	E	E
C (Possible)	L	M	H	E	E
D (Unlikely)	L	L	M	H	E
E (Rare)	L	L	M	H	H

Рис. 1. Матриця ризиків

За даною таблицею маємо такі типи ризиків:

- E = екстремальний: необхідні негайні дії;
- H = високий ризик: необхідна увага старшого керівництва;
- M = помірний: відповідальність керівництва має бути визначена;
- L = низький: управління за допомогою рутинних процедур.

Кожен тип ризику є результатом поєднання двох його властивостей — імовірності (табл. 2) та значущості наслідків (табл. 3).

Таблиця 2. Імовірність наслідків

Рівень	Значення	Опис	Приклад детального опису події
A	5	Безумовно	Очікується, що це відбудеться в більшості випадків
B	4	Імовірно	Імовірно, відбудеться в більшості випадків
C	3	Можливо	Може відбутися у певний час
D	2	Непевно	Може статися через деякий час
E	1	Рідко	Може відбуватися, але тільки за виняткових обставин

Рівні значущості та відповідні пріоритети, що використовуються в JIRA, наведено в табл. 3.

Таким чином, інтегрований показник ВЗД може бути вірогідністю, у той час як важливість зазвичай визначається менеджером.

Таблиця 3. Значущість наслідків

Рівень	Опис	Тип Jira	Приклад детального опису
1	Мізерна	Trivial	Косметична проблема, як помилкові слова або змішаний текст
2	Незначна	Minor	Незначна втрата функції або інші незначні проблеми
3	Помірна	Major	Велика втрата функціональності
4	Значна	Critical	Аварії, утрата даних
5	Катастрофа	Blocker	Блокує розроблення та / або роботу з тестування, процедура виконання проєкту не працює

Також важливість може бути збагачена вагами для типу завдання (баг, фікс), тривалістю виконання завдання (чим довше, тим гірше), а найголовніше — структурою проекту, тобто скільки завдань може бути в очікуванні через поточне, скільки виконавців поточних та супутніх завдань.

Імовірність ризику оцінюємо за табл. 4.

Таблиця 4. Імовірність як інтегрований показник ВЗД

Рівень	Значення	Опис	Інтегрований показник ВЗД
A	5	Безумовно	≥ 10
B	4	Імовірно	(10, 8]
C	3	Можливо	(7, 5]
D	2	Непевно	(5, 2]
E	1	Рідко	< 2

Визначаємо загальну оцінку ризику як добуток імовірності на значущість.

$$VR = A \cdot q\,, \tag{3}$$

де VR — важливість ризику; A — загроза (наслідок, дія) ризику (небажаної події); q — імовірність її настання.

Визначення теми ризику. Для швидкого оцінювання ситуації потрібно розуміти, що саме відбулося не так. У цьому можуть допомогти кілька ключових слів або тематика проблеми.

Одним з найбільш поширених методів побудови тематичних моделей є Latent Dirichlet Allocation (LDA), що моделює документ як розподіл тем і тему як розподіл слів. Тут документ — це коментар.

Розглянемо ігрову модель LDA, що виробляє такі теми:

Тема 0: ‘0,075*"patch" + 0,040*"fix" + 0,039*"cassandra_num_" + +0,020*"trunk" + 0,020*"attach" + 0,017*"v_num_" + 0,015*"apply" + +0,013*"change" + 0,011*"version" + 0,011*"issue"‘.

Тема 1: ‘0,018*"thrift" + 0,017*"table" + 0,016*"make" + 0,015*"change" + 0,014*"cql_num_" + 0,014*"use" + 0,013*"patch" + 0,013*"bq" + +0,012*"would" + 0,012*"think"‘.

Тема 2: ‘0,043*"cql" + 0,028*"id" + 0,020*"select" + +0,020*"_num_e_num_" + 0,016*"make" + 0,016*"would" + 0,010*"loop" + +0,009*"pprop" + 0,009*"eentid" + 0,009*"python"‘.

Тема 3: ‘0,036*"flush" + 0,017*"write" + 0,017*"call" + 0,012*"memtable"+ + 0,012*"segment" + 0,011*"thread" + 0,011*"replay" + 0,011*"get" + +0,010*"new" + 0,009*"need"‘

Тоді подання речення “moves strategy creation into Table instantiation so it can’t be out of sync” в цьому просторі буде [(0; 0,1), (1; 0,50), (2; 0,16), (3; 0,24)].

