Carleton UNIVERSITY

Canada's Capital University

Cross-Platform Software Developer Expertise Learning

Norbert Eke, M.CS. Candidate

April 21st, 2020

Thesis Supervisor: Dr. Olga Baysal

Table of Contents

- Motivation
- Research Questions
- Related Work
- Methodology
- Results
- Implications
- Threats to Validity
- Future Work
- Contributions

Carleton UNIVERSITY
Canada's Capital University

Motivation

- **Recruiters have a hard time finding the right candidates**
  - Difficult to determine the actual expertise of developers from **resumes**

- **Analyzing collaborative platforms** (such as GitHub and Stack Overflow)
  - **User behaviour** is a rich source of data about the **software development process**
  - Excellent source of data for **identifying the right candidate** for a job
  - Developer interest and expertise can be inferred from data

- **Objectives:**
  - Investigate if **users maintain similar expertise profiles** across multiple collaborative platforms
  - **Develop data-driven techniques** that extract developer expertise from GitHub and Stack Overflow

Research Questions

RQ1: Expertise Extraction	How can we extract the major expertise areas of Stack Overflow and GitHub users? How do expertise trends compare on Stack Overflow and GitHub?
RQ2: Cross-platform Expertise	How similar are developer expertise profiles in two different collaborator platforms, Stack Overflow and GitHub?
RQ3: Transferable Knowledge	What knowledge is transferable from one platform to another?
RQ4: Expertise Evolution	How much does developer expertise evolve on Stack Overflow and GitHub?

Related Work Highlights

- **Vasilescu et al. (2013)**
  - One of the first researchers to explore the interaction between Stack Overflow and GitHub activities
- **Tian et al. (2013)**
  - Formulated the task of finding expert developers in open source software communities
- **Greene and Fischer (2016)**
  - Created a tool which extracts, explores and visualizes technical skills of GitHub users
- **Baltes and Diehl (2018)**
  - Created the first comprehensive theory of software development expertise
- **Treude and Wagner (2019)**
  - Studied the characteristics of GitHub and Stack Overflow text corpora

Methodology

- Data Acquisition
- Data Cleaning & Aggregation
- Expertise Study
- Research Roadmap
- Algorithm Design
- Data Analysis

Carleton University
Canada's Capital University

Data Acquisition

The GHTorent Dataset and Tool Suite

Georgios Gousios
Software Engineering Research Group
Delft University of Technology

ghtorrent
The GHTorrent project
The Netherlands http://ghtorrent.org

Repositories 6 | Packages | People | Projects

ghtorrent.org
The GHTorrent project website
website ghtorrent
Ruby 603 105 19 37 Updated 12 hours ago

ghtorrent-streaming
Streaming updates from the GHTorrent databases
ghtorrent
Ruby MIT 0 1 0 0 Updated on Dec 31, 2019

Top languages
Ruby TeX Puppet

People
This organization has no public members. You must be a member to see who's a part of this organization.

Report abuse

+

SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts

Sebastian Baltes
Lorik Dumani
research@sbaltes.com
dumani@uni-trier.de
University of Trier, Germany

Christoph Treude
christoph.treude@adelaide.edu.au
University of Adelaide, Australia

Stephan Diehl
diehl@uni-trier.de
University of Trier, Germany

Home
PUBLIC
Stack Overflow
Tags
Users
Jobs
TEAMS What's this?
Free 30 Day Trial

Array Only Prints Certain Elements and Not Others
Asked today Active today Viewed 16 times

0

I am currently in a Intro to Computer Language class and everyone few lessons I have to develop some minimal programs (we are given 5 different prompts and have to pick three of them to complete). Due to time, I moved on and completed a different program, but I still want to understand what is going wrong with this one. It is supposed to translate a given phrase into Pig Latin using for loops and different methods (as broken down in their template, which I cannot change, though I know there is a more efficient way). I can get the words in the phrases to translate, but when I print out the array (either by converting it to a string or running a for loop to print each element out separately) some of the elements only print the reference code. Could someone tell me what's going on? Below is the code and then a sample of a few print outs.

```
import java.util.Arrays;
import java.util.Scanner;

public class PigLatin {
  public static void main (String[] args){
    Scanner scnr = new Scanner(System.in);
    String userWord;
```

Linked/Joined

StackOverflow and GitHub: Associations Between Software Development and Crowdsourced Knowledge

Bogdan Vasilescu
Dept. of Math and Computer Science,
Eindhoven University of Technology,
The Netherlands
b.n.vasilescu@tue.nl

Vladimir Filkov
Computer Science Dept.,
UC Davis,
USA
filkov@cs.ucdavis.edu

Alexander Serebrenik
Dept. of Math and Computer Science,
Eindhoven University of Technology,
The Netherlands
a.serebrenik@tue.nl

unifiedId	githubEmail	SOUserIds
0	0.x29a.0@gmail.com	875020
1	000000mani@gmail.com	727906
2	001priyank@gmail.com	591785
3	00campbell@gmail.com	1256411
4	01.flakmonkey.10@gmail.com	657108
5	01.mandar@gmail.com	694891
6	0188801@gmail.com	1543768
7	01ttouch@gmail.com	943282
8	01walid@gmail.com	513327
9	05049pyj@gmail.com	247430

DATA that we have:

83,550 linked users, and all of their activity on Stack Overflow and GitHub

Data Cleaning & Aggregation

Two stage data cleaning process:

1. User level text pre-processing:

- **Removal of html links, symbols**
- **Removal of stop-words**, **tags**
- **Tokenization**, then **remove numbers**, but not words that contain numbers

2. Corpus level text pre-processing:

- **Detect frequent phrases**
- **Strip punctuation** and **symbols**
- **Remove rare** and **very common** tokens

Building SO User Profiles	Building GH User Profiles
• **Badge** names • Profile page's **about me** • **Questions** • **Answers** • **Post Titles, Tags** • **Comments**	• **Project Name,** Description, Labels, Languages Used • **Commit Comments** • **Code review** (Pull Request) Comments

Carleton UNIVERSITY
Canada's Capital University

Expertise Study

- Overall goal: **obtain expertise ground truth**
- **Sampled 100 random users active** on both Stack Overflow and GitHub
- **Created 10 different Google Forms**, each containing **10** Stack Overflow and GitHub **user profile links**
- **Evaluated our model outputs against human annotations** using **cosine similarity** scores between the two bag-of-words

Section 1 of 2

GH user's expertise - Bin 1

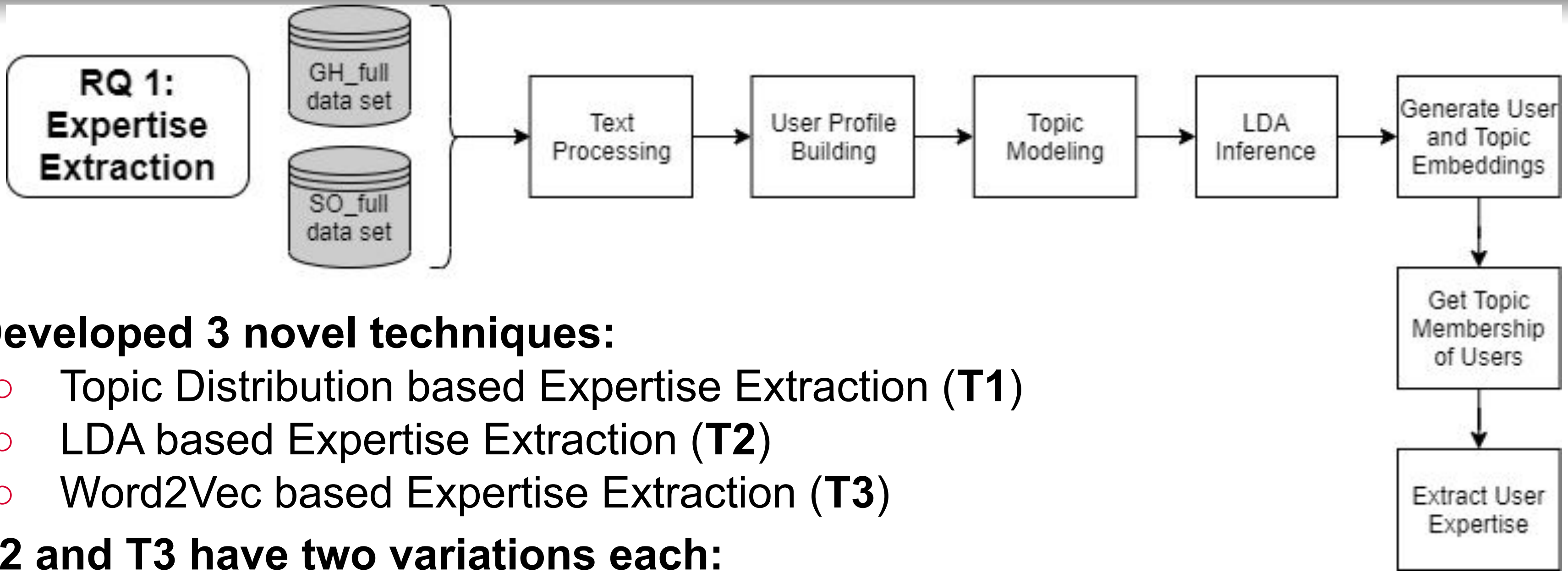
Enter 20 comma-separated words for describing each user's expertise. Your answers needs to come from evaluating a user's full activity (i.e. every publicly available data that you can see and click-through) on Github.

