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

1st Kotori Hieda
Information Science and Electrical Engineering. Kyushu University
Fukuoka, Japan
hieda@f.ait.kyushu-u.ac.jp

2nd Yu Mingzhe
Information Science and Electrical Engineering. Kyushu University
Fukuoka, Japan
yumingzhe@f.ait.kyushu-u.ac.jp

3rd Olivier Nourry
Information Science and Electrical Engineering. Kysushu University
Fukuoka, Japan
onourry3@gmail.com

4th Yasutaka Kamei
Faculty of Information Science and Electrical Engineering. Kyushu University
Fukuoka, Japan
kamei@ait.kyushu-u.ac.jp

***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 a question related to Python and tags a post with the "Python" tag, developers interested in Python can easily find the post. 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 using the following metrics: (1) attractive, (2) fluctuating, (3) stagnant, and (4) terminal based on magnet values and sticky values. We analyze the data of table "Posts" which consists of approximately 42 million SO posts and the data of table "Users" which contains approximately 9 million rows of user information. Results reveal that: (1) There is a relationship between the magnetic and sticky values of a tag and the evolution of a project related to said tag, i.e., the creation of new software or the termination of the project. (2) The characteristics of the classified tags do not change much.**

***Index Terms*—magnet, sticky, tag, user migration, OSS census**

I. 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 the population migrated from other states and *sticky states* are states where a high percentage of adults have been living in that 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 popularity of developers have advantages. Many developers like to work with convenient and easy-to-use tools. To develop a project efficiently, a project needs to attract good developers over a long period of time.

In this study, we focus on new and existing topics of Stack Overflow (SO). Inspired by previous studies [2], we apply the magnet and the sticky metrics to the topics collected in SO. The magnet metric is the number of existing developers who remain involved with a specific topic. We examined magnet and sticky values of tags by classifying them using one of three categories: *programming language*, *framework*, and *environment*. We also look at the evolution of multiple software and web services companies. If any change or evolution is discovered, we investigate whether there is a relationship between change or evolution and magnet and sticky values.

We address the following two research questions:

(RQ1) What are the magnet and sticky values of popular SO tags?

We find that in many cases, the sticky value is higher than the magnet value. In addition, the decrease rate was higher for the magnet value than it was for the sticky value.

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

When the status of a tag changes in four types, there may have been some change or evolution in that tag or another tag related to that tag.

II. 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 [1], we use the magnet and sticky metrics to illustrate the migratory trends of U.S. citizens. The PRC defines magnet states as states where a large proportion of adults are from other states and defines sticky states as state where a high share of the adults who were born there live there now. These definitions are for population studies where a single adult can only live in one state at a time, however, so these definitions are 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 redefined magnet and sticky so that they can be applied in SO.

Magnet. Magnet topics attract a large proportion of new users; thus, the magnet value is calculated as the percentage of new users who ask or answer questions during a year.

Topic	User	2017	2018	Magnet2017	Magnet2018	Sticky2018
1	A	●	●	3/3	1/2	2/3
	B	●	●			
	C	●				
	D					
	E		●			
2	A			2/3	2/2	1/2
	B	●				
	C	●	●			
	D		●			
	E		●			

Fig. 1. Example of magnet and sticky values definition

Sticky. In sticky topics, users continue to participate in the same topic. Thus, we calculate the stickiness of a topic as the proportion of users who remain involved in the topic ' s discussion over the years.

Magnet and Sticky in SO

SO content is made of user comments, questions and answers [3] called Posts in the SOTorrent [4] database. Each question has one or more tags that separate the question into different topics. Each post has an author, a participant and a question. We defined who is the participant of which topic as follows. For example, user A answers a question with a Java tag, user B answers with an Apache tag, and user C answers with a Linux tag. At this time, User C is a participant in Java, Apache, and Linux topics.

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 three questions (α , β , γ) and five users (A, B, C, D, E). Users A, B and C registered in 2017 and users D and E register in 2018 [2] as shown in Figure 1.

Filled black circles in Figure 1 represent user activities (e.g., creating questions, answering questions, and adding comments to the questions) in a topic during specific period.

To calculate the magnet metric, we observe three new users who registered their account in 2017 (A, B, C), who registered their account in 2017 and participate in topic 1 and two users (B, C) who participate in topic 2. In 2017, the magnet value of topic 1 was 3/3 and the magnet value of topic 2 was 2/3.

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

Example (Merging similar subjects into one topic). We merge subjects (i.e., tags) that belong to analogous subjects into one topic. We consider different version numbers (e.g., tag "Python-2.7" and tag "Python-3.6") as one of the common examples of analogous tags.