Алгоритм LDA ґрунтуються на попередньому розподілі Діріхле і передбачає модель «мішок слів» — модель для аналізу текстів, яка враховує тільки частоту слів, але не їх порядок. Ця модель добре підходить для тематичного моделювання, оскільки вона дозволяє виявляти неявні зв’язки між словами. Метод LDA виконує м’яку кластеризацію і припускає, що кожне слово у реченні генерується деякою прихованою темою, яка визначається розподілом імовірностей на множині всіх слів тексту.

Маючи корпус D, що складається з M документів, для документа d, що має N_d слів ($d \in \{1,\ldots,M\}$), LDA моделює D згідно з таким генеративним процесом [14]:

1) вибір поліноміального розподілу φ_t для теми t ($t \in \{1,\ldots,T\}$) з розподілу Діріхле з параметром β;

2) вибір поліноміального розподілу θ_d для документа d ($d \in \{1,\ldots,M\}$) з розподілу Діріхле з параметром α;

3) для кожного слова w_n ($n \in \{1,\ldots,N_d\}$) у документі d:

а) вибрати тему z_n з θ_d;

б) вибрати слово w_n з z_n.

У згаданому генеративному процесі слова в документах — єдині спостережувані змінні, тоді як інші — латентні змінні (θ) і гіперпараметри (α і β). Для того щоб зробити висновок про приховані змінні і гіперпараметри, імовірність спостережуваних даних D обчислюється і максимізується таким чином:

$$P(D \mid \alpha,\beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha)\left(\sum_{n=1}^{N_d} p(z_{dn} \mid \theta_d)\, p(\omega_{dn} \mid z_{dn},\varphi) P(\varphi \mid \beta)\right) d\theta_d\, d\varphi .$$

Унаслідок зв’язку між θ і φ у підінтегральній функції у формулі (3), точний висновок у LDA є нерозв’язним. Різні наближувальні алгоритми, такі як варіаційний висновок або ланцюг Маркова Монте-Карло (MCMC), зазвичай використовуються для виведення в LDA.

У цій роботі будемо використовувати пакет gensim, який має реалізацію online LDA. Цей алгоритм використовує стохастичну оптимізацію, щоб максимізувати варіаційну цільову функцію для моделі тематичного розподілу прихованих Діріхле (LDA). Він тільки оцінює підмножину загального корпусу документів кожної ітерації і тим самим здатний швидко знайти локально оптимальне налаштування варіаційного апостера над темами [16].

Для визначення теми агрегуємо всі коментарі та застосуємо модель LDA, роблячи припущення, що кількість тем відповідає кількості задач.

Для оцінювання моделей теми використовується когерентність теми, для якої існує дві основні метрики — C_v і C_{umass}.

Метрика C_v базується на ковзному вікні, однокомпонентній сегментації топ-слів і непрямій мірі підтвердження, що використовує нормалізовану точкову взаємну інформацію (NPMI) і схожість за косинусом. Ця міра когерентності отримує кількість зустрічання слів для заданих слів за допомогою

ковзного вікна і розміру вікна 110. Підрахунки використовуються для обчислення NPMI кожного топ-слова для кожного іншого топ-слова таким чином, що створює набір векторів для кожного топ-слова:

$$\mathrm{NPMI}(w,w) = \frac{\log \dfrac{P(w_i, w_j) + e}{P(w_j)}}{\log(P(w_i, w_j) + e)}.$$

Однокомпонентна сегментація топ-слів потребує розрахунку подібності вектора кожного топ-слова і суми векторів усіх топ-слів. Як міра подібності використовується косинус. Когерентність — це середнє арифметичне з цих подібностей [17].

Метрику C_{umass} запропонував D. Mimno et al. [18]. Ця метрика бере до уваги упорядкування серед топ-слів теми і має вигляд

$$C_{umass} = \frac{2}{N*(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j) + e}{P(w_j)},$$

де N — кількість топ-слів, узятих для аналізу.