Sample answer for a fictional user:
"pytorch, CNN, RNN, auto-encoders, Keras, git, tensorflow, python, java, C#, web_dev, machine_learning, random_forest, SVM, nlp, Java_streams, distributed_computing, parallel_computing, R, statistics, visualization"

Resulting Data

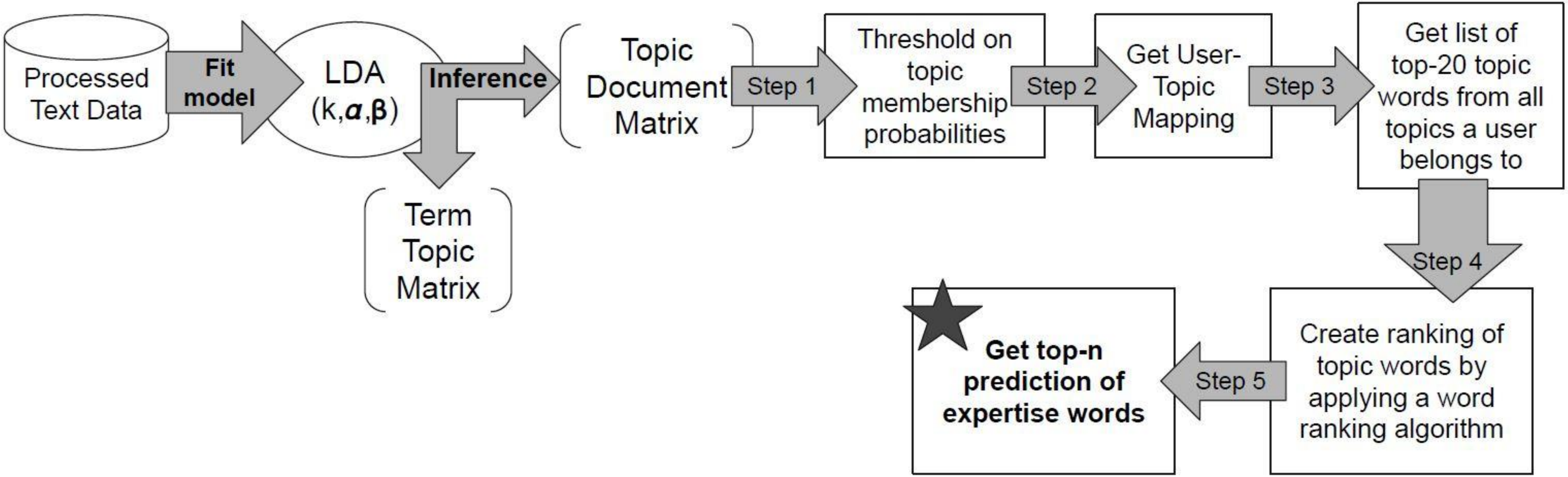
	A	B	G	H
1	sample_ID	profile_url	Processed_Annotator_1	Processed_Annotator_2
2	0	https://stackoverflow.com/users/1081	java;angularjs;jpa;android;localization;ic	java;javascript;object;oriented;sql;jquery;asp
3	1	https://stackoverflow.com/users/1278	unix;linux;tex;latex;server;windows;ubu	git;rxjs;cryptography;mercurial;open;source;
4	2	https://stackoverflow.com/users/1026	haskell;monads;functional_programmin	haskell;functional_programming;monad;ison
5	3	https://stackoverflow.com/users/2097	c#;dot_net;entity_framework;database	c#;dot_net;mvc;aspdot_net;entity;framewo
6	4	https://stackoverflow.com/users/5628	reactjs;enzyme;asynchronous;python;ja	react;javascript;jest;python;multiparadigm;u
7	5	https://stackoverflow.com/users/3698	ssl;python;openssl;nodejs;pips;webserve	ssl;cryptographic;python;hashing;openssl;too
8	6	https://stackoverflow.com/users/9455	azure;python;javascript;dropbox;c#;win	microsoft;azure;python;javascript;dropbox;c
9	7	https://stackoverflow.com/users/1024	python;javascript;jquery;html;css;numpy	numpy;python;javascript;html;css;jquery;php
10	8	https://stackoverflow.com/users/1270	r;dataframe;reshape;aggregate;string;lis	r;data;frame;aggregate;datatables;reshape;t
11	9	https://stackoverflow.com/users/4568	nodejs;docker;javascript;jquery;ajax;htn	java;javascript;object;oriented;sql;jquery;ang
12	10	https://stackoverflow.com/users/2981	python;django;programming;jquery;aspc	fullstack;web;web_design;web_developmen
13	11	https://stackoverflow.com/users/8041	clojure;iphone;ios;oauth;lisp;multithrea	clojure;jvm;java;iphone;lisp;mutithreading;io
14	12	https://stackoverflow.com/users/2052	java;c;bash;c++;shell;linux;memory;poin	debugging;testing;verification;validation;soft
15	13	https://stackoverflow.com/users/5411	computer;science;software_engineering	python;haskell;object;oriented;programming
16	14	https://stackoverflow.com/users/1495	computer;science;software_engineering	java;sql;mongodb;javascript;jvm;database;sc

Algorithm Design

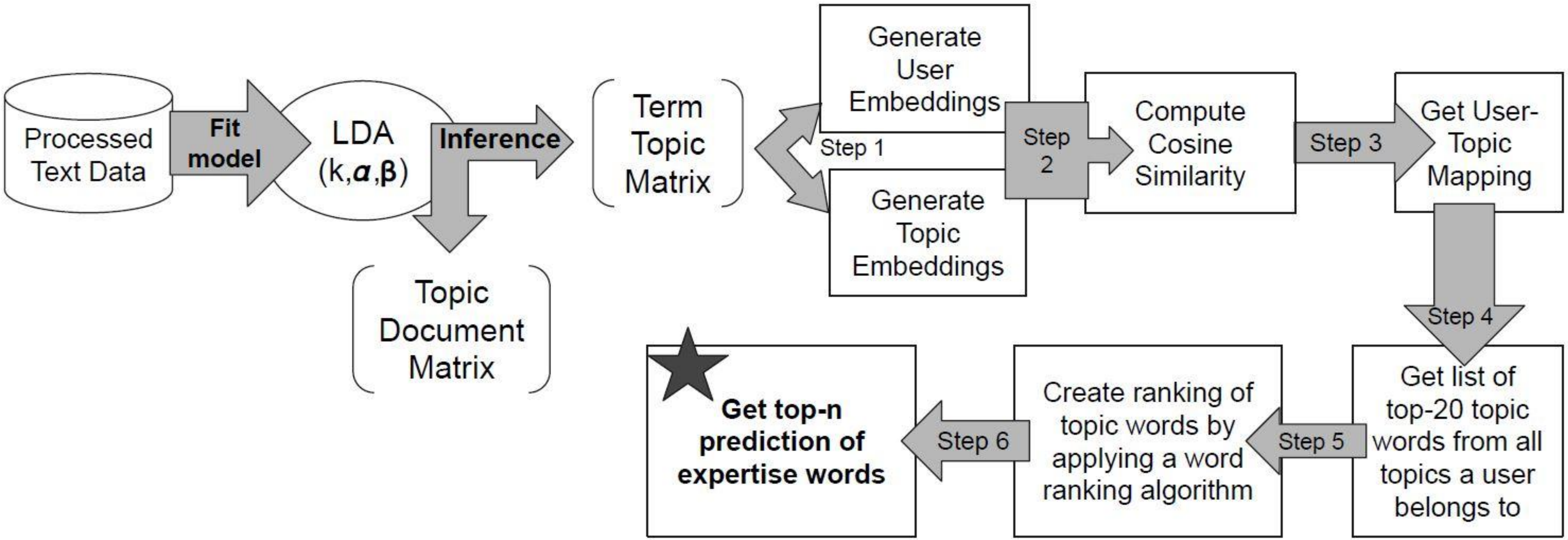


- **Developed 3 novel techniques:**
  - Topic Distribution based Expertise Extraction (**T1**)
  - LDA based Expertise Extraction (**T2**)
  - Word2Vec based Expertise Extraction (**T3**)
- **T2 and T3 have two variations each:**
  - LDA_AVG, LDA_MAX, and W2V_AVG, W2V_MAX
- **Performed 2 experiments* (1 & 2) on 2 different data sets (A & B)**
  - Experiment 1A, 2A on GitHub & Experiment 1B, 2B on Stack Overflow

