Journal of Information Processing Vol.0 1–4 (??? 1992)

[DOI: 10.2197/ipsjjip.0.1]

Technical Note

Magnet or Sticky? A Stack Overflow Tag-by-Tag Typology

Kotori Hieda[1,a)] Yu Mingzhe[1,b)] Yasutaka Kamei[2,c)]

Received: xx xx, xxxx, Accepted: xx xx, xxxx

Abstract: Stack Overflow (SO) is one of the most popular question and answer sites for software developers. SO stores posts assigned with tags that correspond to the keywords of each question. If a developer asks question related to Python and inputs "Python" tag on the post, the developers interested in Python can participate in the post easily. Since 2008, SO has become one of the most trusted online communities. In this study, we explore developers' interest by analyzing how they use tags. We classify tags into four: (1) attractive, (2) stagnant, (3) fluctuating, and (4) terminal based on magnet values and sticky values. We analyze data of table "Posts" of approximately 42 million posts in SO and table "Users" of approximately 9 million rows of user information. Results reveal that some historical events in IT are retrieved, which include launching of new tools and terminating of services with the transition of magnet value and sticky value.

Keywords: magnet, sticky, tag, user migration, OSS census

1. Introduction

The Pew Research Center (PRC) [1] is the U.S. fact finder that provides information on social problems and demographic trends that shape the United States and the world. *magnet states* are defined as states where a high percentage of thepopulation migrated from the outside, whereas sticky states have a high proportion of those living in the same state since birth. Nevada is a *magnet states* because 86% of the population migrated from other states. It is possible to find the movement of American citizens by studying this demographic trend.

For software developers, understanding other developers interests are important as the popurality of developers have advantages. Many developers like to work with convenient and easy-to-use tools. To develop a project efficiently, developers must focus on long-term-projects. In this study, we focus on new and existing topics of Stack Overflow (SO). Inspired by previous studies [2], we apply Magnet and Sticky metrics to the topics collected in SO. The magnet metric is the number of new developers attracted to a topic and Sticky metric is the number of existing developers who stay with the topic. We examined the values of tags "magnets" and tags "sticky" by classifying them to the tags *programming language*, *framework*, and *environment*. We also compared the news and history of software companies and web services. If changes in their characteristics are discovered, we examine factors responsible for the changes based on their magnet values and sticky values.

[1] åååååå¢ ååå
Graduate School of Information Science and Electrical Engineering, Kyushu University
[2] ååå ååå¢
Faculty of Information Science and Electrical Engineering, Kyushu University
[a)] hieda@f.ait.kyushu-u.ac.jp
[b)] yumingzhe@f.ait.kyushu-u.ac.jp
[c)] kamei@ait.kyushu-u.ac.jp

We address the following two research questions:

(RQ1) What are typical values of magnet and sticky in Stack Overflow?

In many cases, the sticky value is higher than the magnet value. In addition, the magnet value rate decreases more than that of the sticky.

(RQ2) How do magnet and sticky values change over time?

We can identify obsolete tags. When the tags move quadrant, we find that something happens.

2. Definition of Magnet and Sticky

This section describes how we measure the appeal and adhesion of users on different topics. Following the Pew Research Center (PRC) definition, we use the Magnet and Sticky metrics to illustrate the migratory trends of the U.S. citizens. The PRC defines magnet states as states where a large proportion of adults are from other states. From the magnet metric, the proportion of adults residing in the magnet states were not born in the state. PRC defines the sticky state as the state where a large proportion of adults born there continue to live in the state. Thus, the sticky metric for the state is the proportion of adult residents born in the state. These definitions are good for a population study where a single adult can only occupy one state at a time. However, the definition is inapplicable to the topics discussed by the SO users as users can ask or answer questions on several topics at the same time. Therefore, we expand new definitions for SO topics 1

Magnet and Sticky in SO

SO contents are questions and answers to the questions and comments [3] called Posts in the SOTorrent database. Each question has one or more tags that separate the question into different topics. Posts on a question have their creator (for the question content, one is the questioner, and for the answer, one is the respondent), a participant of the topics, and the question. We also

	User	2017	2018	Magnet 2017	Magnet 2018	Sticky 2018
Topic1	A	●	●	4/4	2/3	1/4
	B	●	●			
	C	●	●			
	D	●				
	E					
	G					
Topic2	B	●	●	2/4	3/3	1/2
	C	●	●			
	E		●			
	F		●			
	G					
Topic3	A	●	●	2/4	2/3	2/2
	B	●	●			
	C		●			
	E		●			
	F		●			

Fig. 1 -Example of Magnet and Sticky values definition

define the asking or answering questions activity on some topic as a discussion topic. For example, a classical question in SO has three tags like Java, Apache, and Linux asked by a user A and answered by a user B and C. Thus, C are participants of Java, Apache, and Linux-topics.