Оскільки в будуванні моделі враховується безліч коментарів, то для визначення теми документа топ-словами можуть бути слова, не притаманні даній проблемі, але дуже близькі. Для цього краще виконати перетин слів і поза вагою слова у теми врахувати вагу (імовірність) самої теми. Тоді алгоритм визначення теми ризиків задачі виглядатиме так:

1) Створити порожню таблицю T для слів і ваг задачі.
2) Для кожного коментаря C_i обраної задачі:

- визначити список слів W коментаря;
- визначити теми T коментаря;
- для кожної теми T_i та ваги цієї теми *topic_weight* для коментаря C_i:

визначити топ-N слів W_i^t з вагами *word_weigh* кожного слова;

якщо слово наявне в коментарі, то T [*word*] += *word_weigh* * *topic_weight.*

Найбільш зручним поданням теми є хмара слів (word clouds) (рис. 2). Оскільки після моделювання отримаємо набір пар слово–вага, то можемо створити зображення, де розмір слова буде пропорційний його вазі.

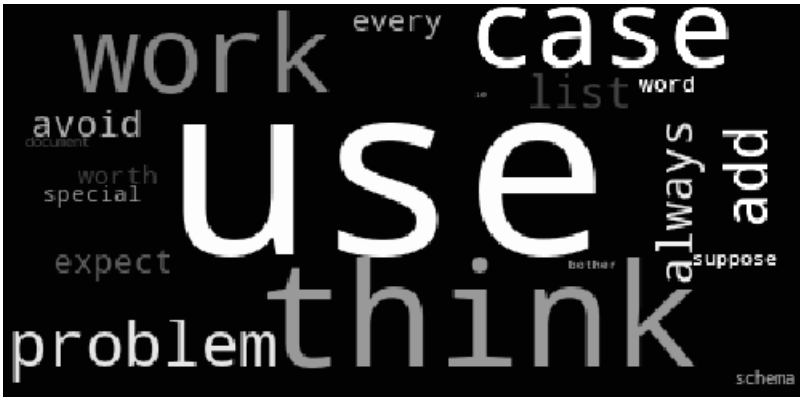

Рис. 2. Приклад хмари слів

ПРИКЛАДИ ЗАСТОСУВАННЯ ТЕОРІЇ

Датасет jira_emotion [19] являє собою набір даних, витягнутих з Jira ITS чотирьох популярних екосистем з відкритим вихідним кодом (а також інструментів та інфраструктури, що використовуються для здобування інформації) спільноти Apache Software Foundation, Spring, JBoss та CodeHaus. Повний набір даних (включаючи проекти Apache) містить 3516 завдань та 25306 коментарів 1375 авторів. У розгляданому випадку використано лише поля *comment, issue_report_id* та *updateDate* з таблиці *jira_issue_comment*, а також поля *id, priority* та *project* з таблиці *jira_issue_report*. Як приклад було обрано достатньо відомий проект CASSANDRA компанії Apache Software Foundation. Тож у подальших розділах розглядаються задачі та коментарі, що стосуються лише цього проекту. Усього 41966 коментарів у 6271 задачах з 2009-03-07 по 2013-12-18.

Після обчислень для кожної задачі маємо:

- імовірність як зважена сума ВЗД коментарів;
- значущість, що визначена менеджером проекту, у цьому випадку поле *priority* з таблиці *jira_issue_report*;
- ключові слова та ваги, з яких можна отримати хмару слів.

Приклади коментарів з найвищими та найнижчими оцінками подано у вигляді рис. 3. Візьмемо перший запис — задачу 333428; вона має лише один коментар такого змісту зі значенням ВЗД 4,62; 3,26; 3,51:

- default_validation_class means "all data that isn’t explicitly in column_metadata conforms to this data type." So you’ve violated that. You have two options:
- set d_v_c to BytesType (the default);
- leave the column definition alone, but only drop the index part (maybe this is what you were trying to do, but you changed from "colour" to "color").

Id	integra_value	integra_value_weighed	llikelIhood	priority	priority_value	rate
333428	11,390000	11,390000	5	Blocker	5	25
329678	11,130000	11,130000	5	Blocker	5	25
333669	10,755556	10,737776	5	Blocker	5	25
331283	10,720000	10,720000	5	Blocker	5	25
329510	10,184286	10,184142	5	Blocker	5	25
333505	11,730000	11,730000	5	Critical	4	20
333436	11,518667	11,514989	5	Critical	4	20
330706	10,75000	10,750000	5	Critical	4	20

Рис. 3. Кінцевий показник та імовірність для розв’язання задач

More generally, note that best practice is to only use d_v_c in CFs with dynamic column names. I.e., if you know what the columns are going to be in the CF ahead of time as you do here, you shouldn’t use d_v_c.