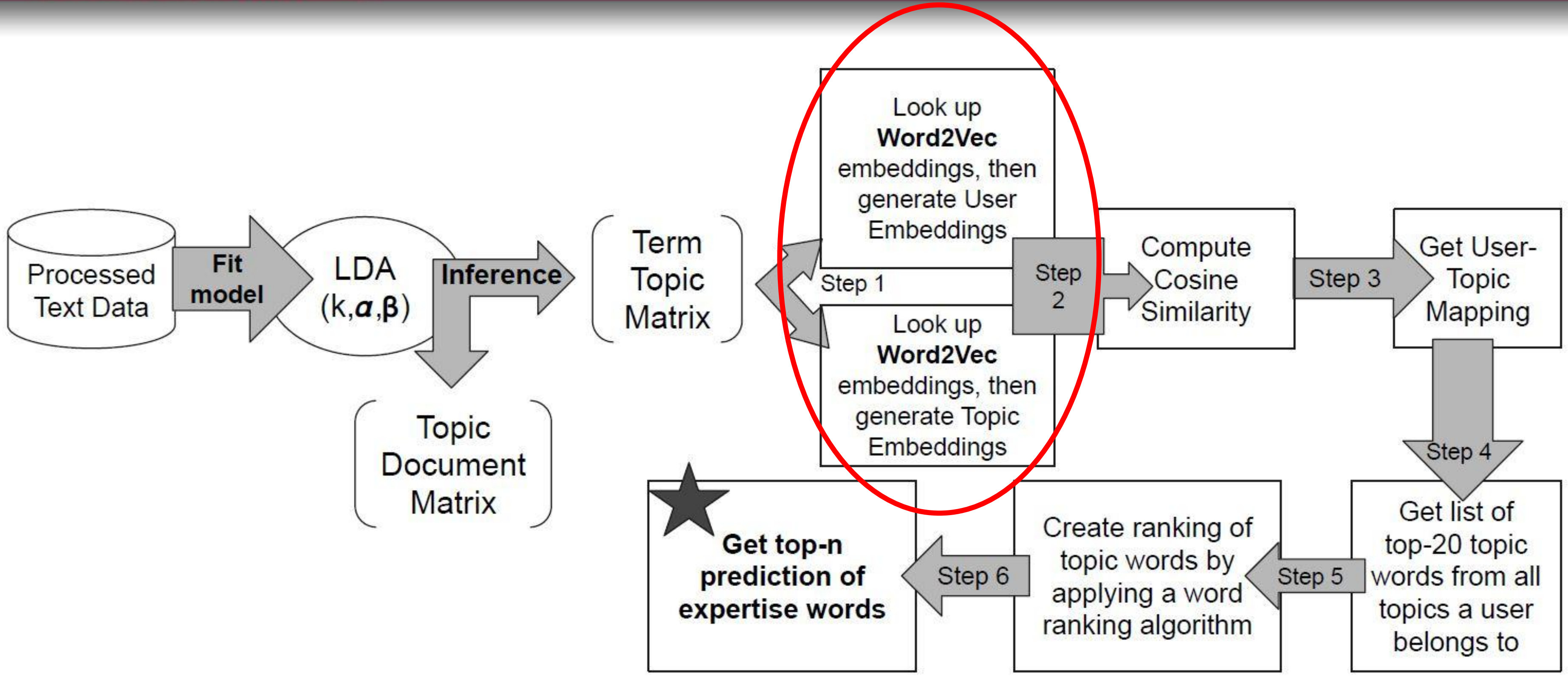
Topic Distribution Based Expertise Extraction



LDA Based Expertise Extraction



Word2Vec Based Expertise Extraction



Results

Carleton UNIVERSITY

Canada's Capital University

RQ 1 Results

Table 4: Results of Experiment 1A - Expertise Extraction from GitHub Data.

Top 3 Results	Metrics \ Model	TDistr	LDA_AVG	LDA_MAX	W2V_AVG	W2V_MAX	Baseline
1	Cosine Sim.	0.6690	0.7187	0.7357	$\boxed{0.7998}$	0.7317	0.5962
	Jaccard Sim.	0.0658	0.0751	0.1040	0.0765	0.1049	0.0286
	BLEU Score	0.1197	0.1340	0.1767	0.1368	0.1782	0.0540
2	Cosine Sim.	0.6689	0.7183	0.7351	0.7959	0.7316	0.5962
	Jaccard Sim.	0.0658	0.0750	0.1037	0.0818	0.1049	0.0286
	BLEU Score	0.1197	0.1338	0.1762	0.1452	0.1782	0.0540
3	Cosine Sim.	0.6683	0.7183	0.7351	0.7959	0.7316	0.5962
	Jaccard Sim.	0.0652	0.0750	0.1037	0.0818	0.1049	0.0286
	BLEU Score	0.1186	0.1338	0.1762	0.1452	0.1782	0.0540

Table 5: Example of Cosine Similarity Scores between Term-Pairs.

Term 1	Term 2	Cosine Similarity	Term 1	Term 2	Cosine Similarity
php	python	0.2529	html	javascript	0.6024
java	python	0.3929	ajax	jquery	0.6315
analysis	visualization	0.4524	sklearn	tensorflow	0.6858
nodejs	reactjs	0.4909	bagging	random-forest	0.7251
java	jdk	0.5517	mysql	postgresql	0.7997
xml	json	0.5866	keras	tensorflow	0.8391

Answer to RQ 1

RQ1: Expertise Extraction

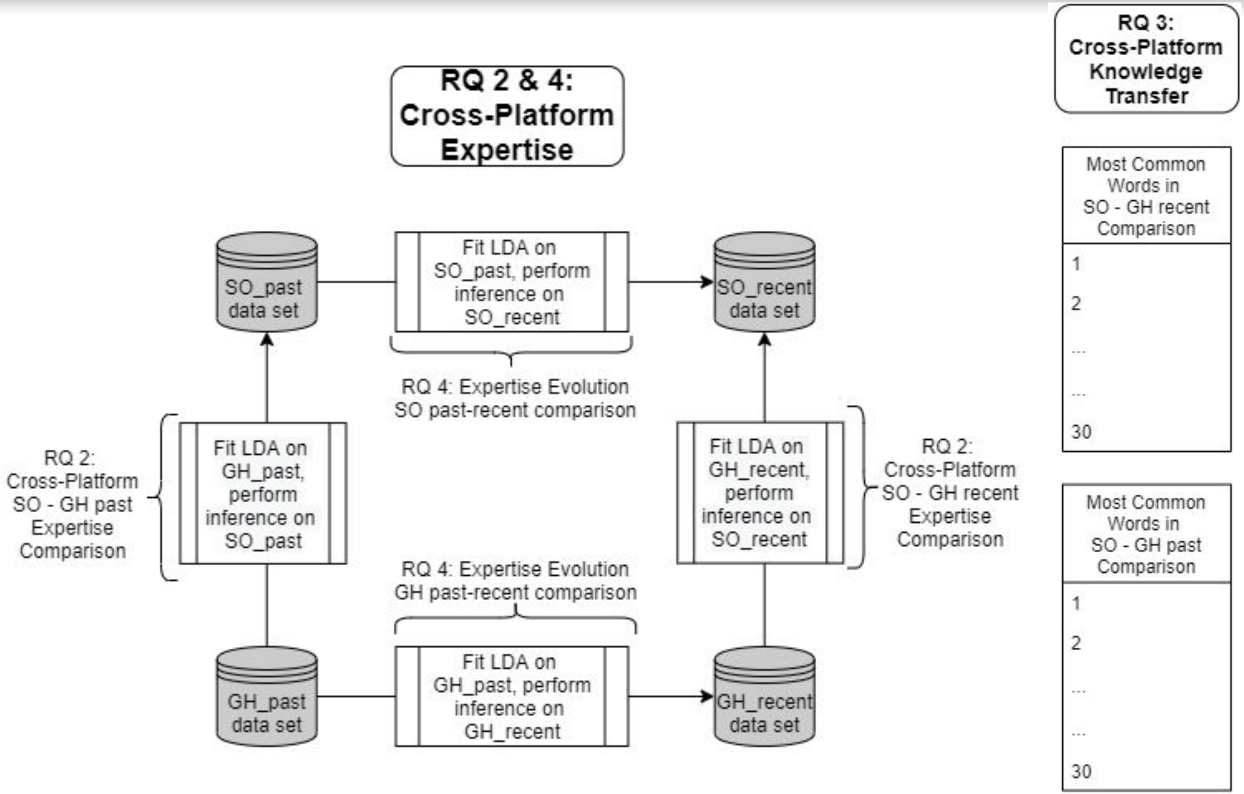
How can we extract the major expertise areas of Stack Overflow and GitHub users?