We consider all different versions of a tool, as well as derived tools, to belong to the same topic. 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."

Question	Tag1	Tag2	Topic1	Topic2
α	python	python-3.x	python	
β	python-2.7	numpy	python	numpy
γ	c++17	boost-asio	c++	boost

Fig. 2. Example of a merge of tags belonging to analogous subjects

Figure 2 shows that question α has tags "python" and "python-3.x" and question β has the tag "python-2.7." According to our merge rule, they all belong to topic "python." The question γ has the tag "c++17," showing that it belongs to topic "c++." Therefore, question β also belongs to topic "numpy" and question γ also belongs to topic "boost."

III. Dataset

We analyze the SO dataset (SOTorrent) provided by Sebastian Baltes et al. [5]. 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 and code blocks. It connects code snippets from SO posts to other platforms by aggregating URLs from surrounding text blocks and comments, and by collecting references from GitHub files to SO posts.

The dataset consists of 20 different data tables. However, we analyze the data of table *Posts* which consists of approximately 42 million posts from SO and table *Users* which contains approximately 9 million rows of User information from July 2008 to September 2018. For the purpose of this study, we looked at the users, tags and the date each question was posted in order to calculate magnet and sticky value. Moreover, we only considered users who ask or answer questions in SO. Those who only comment or like/dislike questions and answers were excluded to narrow down those who are really interested in the topic.

IV. Study Results

In this section, we provide answers to the following questions:

A. (RQ1) What are the magnet and sticky value of popular SO tags?

Approach. We calculated magnet and sticky values as defined in Section II. 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 understanding the tags, we can discover the interests of developers.

Fluctuating: Tags with a high magnet and low sticky value.

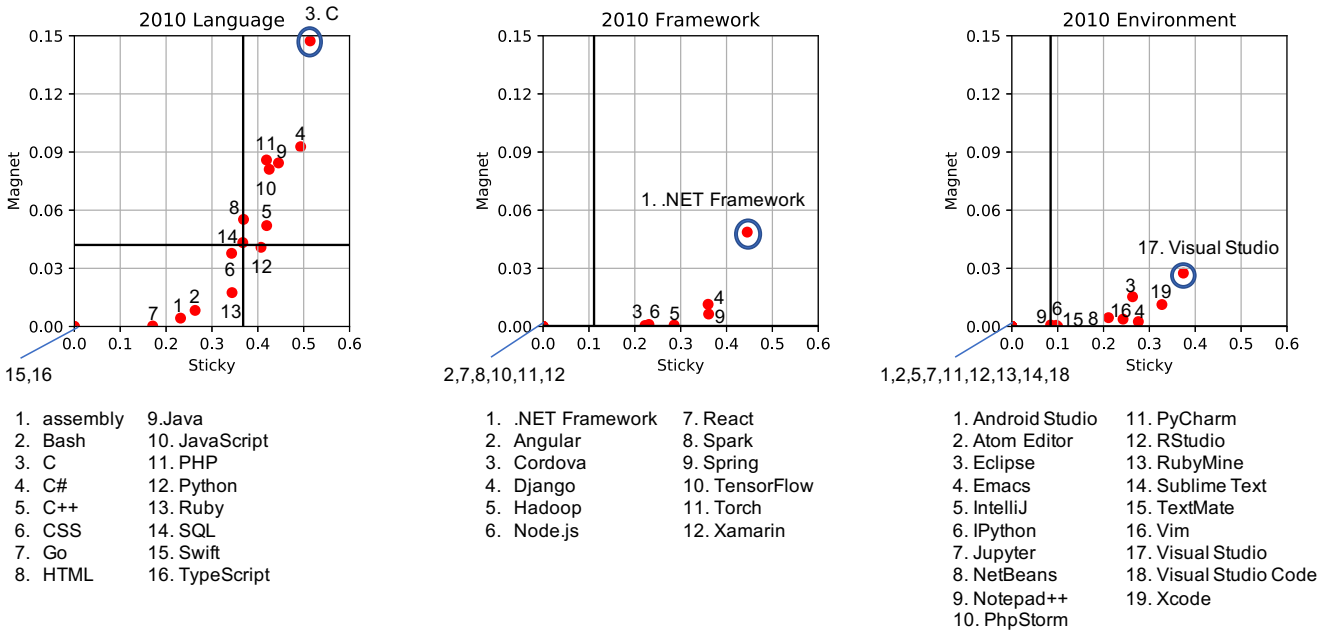

Fig. 3. Distribution of Magnet and Sticky values in Programing Language, Framework and Environment