Автор надав розгорнуту відповідь, але з контексту не зрозуміло, чи вирішує він цю проблему, чи ні. Особливо беручи до уваги речення “So

you've violated that" (Так, що ви порушили це); задача потребує уваги, тож отримане маркування має сенс. Розглянемо ще декілька прикладів (рис. 4).

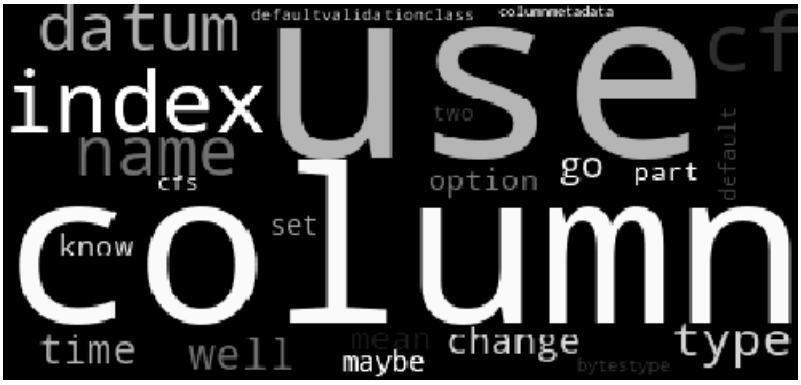

Рис. 4. Приклад маркування

Розглянемо коментарі задачі 329510 (табл. 5). Їх досить багато і лише знаявності логів помилки можна зрозуміти, що проблема є і вона потребує вирішення.

Таблиця 5. Коментарі задачі 329510

Коментар	ВЗД	∑
Those fail for me too, but that should be an easy bisect	[5,14; 1,68; 2,05]	8,87
Looking at TokenMedataTest, it was just assuming the last test in the file was actually running last and apparently that wasn't happening on my box. Ninja-fixed that one in commit 0a5a766 to not depend on the tests execution order	[4,13; 3,33; 3,45]	10,91
Looks pretty similar to what I did in 7de6f9666 to fix them in 1.1, except I used the shotgun method :) So if that fixes it, then +1	[4,2; 3,12; 1,8]	9,12
Alright, ScrubTest was another instance of tests expecting to run in a particular order (don't know why my box don't run them in the order they are declared but well, expecting a particular order is a bad idea in any case) so ninja-fixed that. I've also committed the 2 patches attached above. This fixes the failure I'm saying, except for the pig tests, but as those are clearly a setup thing I don't want to block 1.2.12 for that and I've open CASSANDRA-6376 to deal with them. Closing this issue	[5,74; 2,94; 3,97]	12,65
Btw, also got all pig tests to fail with the following exceptions:{noformat} [junit] Testcase: org.apache.cassandra.pig.CqlTableDataTypeTest: [junit] <error log> I wouldn't block a release because of pig tests, but if it does not just fail for me, it would be nice to fix it too	[4,62; 1,97; 2,66]	9,25
For LeaveAndBootstrapTest, this bisects to CASSANDRA-6244. So I think this is just a case of "we've made things asynchronous so we now check the expected result before the computation is done". Tried adding a few calls to PRCS.blockUntilFinished in the few places that were failing for me and that seems to fix the test. Attaching the resulting patch. [~brandon.williams] can you check it's not entirely stupid?	[4,35; 2,66; 2,76]	9,77
Regarding the ConcurrentModificationException, it seems that the only reason this could get triggered is due to TMD.clearUnsafe(). As this is called by tests, this is not a real problem, but what about making it grab the writeLock like any good citizen to avoid getting scarry stack traces (and don't discard a real bug later on because we've grown used to discarding such stack)? Attaching patch to do that	[4,57; 3,46; 2,69]	10,72

Читати всі коментарі досить важко, особливо, якщо зважати, що вони суто технічні та наповнені термінологією. Але завдяки інструменту побудови назви ризику можна швидко зрозуміти, що проблеми спричинені помилками з junit, patch і можливо іншою задачею проекту (рис. 5).