How do expertise trends compare on Stack Overflow and GitHub?

- **W2V_AVG** model performs best for 3 out of 4 experiments
- Expertise trend similarities:
  - Both include a **few popular programming** language related topics, and both are dominated by **web development** related skills
- Expertise trend differences:
  - GitHub expertise areas are **few, and more general**
  - Stack Overflow expertise areas are **more specific and numerous**

RQ 2-4 Research Roadmap

- Fitted LDA models on **4 text corpora**
- **Evaluation metric used:** Topic Coherence
- Performed **hyper-parameter optimization**
- **When comparing two text corpora**, we fitted LDA on larger corpus, **performed inference** on the other corpus



Answer to RQ 2

RQ2:
Cross-platform Expertise

How similar are developer expertise profiles in two different collaborator platforms, Stack Overflow and GitHub?

- **64% of the population has no overlap** in the GH-recent & SO-recent text corpora comparison
- **67% of the population has no overlap** in the GH-past & SO-past text corpora comparison
- These results suggest that **developers build different expertise profiles** on GitHub and Stack Overflow.

Answer to RQ 3

RQ3:
Transferable
Knowledge

What knowledge is transferable from one platform to another?

Common expertise terms suggest that **source code, version control and web development** related skills are most transferable knowledge.

Answer to RQ 4

RQ4:
Expertise
Evolution

How much does developer expertise evolve on Stack Overflow and GitHub?

- For the comparison of **GH past-recent** text corpora most of the analyzed **GitHub** population **has largely changed their expertise** over time.
- For the comparison of **SO past-recent** text corpora most of the analyzed **Stack Overflow** population **did not or only slightly changed their expertise** over time.

Implications: Our Recommendations

- **Recruiters**
  - For hiring use **expertise profiles** obtained via **data-driven** approaches
- **Project Managers**
  - Consider integrating an **expertise based task assignment system**
- **Stack Overflow and GitHub Users**
  - Consider using **multiple collaborative platforms** to **gain more knowledge** and **become an expert**
- **Researchers**
  - Consider **combining state-of-the-art algorithms** from multiple areas of computer science/statistics in their research work

Threats to Validity

There are several threats, but I will highlight the key ones:

- Data pre-processing
  - The **blend of natural text and source code** in Stack Overflow posts caused some challenges to the text pre-processing routine
  - **Not all code elements** are cleaned up and filtered out properly.

- Data quality
  - The SO-recent data set **lacks active users**
  - Lack of user activity data could lead to **misleading topic trends** in LDA
  - This is the **nature of the data set**, thus we could not mitigate this issue

Future Work

- **Separation of natural text from source code elements**
- Alternatives for **better user and topic vector representation:**
  - Mixing Dirichlet Topic Models and Word Embeddings (LDA2Vec)
  - Topic Modeling in Embedding Spaces (ETM)
- Use of **author-topic models** to model user activities
- Try to **predict, summarize or classify** a user's expertise area

Carleton UNIVERSITY
Canada's Capital University

Contributions

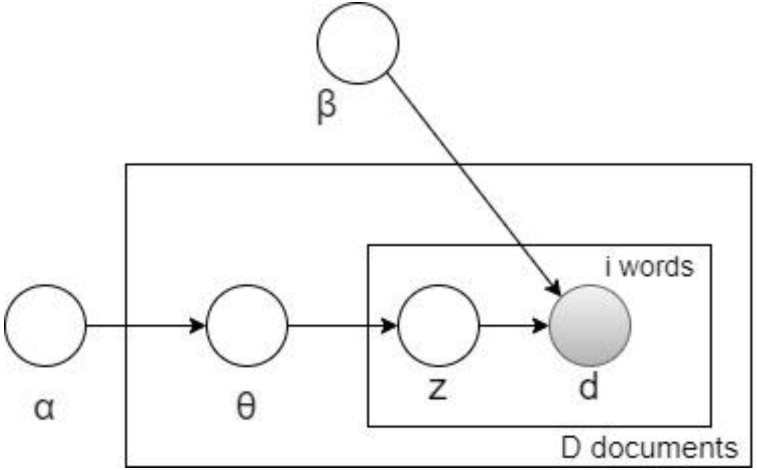
1. **Development of three novel techniques** to extract developer expertise topics from Stack Overflow and GitHub
2. **Analysis of developer expertise trends** on Stack Overflow and GitHub
3. **Comparison of developer expertise** across two collaborative platforms
4. **Empirical evidence about knowledge transfer** between two collaborative platforms
5. Analysis of **developer expertise evolution trends** from two collaborative platforms
6. Collection of developer expertise **ground truth data set**
7. **Development of four new data sets** by aggregating Stack Overflow and GitHub data

Appendix

Topic Modeling - LDA

LDA assumes that it receives a collection of D documents, each of length L_i as input. Being a generative probabilistic model, LDA has a generative process, which can be described in the following steps:

1. For each topic k in $\{1, ..., K\}$ draw a word distribution $\phi_k \sim Dir(\beta)$
2. For each document d in $\{1, ..., D\}$ draw a topic distribution $\phi_d \sim Dir(\alpha)$
3. For each word i of document d draw a topic distribution $z_{d,i} \sim Multinomial(\phi_d)$ and a word distribution $w_{d,i} \sim Multinomial(\phi_{z_{d,i}})$



LDA - Gibbs Sampling

Algorithm 1 LDA Training Process using Gibbs Sampling.

Require: D documents, K number topics, $IterNum$ - Maximum number of Gibbs Sampling iterations

```
for each document d do
  for each word i in document d do
    Randomly assign word i to one of the K topics.
  end for
end for

# Perform IterNum iterations of Gibbs sampling
for index in 1, ..., IterNum do
  for each document d do
    for each word i in document d do
      for each topic t in 1, ..., K topics do
        Compute full conditional probability P from Equation 1
        Reassign word i to topic t with probability P
        # In the LDA model P is the probability that topic t generated word i
      end for
    end for
  end for
end for
```

Stack Overflow User Profiles

Carleton University
Canada's Capital University

Table 2: Stack Overflow user profile extraction.

Attribute Name	Attribute(s)	Description
Badges	Badges.Name	Concatenation of list of badges obtained by the user
About Me	Users.AboutMe	Stack Overflow user profile's about me description
Post Answer	Posts.Body, AcceptedAnswerId	The user's each individual answer concatenated with the question it is related to
Post Question	Posts.Body	The user's each individual question concatenated with the accepted answer it is related to
Title and Tags for Questions	Posts.Tags, Posts.Title	Concatenation of post tags and title for each question that the user asked
Title and Tags for Answers	Posts.Title, Posts.Tags	Concatenation of post tags and title for each answer that the user provided
Comments	Comments.Text, Posts.Body, Posts.Title	Concatenation of the user's each individual comment and the post (question or answer) it is related to

GitHub User Profiles

Table 3: GitHub user profile extraction.

Attribute Name	Attribute(s)	Description
Project Name, Description and Metadata	Projects.[name, description, language], Repo-Labels.name	Description of each user's project together with the repository's name, description, languages used, repository labels it contains.
Commit-Comments	Commit-Comments.body	List of user's commit comments.
Code-review Comments	Pull-Request-Comments.body	List of user's code review (pull request) comments.

Hyper-parameter Optimization

- For **α** hyper-parameter we learnt an asymmetric prior from the data for both.

- For **β** hyper-parameter we defined a parameter search space of [0.001, 1], then performed a hyper-parameter optimization against this search space

- For **k**, number of topics, we defined a parameter search space of [3, 100], then performed a hyper-parameter optimization against this search space, with the evaluation metric selected (or task-based evaluation)

Topic Coherence Measures

Roder et al. proposed a coherence framework that consists of four steps:

1. segmentation of word-pairs
2. estimation of word probabilities computation of confirmation measures
3. which test how strong is the coherence between any two word pairs
4. Aggregating "confirmation measures" to form a single coherence score.

