

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

1<sup>st</sup> Kotori Hieda

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

2<sup>nd</sup> Yu Mingzhe

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

3<sup>rd</sup> Olivier Nourry

Information Science and Electrical  
Engineering, Kyushu University  
Fukuoka, Japan  
onourry3@gmail.com

4<sup>th</sup> 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 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 using the following metrics: (1) attractive, (2) fluctuating, (3) stagnant, 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: (1) There were some historical events in IT such as the launch of new tools and the termination of services when there were characteristics in the transition of magnet and sticky values. (2) The characteristics of these 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*: where a high percentage of the population migrated from other states.

*Sticky*: Sticky states are states where a high percentage of adults currently living there were also born in that state.

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 make a project better, the 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 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 the values of tags “magnets” and tags “sticky” by classifying them using one of three categories: *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.

We address the following two research questions:

**(RQ1) What are typical values of magnet and sticky in SO?**

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 sticky value.

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

When the status of a tag changes in four types, something may have occurred in 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, 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 [1]. From the magnet metric, the proportion of adults residing in the magnet states were not born in the state. PRC defines the sticky states as where a high share of the adults who were born there live there now. These definitions are for a study of populations that single adult can only occupy 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 and Sticky in SO*

SO content is made of user comments, questions and answers [3] called Posts in the SOTorrent [4] database. Each

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

question has one or more tags that separate the question into different topics. Each posts have an author, a participant and a question. We also define activity of asking and answering on some topic as a discussion topic. 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.

**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.

**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 continue to discuss across years.

**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 ( $\alpha$ ,  $\beta$ ,  $\gamma$ ) 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 of a certain year. For example, while User A has activities in Topic 1 of the years 2017 and 2018, but he/she has no activities in Topic 2.

To calculate the magnet metric, we observe three new users who registered their accounts 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”) are one of the common examples of analogous tags.

We considered all the derived tools of a tool, such as a

Question	Tag1	Tag2	Topic1	Topic2
$\alpha$	python	python-3.x	python	
$\beta$	python-2.7	numpy	python	numpy
$\gamma$	c++17	boost-asio	c++	boost

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

special version of a tool, or separated for engineers and the general public, etc. to be the same tool. 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  $\alpha$  has tags “python” and “python-3.x,” question  $\beta$  has the tag “python-2.7.” According to our merge rule, they all belong to topic “python.” The question  $\gamma$  has the tag “c++17,” showing that it belongs to topic “c++.” Therefore, question  $\beta$  also belongs to topic “numpy” and question  $\gamma$  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 and provides access to the version history of SO content of whole posts and individual text or code blocks.

The dataset consists of 20 different tables stored in data.

However, we analyze the data of table *Posts* which consists 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 the date a question was posted. Moreover, we only consider user who ask or answer questions in SO; those who comment or like/dislike questions or answers are excluded from the statistics.

### IV. STUDY RESULTS

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

A. (RQ1) What are the magnet value and sticky value in SO?

**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. This tag attracts people, but it is short-term. Excellent developers will be disinterested.