Рис. 5. Приклад помилки з Unit.patch

Розглянемо декілька задач зі значущістю 1 (табл. 6). Із тексту коментарів бачимо, що задачі дійсно не потребують додаткової уваги.

Таблиця 6. Коментарі задач з низькою значущістю

Задача	Коментар	ВЗД	∑
331618	bah, just realised you can use comparator= ‘Composite Type(UTF8Type, UTF8Type)’	[0,116; 0,021; 0,185]	0,322
332691	duplicate of CASSANDRA-3164	[0,364; 0,289; 0,315]	0,968
330393	Resolving now that it’s in trunk	[0,044; 0,701; 0,275]	1,02
333721	done as part of CASSANDRA-2521	[0,294; 0,851; 0,025]	1,17
329561	{{ECHO OFF}}	[0,136; 0,601; 0,595]	1,33
	(this should be ninja-d)	[0,076; 1,399; 0,045]	1,52

Розглянемо, як розподілені задачі залежно від значущості. Як бачимо з рис. 5, 6 коментарів, які дійсно потребують уваги, небагато, то їх важливо відокремити від інших, щоб швидше реагувати.

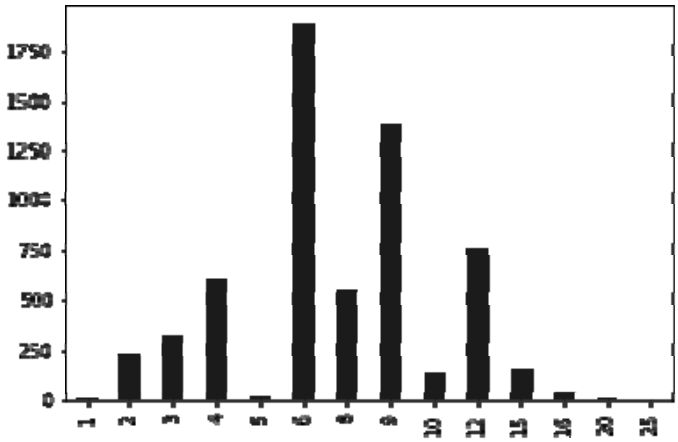

Рис. 6. Частоти появи ризиків

Розглянемо приклад задачі 334028 з найбільш частотною значущістю ризику 6, коментарі якої наведено в табл. 7 Як бачимо зі змісту коментарів, проблема була, але вона успішно вирішена і більше не потребує уваги. Однак перші два коментарі мають великі значення інтегрованого показника, і тоді задача мала б ризик великої ймовірності, а отже, і значущості.

Таблиця 7. Коментарі задачі 334028

Коментар	ВЗД	∑
I did a bit more tests and here are some results which might help: 1. JMX port set to 9090 in cassandra-env.sh 2. On the machine where another service running on 8080 we get exception above 3. On the machine where no service running on 8080 we don't get any exception and MX4J runs on port 9090 Seems like something checks for port 8080 even though it is configured to run on 9090	[3,92; 2,29; 2,83]	9,04
I think there's a confusion. There are two ports in business, one is the JMX port (default is 8080) and one is the MX4J port (default 8081) If the JMX port is used when cassandra starts you see the following exception, which is different from what's pasted in this bug report: <error log> So the problem in this case. I believe was that mx4j's port was bound to a different process To control the port used by mx4j use -Dmx4jport=8082. See https:// issues.apache.org/jira/browse/CASSANDRA-1068 for more details. I think this is not a bug and recommend to close it as such I will, however, attach a patch for trunk to make this more obvious and add -Dmx4jport=8081 to conf/cassandra-env.sh	[5,05; 4,17; 4,06]	13,28
Patch that adds the variables MX4J_ADDRESS and MX4J_PORT to conf/cassandra-env.sh make configuration for mx4j obvious	[1,65; 1,371; 2,39]	5,411
You right Ran, I checked this machine again and I have another service listening on 8081 For some reason I thought that MX4J uses same port With config options we can close it now	[2,32; 2,05; 2,49]	6,86
committed, thanks!	[2,706; 0,881; 1,705]	5,292

Якщо побудуємо хмару слів, то серед найважливіших побачимо "thank" (дякую), що також може бути натяком на розв'язану проблему (рис. 7).