- Four promising topic coherence metrics emerge as the metrics that are most correlated with human judgements and interpretability

- UCI is a metric based on Point-wise Mutual Information (PMI), and it estimates word probabilities based on word co-occurrences. UCI is also known for having the largest correlation with human annotations.
- $UMASS$ is an asymmetric "confirmation" metric between top word pairs, and it takes into consideration the ordering of topic words in a topic.
- $NPMI$ is a metric based on normalized Point-wise Mutual Information (PMI), and it works by generating context vectors for each topic word in a topic.
- C_V is the best performing coherence metric, being a hybrid metric between indirect cosine similarity measures, the $NPMI$ metric and a boolean sliding

Topic Distr. Based Expertise Extraction

Algorithm 2 Topic Distribution Based Expertise Extraction.

```
Require: LDA: Topic Model, threshold: Probability Threshold
Get Topic-Document Matrix M from fitted LDA topic model
User-Topic Mapping = { }
Expertise-Predictions = { }

// Create User-Topic Mapping of each user
for all users user_ID in the data set do
    Get user profile document d of user user_ID
    Get topic distribution TD of document d from matrix M
    listOfTopics = [ ]
    for all topics t with non-zero probability in TD do
        if probability of topic t ≥ threshold then
            Add topic t to listOfTopics
        end if
    end for
    User-Topic Mapping[ d_i ] = listOfTopics

    // Extract expertise for each user
    listOfWords = [ ]
    for all topic t in User-Topic Mapping[ user_ID ] do
        TW = Get top-20 topic words that describe topic t
        Add topic words TW to listOfWords
    end for
    Apply a word ranking algorithm to listOfWords
    Expertise-Predictions[ user_ID ] = sorted listOfWords
end for
return Expertise-Predictions
```

LDA Based Expertise Extraction

Algorithm 5 Expertise Extraction using LDA Based User and Topic Embeddings.

```
Require: D: User Profile Documents, LDA: Topic Model, threshold: Cosine Similarity Threshold
Get Term-Topic Matrix M learned during inference of fitted LDA topic model
User-Topic Mapping = { }
Expertise-Predictions = { }
Run LDA based User Embedding Generation to get UserEmb matrix
Run LDA based Topic Embedding Generation to get TopicEmb matrix

for each user embedding U in UserEmb do
    listOfTopics = [ ]
    user_to_topic_sim = [ ]
    for each topic embedding T in TopicEmb do
        Computer cosine similarity SIM between user embedding U and topic embedding T
        Add SIM to user_to_topic_sim
    end for

    Get list of topics LT associated with non-zero cosine similarities in user_to_topic_sim
    for topic t in LT do
        if user_to_topic_sim[ t ] ≥ threshold then
            Add topic t to listOfTopics
        end if
    end for
    Get User_ID associated with user embedding U
    User-Topic Mapping[ User_ID ] = listOfTopics

    // Extract expertise for each user
    listOfWords = [ ]
    for all topic t in User-Topic Mapping[ user_ID ] do
        TW = Get top-20 topic words that describe topic t
        Add topic words TW to listOfWords
    end for
    Apply a word ranking algorithm to listOfWords
    Expertise-Predictions[ user_ID ] = sorted listOfWords
end for
return Expertise-Predictions
```

Carleton UNIVERSITY

Canada's Capital University

LDA Based User Embeddings

Algorithm 3 LDA Based User Embedding Generation.

Require: M: Term-Topic Matrix, D: User Profile Documents

```
listOfEmbeddings = [ ]
for all users user_ID in the data set do
    Get user profile document d for user user_ID from D
    Get unique set words WordSet in document d
    listOfVectors = [ ]

    for each word w in WordSet do
        Get 1 × k vector v of latent scores for word w from matrix M
        Add vector v to listOfVectors
    end for
    Convert listOfVectors to matrix LV with dimensions n × k; n = length(WordSet)
    Perform column-wise max-pooling or average-pooling to reduce matrix LV to a user embedding vector emb, dimension 1 × k
    Add user embedding vector emb to listOfEmbeddings
end for
Convert listOfEmbeddings to matrix E with dimensions u × k; where u = number of users in the data set
return Matrix E
```

LDA Based Topic Embeddings

Algorithm 4 LDA Based Topic Embedding Generation.

Require: LDA: Topic Model, M: Term-Topic Matrix

listOfEmbeddings = [ ]
for each topic T_i in LDA fitted model **do**
    Get top-20 topic words TW describing topic T_i
    listOfVectors = [ ]
    for each word w in TW **do**
        Get $1 \times k$ vector v of latent scores for word w from matrix M
        Add vector v to listOfVectors
    end for
    Convert listOfVectors to matrix LV with dimensions $n \times k$; where $n = 20$, $k =$ number of topics in LDA model
    Perform column-wise max-pooling or average-pooling to reduce matrix LV to a topic embedding vector emb, dimension $1 \times k$
    Add topic embedding vector emb to listOfEmbeddings
end for
Convert listOfEmbeddings to matrix E with dimensions $k \times k$
return Matrix E

Word2Vec Based Expertise Extraction

Algorithm 8 Expertise Extraction Using Pre-trained Word2Vec Based User and Topic Embeddings.

Require: D: User Profile Documents, LDA: Topic Model, *threshold*: Cosine Similarity Threshold, $Word2Vec$: Pre-trained Model

```
Get Term-Topic Matrix M learned during inference of fitted LDA topic model
User-Topic Mapping = { }
Expertise-Predictions = { }
Run Word2Vec based User Embedding Generation to get UserEmb matrix
Run Word2Vec based Topic Embedding Generation to get TopicEmb matrix

for each user embedding U in UserEmb do
  listOfTopics = [ ]
  user_to_topic_sim = [ ]
  for each topic embedding T in TopicEmb do
    Computer cosine similarity SIM between user embedding U and topic embedding T
    Add SIM to user_to_topic_sim
  end for

  Get list of topics LT associated with non-zero cosine similarities in user_to_topic_sim
  for topic t in LT do
    if user_to_topic_sim[ t ] ≥ threshold then
      Add topic t to listOfTopics
    end if
  end for
  Get User_ID associated with user embedding U
  User-Topic Mapping[ User_ID ] = listOfTopics

  // Extract expertise for each user
  listOfWords = [ ]
  for all topic t in User-Topic Mapping[ user_ID ] do
    TW = Get top-20 topic words that describe topic t
    Add topic words TW to listOfWords
  end for
  Apply a word ranking algorithm to listOfWords
  Expertise-Predictions[ user_ID ] = sorted listOfWords
end for
return Expertise-Predictions
```

Word2Vec Based User Embeddings

Algorithm 6 Pre-trained Word2Vec Based User Embedding Generation.

Require: D: User Profile Documents,*word2vec*: Pre-trained Model

listOfEmbeddings = [ ]
for all users $user\_ID$ in the data set **do**
  Get user profile document d for user $user\_ID$ from D
  Get unique set words $WordSet$ in document d
  listOfVectors = [ ]

  for each word w in $WordSet$ **do**
    Look up vector representation v of word w in pre-trained $Word2Vec$ model
    Add vector v to listOfVectors
  end for
  Convert listOfVectors to matrix LV with dimensions $n \times d$; $n = length(WordSet)$,d = dimensionality of $Word2Vec$ model
  Perform column-wise max-pooling or average-pooling to reduce matrix LV to a user embedding vector emb, dimension $1 \times d$
  Add user embedding vector emb to listOfEmbeddings
end for
Convert listOfEmbeddings to matrix E with dimensions $u \times d$; where u = number of users in the data set
return Matrix E

Word2Vec Based Topic Embeddings

Algorithm 7 Pre-trained Word2Vec Based Topic Embedding Generation.

Require: LDA: Topic Model, $Word2Vec$: Pre-trained Model

listOfEmbeddings = [ ]
for each topic T_i in LDA fitted model **do**
  Get top-20 topic words TW describing topic T_i
  listOfVectors = [ ]
  for each word w in TW **do**
    Look up vector representation v of word w in pre-trained $Word2Vec$ model
    Add vector v to listOfVectors
  end for
  Convert listOfVectors to matrix LV with dimensions $n \times d$; where $n = 20$, d = dimensionality of $Word2Vec$ model
  Perform column-wise max-pooling or average-pooling to reduce matrix LV to a topic embedding vector emb, dimension $1 \times d$
  Add topic embedding vector emb to listOfEmbeddings
end for
Convert listOfEmbeddings to matrix E with dimensions $k \times d$, k = number of topics in LDA model
return Matrix E

Carleton UNIVERSITY
Canada's Capital University

RQ 1 - Experiment 1B Results

Table 7: Results of Experiment 1B - Expertise Extraction from Stack Overflow Data.

Top 3 Results	Metrics \ Model	TDistr	LDA_AVG	LDA_MAX	W2V_AVG	W2V_MAX	Baseline
1	Cosine Sim.	0.5044	0.5582	**0.5837**	0.5607	0.5820	0.3721
	Jaccard Sim.	0.0160	0.0406	0.0435	0.0320	0.0556	0.0104
	BLEU Score	0.0313	0.0770	0.0823	0.0612	0.1041	0.0199
2	Cosine Sim.	0.4997	0.5560	0.5717	0.5574	0.5746	0.3721
	Jaccard Sim.	0.0295	0.0747	0.0814	0.0404	0.0784	0.0104
	BLEU Score	0.0560	0.1366	0.1471	0.0768	0.1424	0.0199
3	Cosine Sim.	0.4755	0.5409	0.5676	0.5537	0.5736	0.3721
	Jaccard Sim.	0.0127	0.0754	0.0821	0.0718	0.0305	0.0104
	BLEU Score	0.0247	0.1366	0.1478	0.1312	0.0584	0.0199

Table 5: Example of Cosine Similarity Scores between Term-Pairs.