TABLE I
AVERAGE QUADRANT TRANSITION RATE

Language

	Attractive	Fluctuating	Stagnant	Terminal	*
Attractive	81.3	10.6	5.0	3.1	0.0
Fluctuating	12.5	87.5	0.0	0.0	0.0
Stagnant	7.3	0.0	86.5	6.3	0.0
Terminal	3.1	0.0	5.0	91.9	0.0
*	0.0	0.0	25.0	50.0	25.0

Framework

	Attractive	Fluctuating	Stagnant	Terminal	*
Attractive	92.9	7.1	0.0	0.0	0.0
Fluctuating	20.0	50.0	0.0	30.0	0.0
Stagnant	30.0	0.0	20.0	50.0	0.0
Terminal	0.0	0.0	33.3	66.7	0.0
*	0.0	0.0	9.2	14.6	76.3

Environment

	Attractive	Fluctuating	Stagnant	Terminal	*
Attractive	83.1	9.3	3.6	4.1	0.0
Fluctuating	24.8	60.5	0.0	14.8	0.0
Stagnant	9.2	4.2	69.6	17.1	0.0
Terminal	1.8	0.0	8.1	90.1	0.0
*	2.5	8.3	12.5	13.3	63.3

This tag attracts people but only for a short period of time. Developers lose interests over time.

Stagnant: Tags with a low magnet and high sticky value. This tag does not attract many new users, but it can retain existing users.

Terminal: Tags with low magnet and sticky value. This tag can neither attract new 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 Section II, the sticky value depends on the number of users that remain involved in a tag's discussion over time. To answer RQ1, we calculated the sticky value for nine years from 2009 to 2018. We used information posted by users only those who posted questions or answers with tags in SO. The magnet and sticky values depend on the number of new tag users, but, if the number of new tag users in a year is too low, the values will be too small to accurately classify tag status as attractive, fluctuating, stagnant, and terminal. To remove noise, we set thresholds for each topic. If the magnet and sticky values are less than the threshold, we assign an asterisk to its magnet and sticky value.

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 programming language, framework, and environmental tags[2], revealing that the magnet value is lower than the sticky value. U.S. citizens are more likely to live in the same house for an extended period because it is not easy to move into new homes. Similarly, developers are more likely to continue developing the same type of content on one project than to keep changing project because it is not easy to work on new inexperienced topics. So sticky value is higher than magnet value same as PRC results.

Summary. Tags with a high magnet value are easy to use even for beginners. A tag familiar to beginners such as C or Visual Studio have higher magnet and sticky values.

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

Approach: From 2010 to 2018, we calculated the probability that the tag will move in the quadrant from one year to the following year. For example, there were six attractive tags in 2010. Of the six attractive tags in 2010, five 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 I shows the transition probability of tag states for 2010 – 2018. Since the probability of transition from the a state to the same state is the highest, you can see that the popularity of each topic tends to be stable. Since the probability of transition from the terminal state to another state is the lowest, you can see that tags that lose in popularity may remain unpopular.

Manual analysis: Table II shows the transition of each tag. In the framework category, it reveals how the tags move in the quadrant. From this, you can see that Visual Studio [6] and Xcode [7] have been attractive for a long time. Visual Studio is an integrated development environment for Windows and Xcode is an integrated development environment for Apple. Since these IDEs are likely to be used by beginners in their respective environments, it is common to see questions about them on SO. In addition, we can see that Jupyter [8], PyCharm [9] and RStudio [10] have gained popularity as opposed to the decrease in popularity of Sublime Text [11] and IPython [12]. This is likely because the popularity of Python has increased in recent years and because Jupyter has made IPython available in multiple languages.

Summary: The transition probability of tags tends to be stable. This indicates that once a tag has become popular enough, the number of users of that tag will not significantly reduce over time. This also indicates that tags that lose in popularity may remain unpopular. When the state of a tag changes, another topic may have evolved or a new topic may have appeared within the same category.

[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.

V. Conclusions

Critical development of a programming language, framework, or environment depends on developers' ability of a project to keep the community alive and attract more people to participate in the development. This study applied the magnet and sticky population concepts to explore topics in SO. The results show that the number of participating topics are exploding with the development of computer technology. Even the most popular themes that did not attract people's attention 10 years ago now attract a large number of participants. Under respective major categories: language, framework, and environment, the most popular topics are still very popular after 10 years, and only a small number of these categories 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 show good tools to use in the current era for your development.

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, L. Dumani, C. Treude, and S. Diehl, "Sotorrent: Reconstructing and analyzing the evolution of stack overflow posts," in *Proceedings of the 15th International Conference on Mining Software Repositories*. ACM, 2018, pp. 319–330.