Рис. 7. Хмара слів для "thank"

АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Із наданих прикладів можна бачити, що використання лише цих трьох емоційних складових достатньо для ранжування коментарів та задач залежно від наявності потенційних проблем та напруження у коментарях.

Надалі можна розглянути та застосувати інші емоційні фреймворки та показники, наприклад, визначати рівень ввічливості або агресивності. Також можна застосувати сентиментальний аналіз і використати його оцінку як окремий вимір.

Із погляду семантики поліпшити результати можна за рахунок особливостей сучасної комунікації, а саме: сленгу, смайлів, які можуть знижувати або підвищувати показник ризику, позначення типу «+1» повинні мати позитивні властивості. Щодо структури текстів, то варто дослідити вплив довжини коментарю на значення результуючого показника.

Оскільки назва ризику задачі формується зі слів, що наявні в коментарях, маємо релевантні результати, проте можна дослідити зміну відображення назви у разі зміни кількості тем і топ-N для врахування.

Якщо такі інструменти використовуватимуться в реальних проектах, потрібно автоматизовувати вибір оптимальних параметрів за замовчуванням. Утім вони можуть установлюватися менеджером у налаштуваннях так само, як періодичність оновлення моделі новими коментарями.

Якщо надати користувачам інструмент для перейменування ризиків з ключових слів, можна використовувати моделі для генерування більш природних назв.

Оскільки одна задача може містити декілька ризиків, то виокремити їх з розмови розробників — інше складне завдання, яке потребує дослідження.

ВИСНОВКИ

У ході дослідження вивчено джерела для визначення актуального напряму розвитку роботи та методів, що використовуються в даній сфері. На основі отриманої інформації створено огляд літератури, складено методологію роботи, розроблено інструменти для її використання і проведено експерименти для перевірки адекватності методу.

Для експерименту використовувалися коментарі відкритого jira-сховища Apache, потім токенізіровались, лематизувалися і для кожного з них обчислювалися характеристики VAD. Імовірність появи ризиків у задачі розраховується як середнє інтегроване зважене значення цих показників, де вагами є актуальність коментарю. У результаті коментарі з більш яскравим емоційним забарвленням дійсно мають великі показники, що може слугувати сигналом для менеджерів щодо завчасного реагування. Варто також проаналізувати сучасні комунікаційні компоненти (сленгові слова, смайли, скорочення, терміни) для отримання більш точних результатів. Новим виміром можна обрати показник сентиментальності коментарю та розглянути інші фреймворки визначення емоцій.

Запропоновано підхід для визначення назви потенційних ризиків задачі з її коментарів на основі побудови моделі LDA та використання отриманих коефіцієнтів для побудови хмари слів. Доцільно зробити акцент на автоматизацію обрання оптимальних параметрів (кількість тем та слів, що формують тему), які можуть змінюватися від проекту до проекту.

Виконано маркетинговий аналіз потенційного продукту, що може бути створений на основі запропонованої методології. З результатів опитування випливає, що майбутній додаток має містити базовий функціонал та реєстр ризиків, що може зробити його більш привабливим для можливих користувачів. Чим більше користувачів і даних, тим краще уявлення про ризики у розробленні програмних продуктів, що дає перспективи для швидшого виявлення ризиків та можливого усунення. Застосування аналізу часових рядів для обраних показників може допомогти передбачати проблеми за змінами у листуванні, а можливо і навіть з назви задачі.