Term 1	Term 2	Cosine Similarity	Term 1	Term 2	Cosine Similarity
php	python	0.2529	html	javascript	0.6024
java	python	0.3929	ajax	jquery	0.6315
analysis	visualization	0.4524	sklearn	tensorflow	0.6858
nodejs	reactjs	0.4909	bagging	random-forest	0.7251
java	jdk	0.5517	mysql	postgresql	0.7997
xml	json	0.5866	keras	tensorflow	0.8391

RQ 1 - Experiment 2A Results

Carleton University

Canada's Capital University

Table 8: Results of Experiment 2A - Expertise Extraction from GitHub Data.

Top 3 Results	Metrics \ Model	TDistr	LDA_AVG	LDA_MAX	W2V_AVG	W2V_MAX	Baseline
1	Cosine Sim.	0.6828	0.7377	0.7602	**0.7761**	0.7680	0.5962
	Jaccard Sim.	0.0246	0.0218	0.0251	0.0209	0.0144	0.0286
	BLEU Score	0.0277	0.0245	0.0279	0.0212	0.0127	0.0540
2	Cosine Sim.	0.6827	0.7364	0.7598	0.7750	0.7672	0.5962
	Jaccard Sim.	0.0249	0.0256	0.0254	0.0204	0.0138	0.0286
	BLEU Score	0.0281	0.0292	0.0288	0.0208	0.0122	0.0540
3	Cosine Sim.	0.6821	0.7362	0.7595	0.7748	0.7671	0.5962
	Jaccard Sim.	0.0241	0.0244	0.0259	0.0218	0.0131	0.0286
	BLEU Score	0.0271	0.0278	0.0295	0.0222	0.0116	0.0540

Table 5: Example of Cosine Similarity Scores between Term-Pairs.

Term 1	Term 2	Cosine Similarity	Term 1	Term 2	Cosine Similarity
php	python	0.2529	html	javascript	0.6024
java	python	0.3929	ajax	jquery	0.6315
analysis	visualization	0.4524	sklearn	tensorflow	0.6858
nodejs	reactjs	0.4909	bagging	random-forest	0.7251
java	jdk	0.5517	mysql	postgresql	0.7997
xml	json	0.5866	keras	tensorflow	0.8391

RQ 1 - Experiment 2B Results

Table 9: Results of Experiment 2B - Expertise Extraction from Stack Overflow Data.

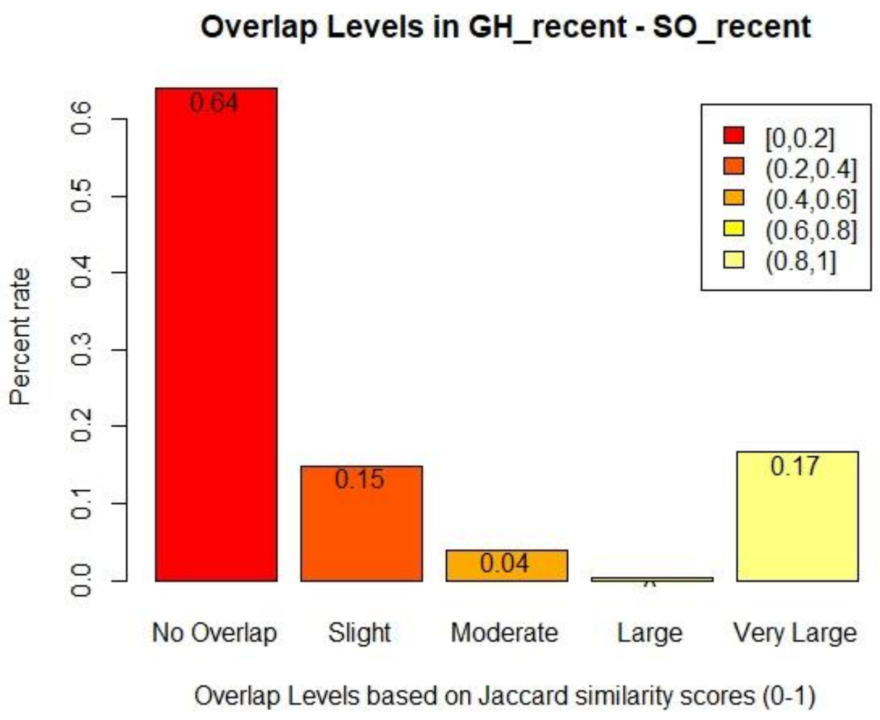
Top 3 Results	Metrics \ Model	TDistr	LDA_AVG	LDA_MAX	W2V_AVG	W2V_MAX	Baseline
1	Cosine Sim.	0.5263	0.5361	0.5837	**0.5902**	0.5117	0.3721
	Jaccard Sim.	0.0099	0.0230	0.0435	0.0295	0.0102	0.0104
	BLEU Score	0.0117	0.0231	0.0822	0.0363	0.0087	0.0199
2	Cosine Sim.	0.5080	0.5304	0.5717	0.5821	0.5087	0.3721
	Jaccard Sim.	0.0223	0.0175	0.0816	0.0196	0.0120	0.0104
	BLEU Score	0.0267	0.0196	0.1471	0.0222	0.0101	0.0199
3	Cosine Sim.	0.5036	0.5266	0.5697	0.5698	0.5086	0.3721
	Jaccard Sim.	0.0111	0.0149	0.0303	0.0180	0.0106	0.0104
	BLEU Score	0.0136	0.0169	0.0579	0.0180	0.0090	0.0199

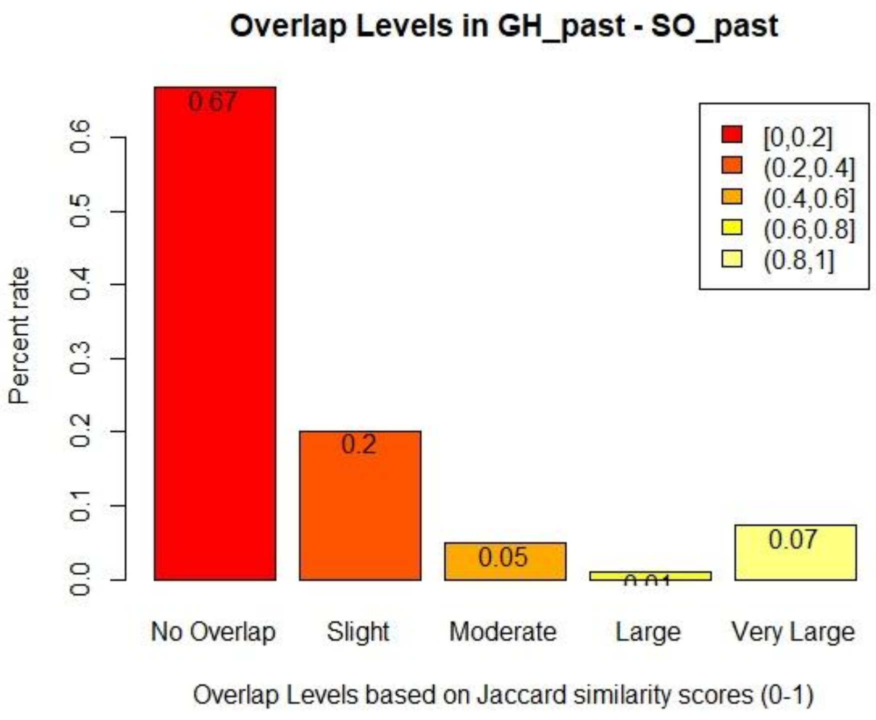
Table 5: Example of Cosine Similarity Scores between Term-Pairs.