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

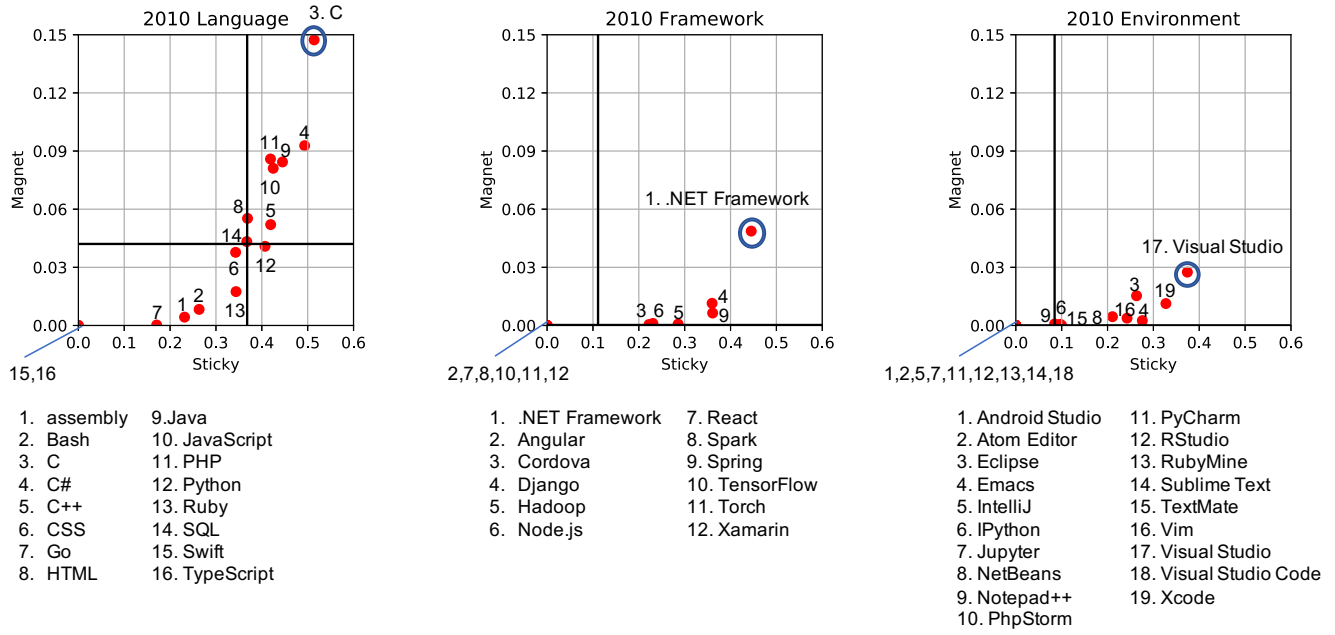


Fig. 3. Distribution of Magnet and Sticky values in Programming Language, Framework and Environment

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

existing users.

**Terminal:** Tags with the 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 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 who asked questions and answers with tags in SO from 2009 to 2018. 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 remove noise, we set thresholds for each topic. If the magnet and sticky values are less than the threshold, the value is 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 <sup>1</sup>. 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,

<sup>1</sup><https://insights.stackoverflow.com/survey/2018>

and environmental tags<sup>2</sup>, revealing that the magnet value is lower than the sticky value. Our results are similar to the findings of the PRC. For example, the U.S. citizens are more likely to live in the same house than to move/change houses; it is easier for developers to continue developing the same content than change.

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

*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 that the proportion of vertical axis is in the quadrant for 2010 - 2018. From the table, the ratio of tags is almost 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 fluctuating, 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 II shows the transition of each tag in the framework, revealing 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 are the first IDEs that beginners will use in their respective environments, it is likely that new users tend to ask questions about them in 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 because the popularity of Python has increased in recent years and that Jupyter has made IPython available in multiple languages.

**Summary:** Since there are many beginners asking questions on SO, IDE tags tend to be attractive for beginners. Also, with the emergence of easier-to-use tools, the popularity of what were once attractive such as Sublime Text and IPython decreases, and the popularity of new easy-to-use tools such as Jupyter, PyCharm, and RStudio increases.

## V. CONCLUSIONS

Critical development of a programming language, framework, and environment depends on their ability to keep the community alive and attract more people to participate in 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, 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 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.

<sup>2</sup>We choose the year 2010 because it is the first year for which yearly data of sticky value can be obtained.

TABLE II  
QUADRANT TRANSITION OF FRAMEWORK 2010 - 2018

[illegible]

Framework	Name	2010	2011	2012	2013	2014	2015	2016	2017	2018
.NET Framework		A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	Fluctuating	A ttractive
	Angular	*	*	*	*	*	S tagnant	A ttractive	A ttractive	A ttractive
	Cordova	A ttractive	A ttractive	A ttractive	Fluctuating	Fluctuating	Fluctuating	Fluctuating	T erminal	T erminal
	Django	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive
	Hadoop	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	Fluctuating	T erminal	T erminal	T erminal
	Node.js	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive
	React	*	*	*	*	*	*	*	*	T erminal
	Spark	*	*	*	*	*	*	*	*	T erminal
	Spring	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive	A ttractive
	TensorFlow	*	*	*	*	*	*	S tagnant	A ttractive	A ttractive
	xamarin	*	*	*	*	*	S tagnant	T erminal	T erminal	T erminal
	Xamarin	*	*	T erminal	S tagnant	S tagnant	T erminal	T erminal	S tagnant	T erminal

[illegible]