Magnet. Magnet topics attract a large proportion of new users; thus, we calculate the magnetism of a topic as the proportion of users who ask or answer question during the time period under new registered users in a specific year.

Sticky. In sticky topics, a large proportion of the users keep participating in the discussion in the time period. Thus, we calculate the stickiness of a topic as the proportion of the users who discuss the topic in the time period.

Example (Calculating magnet and sticky values). To calculate magnet and sticky values of topics that belong to a major category, we use a total of six questions (a, b, c, d, e, f) and seven users (A, B, C, D, E, F, G); the Last Activity Date of question a, b, c was in 2017, and question d, e, f in 2018. The registration date of user A, B, C, D was in 2017, and the registration date of user E, F, G was in 2018 [2].

To calculate the magnet metric, we observe four new users who registered their accounts in 2017 (A, B, C, and D), and all of them discuss topic 1, whereas two of them (B, C) participate in the discussion topic 2 and 3. In this case, the Magnet value of topic 1 in 2017 is 4/4, topic 2 is 2/4, and topic 3 is 2/4.

To calculate the sticky metric on topic 1, three users participated in the discussion in 2017 (A, B, and C). Only one of them participated in the discussion in 2018 (A). Hence, the sticky value of project 1 is 1/4. In topic 2, two users participated in the discussion in 2017 (B and C); however, only one of them participated in the discussion in 2018 (B). Though new users E, F, G, participated in the discussion in 2018, we still calculate the value of sticky as 1/2. For the same reason, the sticky of topic 3 is 2/2 in 2018.

Example (Merging similar subjects into one topic). We merge subjects (i.e., tags) that belong to analogous subjects into one topic. For example, we consider that different version numbers

Question	Last Activity Date	Tag1	Tag2	Topic1	Topic2
a	2017	1.0		1	
b	2017	1.1		1	
c	2017	1.0	3.1	2	3
d	2018	2.1		2	
e	2018	3.0		3	
f	2018	1.2		1	

Fig. 2 -Example of the merge of tags belonging to analogous subjects

(e.g., tag “Python-2.7” and tag “Python-3.6”) are one of the common examples of analogous tags.

We also need to merge derivatives of the same technology on different platforms, merge derivatives of special tools in a certain tool family, or a combination of technology with a common used library, etc. For example, the tag “reactjs,” “react-router,” “reactjs-flux,” “create-react-app” should be merged into one topic “react.” We can get this information from the “Related Tag” column of the “Tag Info” of SO.”

Figure 2 shows that question a has tag 1.0, question b has tag 1.1 and question f has tag 1.2. According to our merge rule, they all belong to topic 1. The question c has tag 2.0 and tag 3.1, showing that it belongs to topic 2 and 3. Therefore, question d belongs to topic 2, and question e belongs to topic 3.

3. Dataset

We analyze the SO dataset (SOTorrent) provided by Sebastian Baltes et al. [4]. SOTorrent is an open dataset based on the official SO data dump. “SOTorrent provides access to the version history of SO content at the level of whole posts and individual text or code blocks.”

The dataset consists of 20 different tables stored in data on official SO data dump and data extracted from the original official SO data dump.

However, we only analyze the data from table *Posts* of approximately 42 million posts from SO and table *Users* of approximately 9 million rows of user information from July 2008 to September 2018. We focus on users, tags, and time of questions. Moreover, we consider users who ask or answer questions in SO. Those who comment or like/dislike questions or answers are excluded from the statistics.

In this paper, we consider a user to be one who asks or answers questions in Stack Overflow. Those who comment or like/unlike on questions or answers are not counted in the statistics.

4. Study Results

In this section, we provide answers to these questions:

4.1 (RQ1) What are the typical values of magnet and sticky in Stack Overflow?

Approach. We calculated magnet and sticky values as defined in Section2. We plot the magnet value on the vertical axis and the sticky value on the horizontal axis. We classify the plotted points into four quadrants.