Term 1	Term 2	Cosine Similarity	Term 1	Term 2	Cosine Similarity
php	python	0.2529	html	javascript	0.6024
java	python	0.3929	ajax	jquery	0.6315
analysis	visualization	0.4524	sklearn	tensorflow	0.6858
nodejs	reactjs	0.4909	bagging	random-forest	0.7251
java	jdk	0.5517	mysql	postgresql	0.7997
xml	json	0.5866	keras	tensorflow	0.8391

Carleton UNIVERSITY
Canada's Capital University

RQ 2 - Results





RQ 3 - Results

Carleton University
Canada's Capital University

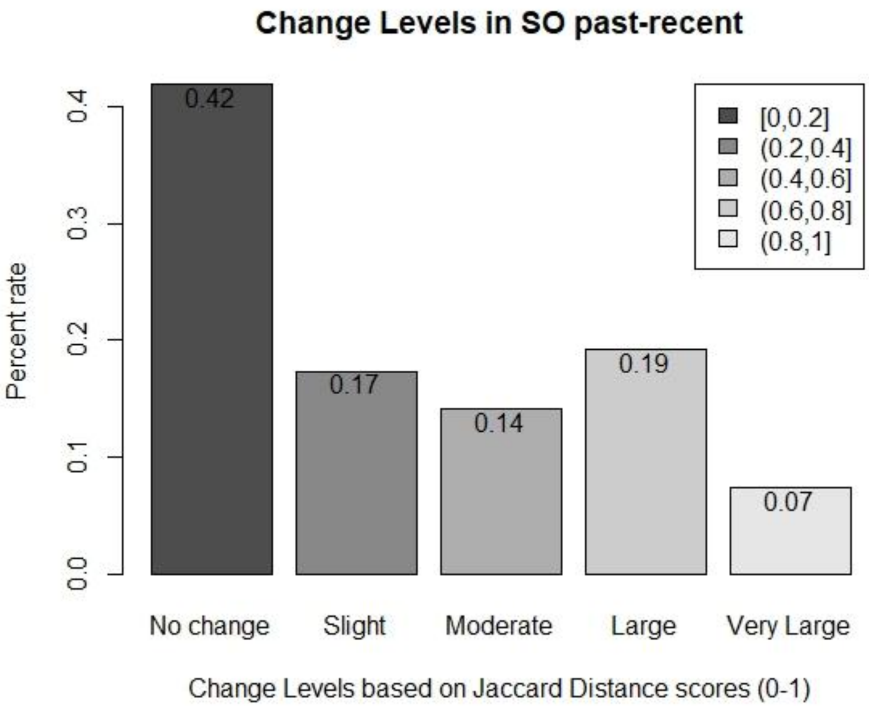
Table 12: Most common words in GH-past and SO-past text corpora.

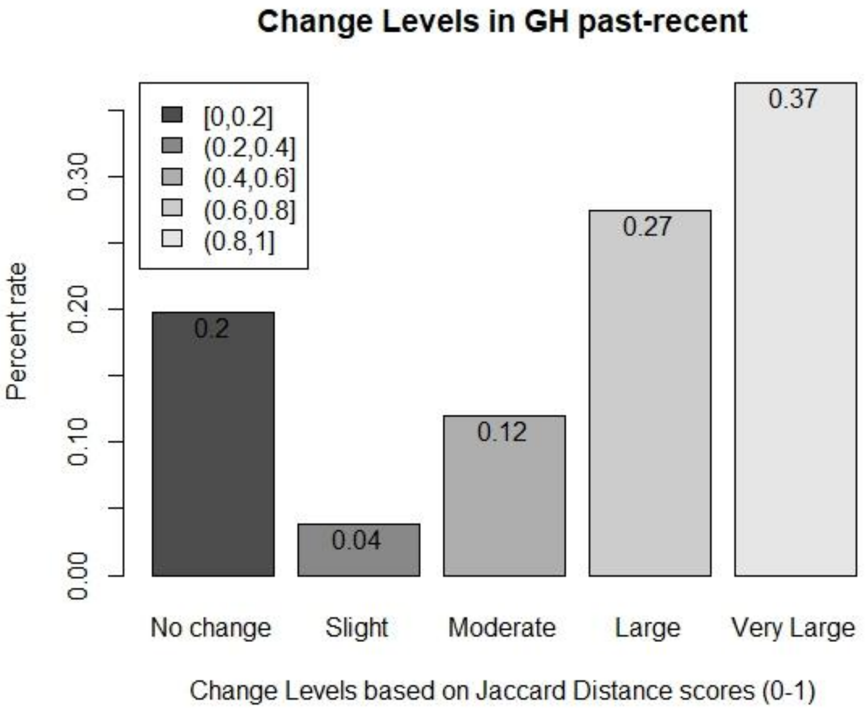
Ranking	Keyword	Frequency	Ranking	Keyword	Frequency
1	library	43,123	16	base	16,798
2	code	37,621	17	implementation	16,670
3	simple	32,762	18	client	15,833
4	type	30,948	19	test	15,723
5	javascript	30,044	20	http	15,674
6	project	26,255	21	page	15,423
7	web	25,083	22	game	13,890
8	tool	24,967	23	website	13,255
9	https	24,738	24	package	12,982
10	file	24,737	25	repository	11,690
11	html	22,333	26	add	11,421
12	github	21,317	27	method	11,421
13	script	20,318	28	line	11,252
14	source	20,266	29	api	11,144
15	language	19,855	30	datum	10,980

Table 13: Most common words in GH-recent and SO-recent text corpora.

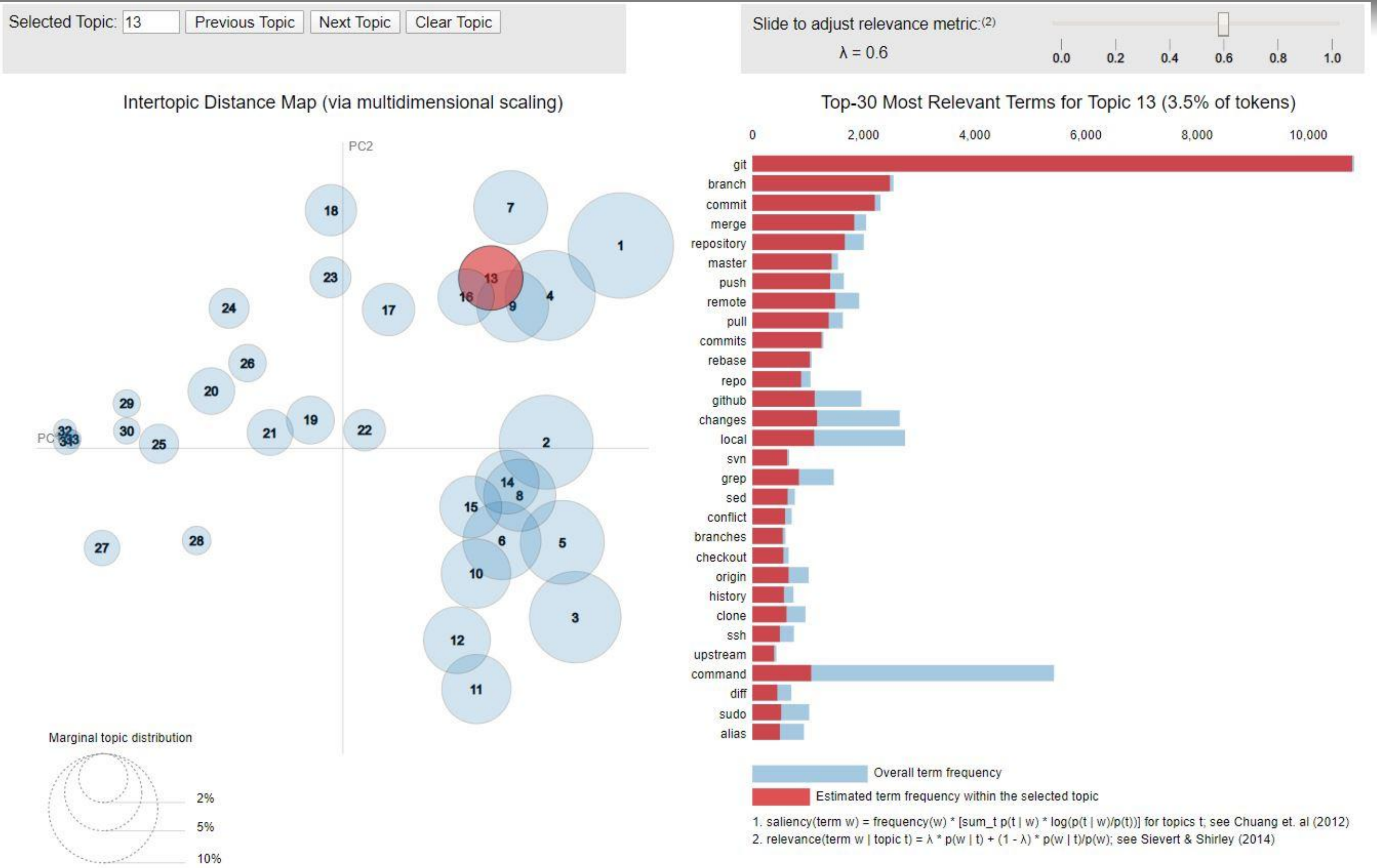
Ranking	Keyword	Frequency	Ranking	Keyword	Frequency
1	test	56,302	16	heroku	15,764
2	simple	51,226	17	buildpack	15,764
3	library	44,781	18	cli	15,764
4	app	43,750	19	rb	15,764
5	api	35,881	20	activerecord	15,764
6	base	26,075	21	rspec	15,764
7	client	25,381	22	active	15,764
8	code	22,052	23	github	15,297
9	file	21,538	24	web	12,890
10	application	19,620	25	add	12,849
11	https	16,764	26	change	12,849
12	ruby	15,764	27	remove	12,849
13	rail	15,764	28	check	12,849
14	ember	15,764	29	make	11,231
15	gem	15,764	30	comment	11,231