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

[6] B. Johnson, *Professional Visual Studio 2012*. John Wiley & Sons, 2012.

[7] F. Tisato and R. Zicari, "The xcode machine," *ACM Sigsmall Newsletter*, vol. 10, no. 1, pp. 12–23, 1984.

[8] J. M. Perkel, "Why jupyter is data scientists' computational notebook of choice," *Nature*, vol. 563, no. 7732, pp. 145–147, 2018.

[9] Q. N. Islam, *Mastering PyCharm*. Packt Publishing Ltd, 2015.

[10] J. Allaire, "Rstudio: integrated development environment for r," *Boston, MA*, vol. 770, 2012.

[11] D. Peleg, *Mastering Sublime Text*. Packt Publishing Ltd, 2013.

[12] F. Pérez and B. E. Granger, "Ipython: a system for interactive scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 21–29, 2007.

TABLE II
QUADRANT TRANSITION OF FRAMEWORK 2010 - 2018

Language Name	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	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating
C#	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Attractive	Fluctuating	Attractive
C++	Attractive	Attractive	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal
CSS	Terminal	Attractive	Attractive	Attractive	Attractive	Attractive	Fluctuating	Fluctuating	Fluctuating
Go	Terminal	Terminal	Terminal	Terminal	Terminal	Stagnant	Stagnant	Stagnant	Stagnant
HTML	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Attractive	Attractive	Attractive
Java	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
JavaScript	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
PHP	Attractive	Fluctuating	Attractive	Attractive	Attractive	Fluctuating	Fluctuating	Fluctuating	Fluctuating
Python	Stagnant	Stagnant	Stagnant	Stagnant	Attractive	Attractive	Attractive	Attractive	Attractive
Ruby	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant
SQL	Attractive	Stagnant	Attractive	Attractive	Stagnant	Stagnant	Terminal	Stagnant	Terminal
Swift	*	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant
TypeScript	*	*	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal

Framework Name	2010	2011	2012	2013	2014	2015	2016	2017	2018
.NET Framework	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Fluctuating	Attractive
Angular	*	*	*	*	*	Stagnant	Attractive	Attractive	Attractive
Cordova	Attractive	Attractive	Attractive	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Terminal	Terminal
Django	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
Hadoop	Attractive	Attractive	Attractive	Attractive	Attractive	Fluctuating	Terminal	Terminal	Terminal
Node.Js	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
React	*	*	*	*	*	*	*	*	Terminal
Spark	*	*	*	*	*	*	*	*	Terminal
Spring	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
TensorFlow	*	*	*	*	*	*	Stagnant	Attractive	Attractive
xamarin	*	*	*	*	*	Stagnant	Terminal	Terminal	Terminal
Xamarin	*	*	Terminal	Stagnant	Stagnant	Terminal	Terminal	Stagnant	Terminal

Environment Name	2010	2011	2012	2013	2014	2015	2016	2017	2018
Android Studio	*	*	*	Fluctuating	Attractive	Attractive	Attractive	Attractive	Attractive
Atom Editor	*	*	*	*	*	Terminal	Terminal	Terminal	Terminal
Eclipse	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Fluctuating	Attractive	Attractive
Emacs	Attractive	Attractive	Attractive	Attractive	Attractive	Stagnant	Stagnant	Stagnant	Stagnant
IntelliJ	*	*	*	*	*	Terminal	Terminal	Terminal	Terminal
IPython	Attractive	Fluctuating	Attractive	Attractive	Attractive	Attractive	Attractive	Terminal	Terminal
Jupyter	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Stagnant	Terminal	Attractive
NetBeans	Attractive	Attractive	Attractive	Attractive	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating
Notepad++	Attractive	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Terminal	Terminal
PhpStorm	Stagnant	Stagnant	Stagnant	Terminal	Stagnant	Terminal	Stagnant	Stagnant	Terminal
PyCharm	*	Stagnant	Stagnant	Stagnant	Stagnant	Stagnant	Fluctuating	Attractive	Attractive
RStudio	*	*	Stagnant	Stagnant	Stagnant	Attractive	Stagnant	Attractive	Attractive
RubyMine	*	Attractive	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal
Sublime Text	*	*	Fluctuating	Attractive	Fluctuating	Fluctuating	Fluctuating	Fluctuating	Terminal
TextMate	Attractive	Attractive	Fluctuating	Terminal	Terminal	Terminal	Terminal	Terminal	Terminal
Vim	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
Visual Studio	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive
Visual Studio Code	*	*	*	*	*	Stagnant	Stagnant	Attractive	Attractive
Xcode	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive	Attractive