Attractive: Tags with a high magnet and sticky value. By under-

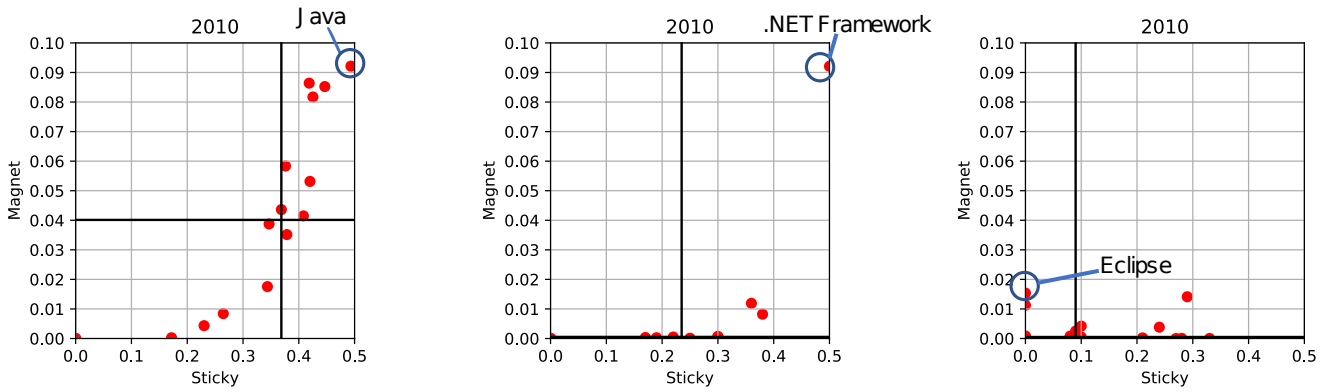

Fig. 3 -Distribution of Magnet and Sticky values in Programing Language, Framework and Environment

Table 1 Quadrant Transition of Framework 2010 - 2018

Programing Language	2010	2011	2012	2013	2014	2015	2016	2017	2018
assembly	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal
Bash	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal
C	Fluctuating	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal
C#	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
C++	Attractive	Attractive	Attractive	Attractive	Attractive	Stagnant	Stagnant	Stagnant	Stagnant
CSS	Terminal	Stagnant	Stagnant	Terminal	Terminal	Stagnant	Stagnant	Terminal	Terminal
Go	*	*	Terminal	Terminal	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating
HTML	Attractive	Attractive	Attractive	Attractive	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant
Java	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
JavaScript	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
PHP	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
Python	Attractive	Fluctuating	Fluctuating	Attractive	Attractive	Fluctuating	Fluctuating	Attractive	Attractive
Ruby	Terminal	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Terminal	Terminal	Terminal
SQL	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant
Swift	*	*	*	*	Terminal	Fluctuating	Fluctuating	Fluctuating	Fluctuating
TypeScript	*	*	*	Terminal	Terminal	Terminal	Fluctuating	Fluctuating	Fluctuating

standing the tags, we can discover the interests of developers.

Fluctuating: Tags with a higher magnet and lower sticky value. This tag attracts people, but it is short-term. Excellent developers will not continue to be disinterested.

Stagnant: Tags with a lower magnet and higher sticky value. These tags are difficult to attract new users, but it can maintain existing users.

Terminal: Tags with the lower magnet and lower sticky value. This tag can neither attract new users' developers nor keep them interested.

The median of the magnet and sticky values for each year is used for the quadrant threshold because the median is unaffected by outliers. From the sticky value definition in Section2, the sticky value depends on the number of tag users in a year and the following year. To answer RQ1, we got 9 years worth of sticky values based on the information on the number of tag users from 2009 to 2018. The sticky value must depend on the number of new tag users, but, if the number of new tag users in a year is too low, the sticky value will be too small. To remove noise, we fix thresholds for each topic When the magnet and sticky value is less than the threshold value, it is set to 0.

We did not analyze all the tags at once, but divided them into three categories for analysis. The selected categories and their contents are:

- programming languages (assembly, Bash, C, C, C ++, CSS, Go, HTML, Java, JavaScript, PHP, Python, Ruby, SQL, Swift, TypeScript)
- frameworks (.NET Framework, Angular, Cordova, Django, Hadoop, Node.js, React, Spark, Spring, TensorFlow, Torch, Xamarin)
- environment (Android Studio, Atom Editor, Eclipse, Emacs, IntelliJ, IPython, Jupyter, NetBeans, Notepad++, PhpStorm, PyCharm, RStudio, RubyMine, Sublime Text, TextMate, Vim, Visual Studio, Visual Studio Code, Xcode)

We chose these tags based on SO's survey of over 100,000 developers in 2018 [*1]. We focused on tags used by more than 5% of developers who answered the questionnaire.