RQ 4 - Results


Change Levels in SO past-recent
Percent rate
0.0
0.1
0.2
0.3
0.4
0.42
0.17
0.14
0.19
0.07
[0,0.2]
(0.2,0.4]
(0.4,0.6]
(0.6,0.8]
(0.8,1]
No change
Slight
Moderate
Large
Very Large
Change Levels based on Jaccard Distance scores (0-1)

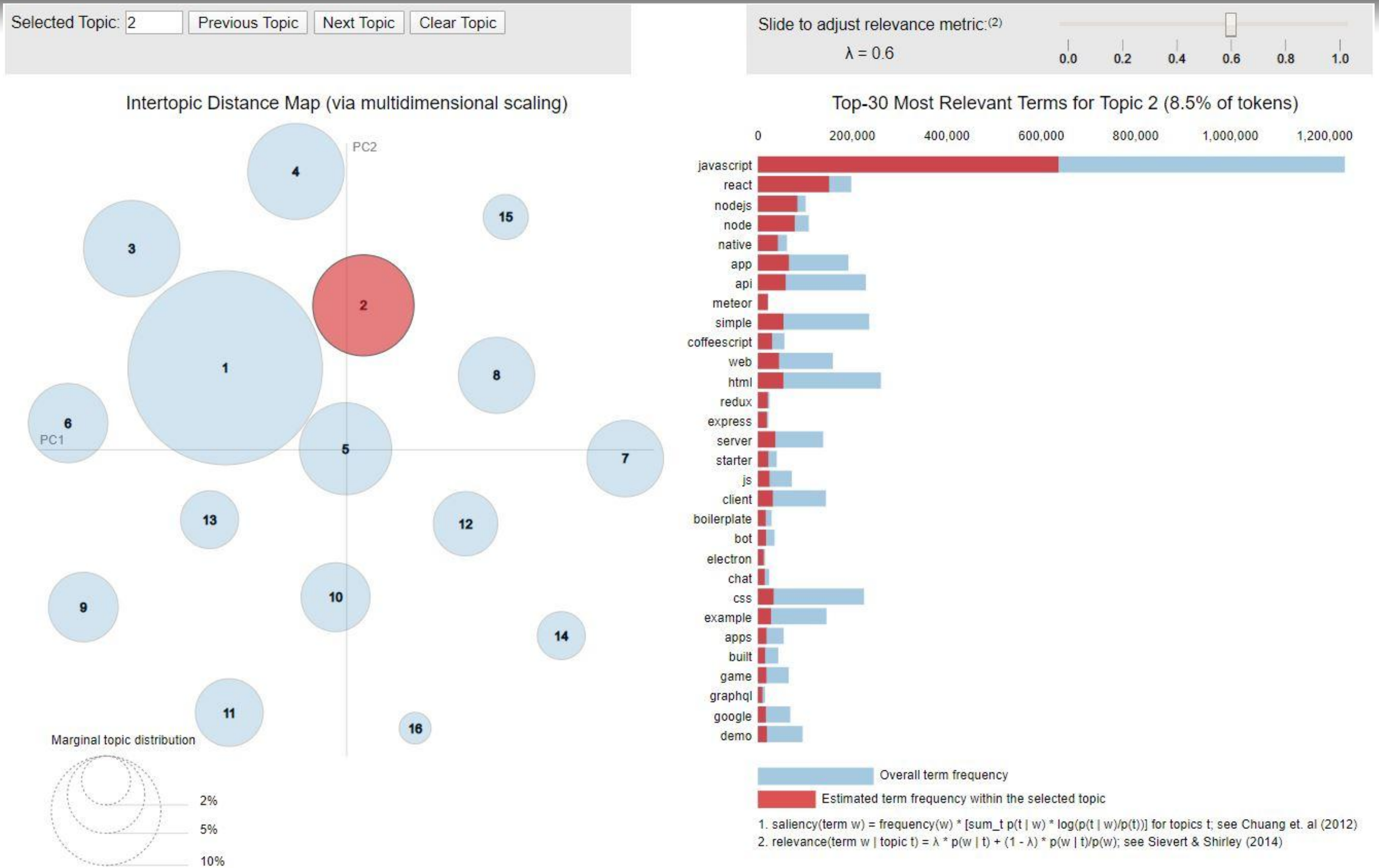

Change Levels in GH past-recent
Percent rate
0.00
0.10
0.20
0.30
0.2
0.04
0.12
0.27
0.37
[0,0.2]
(0.2,0.4]
(0.4,0.6]
(0.6,0.8]
(0.8,1]
No change
Slight
Moderate
Large
Very Large
Change Levels based on Jaccard Distance scores (0-1)

Stack Overflow Topic Modeling Visualization



GitHub Topic Modeling Visualization

Carleton University
Canada's Capital University



Expertise Study Google Forms

Section 1 of 2

GH user's expertise - Bin 1

Enter 20 comma-separated words for describing each user's expertise. Your answers needs to come from evaluating a user's full activity (i.e. every publicly available data that you can see and click-through) on Github.

Sample answer for a fictional user:
"pytorch, CNN, RNN, auto-encoders, Keras, git, tensorflow, python, java, C#, web_dev, machine_learning, random_forest, SVM, nlp, Java_streams, distributed_computing, parallel_computing, R, statistics, visualization"

What is your name ? *

Short answer text

GH - user 1 - https://github.com/brunnurs/ *

Long answer text

GH - user 2 - https://github.com/whiteinge/ *

Long answer text

GH - user 3 - https://github.com/Gabriel439/ *

Long answer text

GH - user 4 - https://github.com/davideicardi/ *

Section 2 of 2

SO user's expertise - Bin 2

Enter exactly 20 comma-separated words for describing each user's expertise. Your answers needs to come from evaluating a user's full activity (i.e. every publicly available data that you can see and click-through) on Stack Overflow.

Sample answer for a fictional user:
"pytorch, CNN, RNN, auto-encoders, Keras, git, tensorflow, python, java, C#, web_dev, machine_learning, random_forest, SVM, nlp, Java_streams, distributed_computing, parallel_computing, R, statistics, visualization"

SO - user 1 - https://stackoverflow.com/users/298171/fish2000/ *

Long answer text

SO - user 2 - https://stackoverflow.com/users/80410/Mark Probst/ *

Long answer text

SO - user 3 - https://stackoverflow.com/users/20520/Diomidis Spinellis/ *

Long answer text

SO - user 4 - https://stackoverflow.com/users/541136/Aaron Hall/ *

Long answer text

Carleton UNIVERSITY
Canada's Capital University

Model Parameters - Experiment 1

Table 18: Parameters of Models from Experiment 1A - Expertise Extraction from GitHub Data

Top 3 Results	Model / Parameters	TDistr	LDA_AVG	LDA_MAX	W2V_AVG	W2V_MAX
1	Topic Number	11	10	11	**16**	11
	Beta Value	0.01	0.005	0.05	**0.05**	0.05
2	Topic Number	11	10	11	16	11
	Beta Value	0.005	0.001	0.001	0.001	0.001
3	Topic Number	11	10	11	16	11
	Beta Value	0.05	0.01	0.005	0.005	0.005

Table 19: Parameters of Models from Experiment 1B - Expertise Extraction from Stack Overflow Data

Top 3 Results	Model / Parameter	TDistr	LDA_AVG	LDA_MAX	W2V_AVG	W2V_MAX
1	Topic Number	40	33	**33**	31	48
	Beta Value	0.01	1	**1**	0.1	1
2	Topic Number	29	27	27	48	27
	Beta Value	0.1	0.5	0.5	1	1
3	Topic Number	45	16	27	14	31
	Beta Value	0.1	0.5	1	1	1

Carleton UNIVERSITY
Canada's Capital University

Model Parameters - Experiment 2

Table 20: Parameters of Models from Experiment 2A - Expertise Extraction from GitHub Data

Top 3 Results	Parameters \ Model	TDistr	LDA_AVG	LDA_MAX	W2V_AVG	W2V_MAX
1	Topic Number	11	21	16	16	16
	Beta Value	0.005	0.05	0.05	0.01	0.05
2	Topic Number	11	11	10	16	16
	Beta Value	0.01	0.01	0.001	0.005	0.01
3	Topic Number	11	11	10	16	16
	Beta Value	0.001	0.001	0.005	0.05	0.001

Table 21: Parameters of Models from Experiment 2B - Expertise Extraction from Stack Overflow Data

Top 3 Results	Parameter \ Model	TDistr	LDA_AVG	LDA_MAX	W2V_AVG	W2V_MAX
1	Topic Number	40	13	33	14	30
	Beta Value	0.01	0.1	1	1.0	0.001
2	Topic Number	13	36	27	32	30
	Beta Value	0.1	1.0	0.5	0.1	0.1
3	Topic Number	45	27	36	16	30
	Beta Value	0.1	0.01	1	0.5	0.005