ЛІТЕРАТУРА

1. *PMI. Success Rates Rise 2017 9th Global Project Management Survey. Pulse of the Profession,* 2017. Available: https://www.pmi.org//media/pmi/ documents/public /pdf/learning/thought-leadership/pulse/pulse-of-the-profession2017.pdf.
2. A. Murgia, P. Tourani, B. Adams, and M. Ortu, *Do Developers Feel Emotions? An Exploratory Analysis of Emotions in Software Artifacts*, 2014. Available: https://alessandromurgia.files.wordpress.com/2014/03/emotionanalysis.pdf.
3. *The Top 20 Most Popular Project Management Software*, 2018. Available: https://www.capterra.com/project-managementsoftware /#infographic.
4. *System Dashboard*. Available: https://issues.apache.org/jira/secure/ Dashboard.jspa.
5. *Teamwork. Right tools, right people, and right practices*. Available: https://www.atlassian.com/teamwork.
6. R.Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, and A. García-Crespo, *Using the Affect Grid to Measure Emotions in Software Requirements Engineering*, 2011. Available: http://www.jucs.org/jucs_17_9/using_the_affect_grid/jucs_17_09_ 1281_1298_colomo.pdf.
7. *L. Jun, Human factors in agile software development*, 2015. Available: https://arxiv.org/ftp/arxiv/papers/1502/1502.04170.pdf.
8. M. Ortu et al., *The JIRA Repository Dataset: Understanding Social Aspects of Software Development,* 2015. Available: http://mcis.polymtl.ca/publications/2015/ ortu_promise.pdf.
9. R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, *On negative results when using sentiment analysis tools for software engineering research*, 2017. Available: https://doi.org/10.1007/s10664-016-9493-x.
10. M. Mäntylä et al., *Mining Valence, Arousal, and Dominance – Possibilities for Detecting Burnout and Productivity?*, 2016. Available: https://arxiv.org/pdf/1603.04287.pdf.
11. M. Ortu et al., "Arsonists or Firefighters? Affectiveness in Agile Software Development", *Lecture Notes in Business Information Processing*, issue 251, 2016.
12. A.B. Warriner, V. Kuperman, and M. Brysbaert, *Norms of valence, arousal, and dominance for 13,915 English lemmas. Behavior Research Methods*, 2013. Available: https://doi.org/10.37.
13. T. Hofmann, *Probabilistic latent semantic indexing*, 1999. Available: https://www. researchgate.net/publication/2941307_Probabilistic_Latent_ Semantic_Indexing.
14. D. Blei, A. Ng, and M. Jordan, *Latent Dirichlet Allocation*, 2003. Available: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.
15. E. Guzman, *Visualizing emotions in software development projects*, 2013. Available: https://doi.org/10.1109/VISSOFT.2013.6650529.
16. D. Hoffman, M. Blei, and B. Francis, *Online Learning for Latent Dirichlet Allocation*, 2010. Available: https://www.di.ens.fr/~fbach/mdhnips 2010.pdf.
17. M. Röder, A. Both, and A. Hinneburg, *Exploring the Space of Topic Coherence Measures*, 2015. Available: https://svn.aksw.org/papers/2015/WSDM_ Topic_Evaluation/public.pdf.
18. D. Mimno et al., *Optimizing semantic coherence in topic models*, 2011. Available: http://dirichlet.net/pdf/mimno11 optimizing.pdf.
19. *jira-social-repository*. Available: https://github.com/marcoortu/jira-social-repository.

Надійшла 31.01.2020

INFORMATION ON THE ARTICLE

A.A. Liednikova, Educational and Scientific Complex "Institute for Applied System Analysis" of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: anna.liednikova@gmail.com.

D.V. Shypik, ORCID: 0000-0002-7667-4701, Educational and Scientific Complex "Institute for Applied System Analysis" of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: shypikd@gmail.com.

P.I. Bidyuk, ORCID: 0000-0002-7421-3565, Educational and Scientific Complex "Institute for Applied System Analysis" of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: pbidyuke_00@ukr.net.

PROJECT RISK ANALYSIS USING TEXT DATA MINING OF COMMENTS IN PROJECT MANAGEMENT SYSTEM JIRA / A.A. Liednikova, D.V. Shypik, P.I. Bidyuk

Abstract. During the study, a methodology was developed, and a software product was developed for project risk assessment based on developer communications, as well as the results of the program work on the data of the real project CASSANDRA of Apache Software Foundation. The methodology is implemented based on already well-known algorithms for determining the emotional components in the text of the VAD and matrix methods for project risk analysis using their developments that allow combining these different approaches. Obtaining the names of potential risks is performed using the model of constructing the LDA themes. The results allow us to determine the importance of the task by the communications and rank them in the middle of the project by the importance and need for additional attention that will allow project managers to understand and solve problems more quickly in the context of the product.

Keywords: project risk analysis, probabilistic latent semantic analysis, latent Dirichlet allocation model, natural language processing, sentiment analysis.