Results: Figure 3 shows a quadrant plot of the magnet and sticky values of the 2010 framework, programming language, and environmental tag[*2], revealing that the magnet value is lower than the sticky. value. The results are similar to the findings of the PRC. For example, the U.S. citizens are likely to live in the same house than to change houses; it is easier for developers to continue developing the same content.

Summary. Tags with high magnet value are easy to use even

[*1] https://insights.stackoverflow.com/survey/2018

[*2] We choose the year 2010 because it is the first year for which yearly data of sticky value can be obtained.

for beginners. A tag familiar to Java has a high magnet value and sticky value, and it is more attractive.

4.2 (RQ2) How do magnet and sticky values change over time?

Approach: From 2010 to 2018, we calculated the probability of the tags moves quadrants from one year to the following year. For example, there were six Attractive tags in 2010. Of the six Attractive tags in 2010, five that were Attractive the following year. Therefore, the transition probability from Attractive to Attractive for 2010 - 2011 is 5/6 or 83.3%.

Quantitative results: Table **??** shows the proportion of the vertical axis is in the quadrant for 20102018, the horizontal axis is the year old. From the table, the ratio of tags is the highest for those that do not move the quadrants from the previous year to the following year in any programming language, framework, and environment. Since the tags have hardly changed from any quadrant to 2, once the tags have become popular to a certain extent, the users of the tags have not significantly reduced. This shows that once tags have become less popular, it may continue to be unpopular.

Manual analysis: Table 1 shows the transition of each tag quadrant in the framework, revealing how the tags move in the quadrant. Xamarin is an interesting example. Xamarin is an API for Android and iOS developed with C#. Thus, it is difficult to develop an application without having knowledge of both applications. Developing Android and iOS apps in Windows and Visual Studio requires a lot of programming knowledge and is not good for beginners. When Xamarin was launched, it attracted the attention of many developers owing to its efficiency. However, when beginners ask questions on sites such as SO, its popularity gradually declines owing to its application difficulty for beginners.

Similarly, React is a Facebook JavaScript library that builds the web application user interface efficiently. Riact was first launched on Facebooks news feed in 2011 and on Instagram in 2012. It was an open source at the JSConf US on May 2013. Social networking services (SNSs) such as Facebook and Instagram became popular around 20152016 when React changed from Terminal to Floating. Therefore, it seems that the popularity of the Framework changed based on its application on the popular SNSs.

Summary: If tags were the popular tool, their popularity would decline if they were difficult to use. Even if they were not a popular tool and they turn out to be an efficient tool, they will be popular among developers.

5. Conclusions

Critical development of a programming language, a program framework, or an operating system, depends on their ability to keep the community alive and attract more people to participate in discussions. SO, the worlds largest program Q&A platform, has enough questions and answers on a topic to solve users problems. This study applied the magnet and sticky population concepts to explore topics in SO. The results show that the numbers of participating topics are exploding with the development of computer technology. Even the most popular that did not attract peoples attention 10 years ago now attract a large number of participants. Under respective major categories, the most popular topics are still very popular after 10 years, and only a small number of languages or frameworks can become one of the most popular topics. This research provides a reference for enterprises to choose their main technology stack. It can also be used as a reference for computer science students to learn new technologies. The study (1) can predict the trend of computer technology in the next few years and (2) can identify easier technologies to access the questions and answers.

References

[1] C. for Community and C. E. Economic Development University of Wisconsin-Extension, "Community & economic development update," 2011, https://myemail.constantcontact.com/News-from-the-Center-for-Community---Economic-Development.html?soid=1104293309477\&aid=FnOfZtbhFgo.

[2] K. Yamashita, Y. Kamei, S. McIntosh, A. E. Hassan, and N. Ubayashi, "Magnet or sticky? measuring project characteristics from the perspective of developer attraction and retention," *Journal of Information Processing*, vol. 24, no. 2, pp. 339–348, 2016.

[3] X. Liu and H. Zhong, "Mining stackoverflow for program repair," in *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2018, pp. 118–129.

[4] S. Baltes, C. Treude, and S. Diehl, "Sotorrent: Studying the origin, evolution, and usage of stack overflow code snippets," 2019.