АНАЛИЗ РИСКОВ ПРОЕКТА С ПОМОЩЬЮ ТЕКСТОВОГО ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ КОММЕНТАРИЕВ В СИСТЕМЕ УПРАВЛЕНИЯ ПРОЕКТАМИ JIRA / А.А. Ледникова, Д.В. Шипик, П.И. Бидюк

Аннотация. В ходе исследования разработана методология и создан программный продукт для определения рисков проекта на базе коммуникации разработчиков, а также представлены результаты работы программы на данных реального проекта CASSANDRA компании Apache Software Foundation. Методология реализована на основе уже известных алгоритмов определения эмоциональных составляющих в тексте VAD и матричных методов анализа рисков проекта с использованием собственных разработок, позволять соединить эти разные подходы. Название потенциальных рисков определяется с помощью модели построения тем LDA. Полученные результаты позволяют определять важность задачи в соответствии с коммуникаций и ранжировать их в середине проекта по важности и необходимости дополнительного внимания, что в контексте продукта позволит менеджерам проекта более быстро понимать и решать проблемы.

Ключевые слова: анализ рисков проекта, вероятностный латентный семантический анализ, модель латентного распределения Дирихле, обработка природного языка, анализ тональности текста.

REFERENCES

1. *PMI. Success Rates Rise 2017 9th Global Project Management Survey. Pulse of the Profession,* 2017. Available: https://www.pmi.org//media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession2017.pdf.
2. A. Murgia, P. Tourani, B. Adams, and M. Ortu, *Do Developers Feel Emotions? An Exploratory Analysis of Emotions in Software Artifacts*, 2014. Available: https://alessandromurgia.files.wordpress.com/2014/03/emotionanalysis.pdf.

3. *The Top 20 Most Popular Project Management Software*, 2018. Available: https://www.capterra.com/project-managementsoftware /#infographic.
4. *System Dashboard*. Available: https://issues.apache.org/jira/secure/ Dashboard.jspa.
5. *Teamwork. Right tools, right people, and right practices*. Available: https://www.atlassian.com/teamwork.
6. R.Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, and A. García-Crespo, *Using the Affect Grid to Measure Emotions in Software Requirements Engineering*, 2011. Available: http://www.jucs.org/jucs_17_9/using_the_affect_grid/ jucs_17_09_ 1281_1298_colomo.pdf.
7. *L. Jun, Human factors in agile software development*, 2015. Available: https://arxiv.org/ftp/arxiv/papers/1502/1502.04170.pdf.
8. M. Ortu et al., *The JIRA Repository Dataset: Understanding Social Aspects of Software Development,* 2015. Available: http://mcis.polymtl.ca/publications/2015/ ortu_promise.pdf.
9. R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, *On negative results when using sentiment analysis tools for software engineering research*, 2017. Available: https://doi.org/10.1007/s10664-016-9493-x.
10. M. Mäntylä et al., *Mining Valence, Arousal, and Dominance – Possibilities for Detecting Burnout and Productivity?*, 2016. Available: https://arxiv.org/pdf/1603. 04287.pdf.
11. M. Ortu et al., "Arsonists or Firefighters? Affectiveness in Agile Software Development", *Lecture Notes in Business Information Processing*, issue 251, 2016.
12. A.B. Warriner, V. Kuperman, and M. Brysbaert, *Norms of valence, arousal, and dominance for 13,915 English lemmas. Behavior Research Methods*, 2013. Available: https://doi.org/10.37.
13. T. Hofmann, *Probabilistic latent semantic indexing*, 1999. Available: https://www. researchgate.net/publication/2941307_Probabilistic_Latent_Semantic_Indexing.
14. D. Blei, A. Ng, and M. Jordan, *Latent Dirichlet Allocation*, 2003. Available: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.
15. E. Guzman, *Visualizing emotions in software development projects*, 2013. Available: https://doi.org/10.1109/VISSOFT.2013.6650529.
16. D. Hoffman, M. Blei, and B. Francis, *Online Learning for Latent Dirichlet Allocation*, 2010. Available: https://www.di.ens.fr/~fbach/mdhnips 2010.pdf.
17. M. Röder, A. Both, and A. Hinneburg, *Exploring the Space of Topic Coherence Measures*, 2015. Available: https://svn.aksw.org/papers/2015/WSDM_Topic_ Evaluation/public.pdf.
18. D. Mimno et al., *Optimizing semantic coherence in topic models*, 2011. Available: http://dirichlet.net/pdf/mimno11 optimizing.pdf.
19. *jira-social-repository*. Available: https://github.com/marcoortu/jira-social-repository.