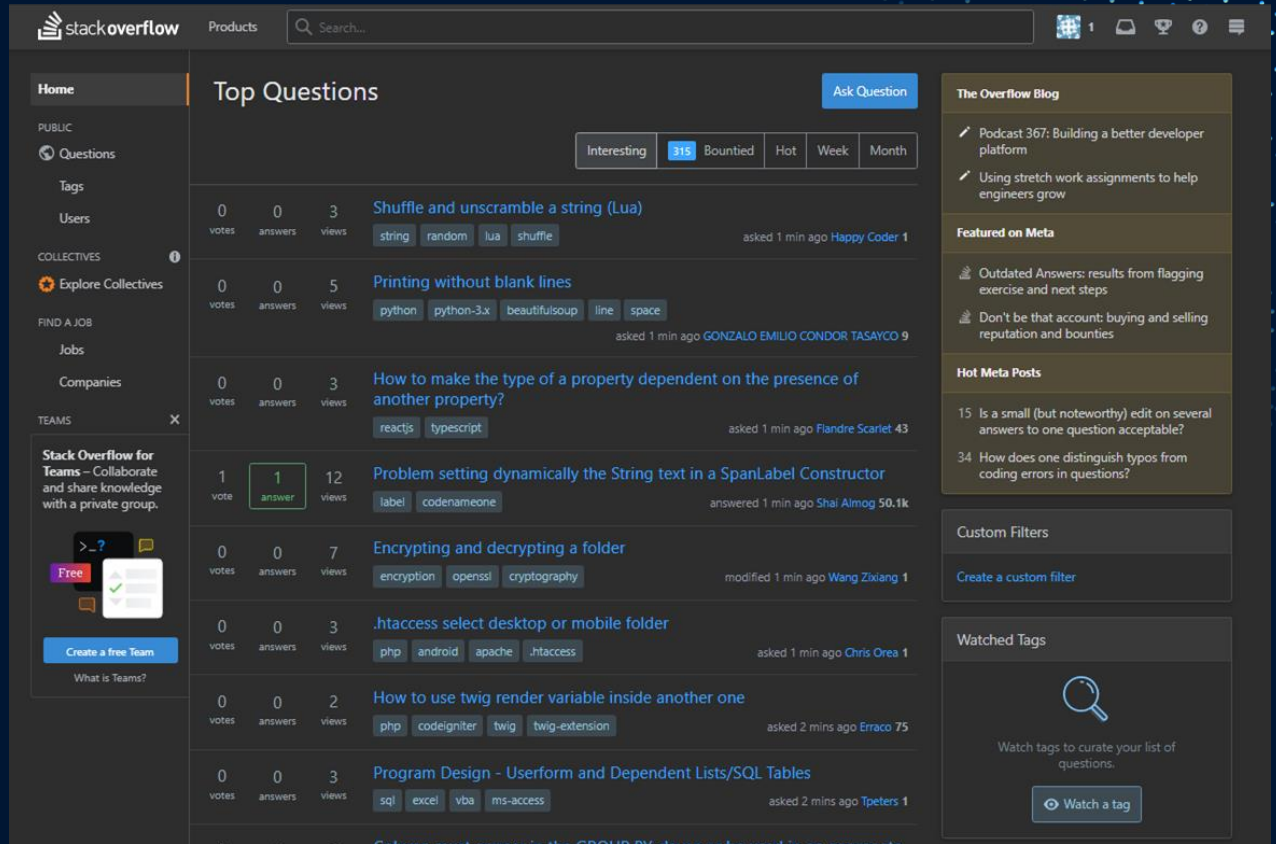
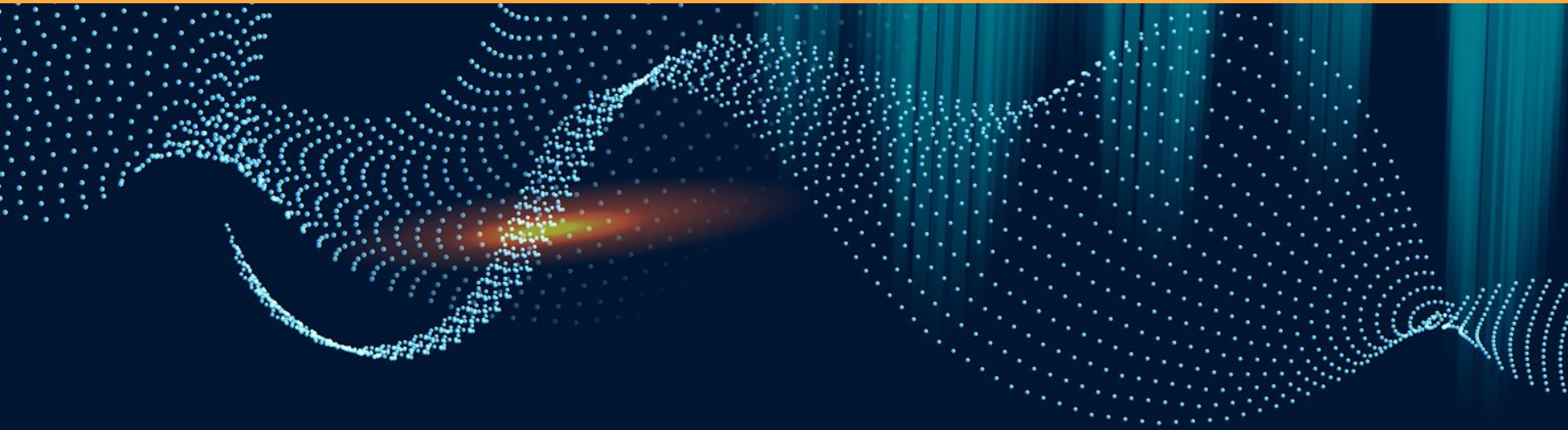
Analyzing *Stack Overflow* Community Posts to Automate Knowledge Organization

U of T UnERD 2021

Arjun Sridharkumar, **Hamza Dugmag**,
Iftekhar Ahmed, Shurui Zhou

StackOverflow.com

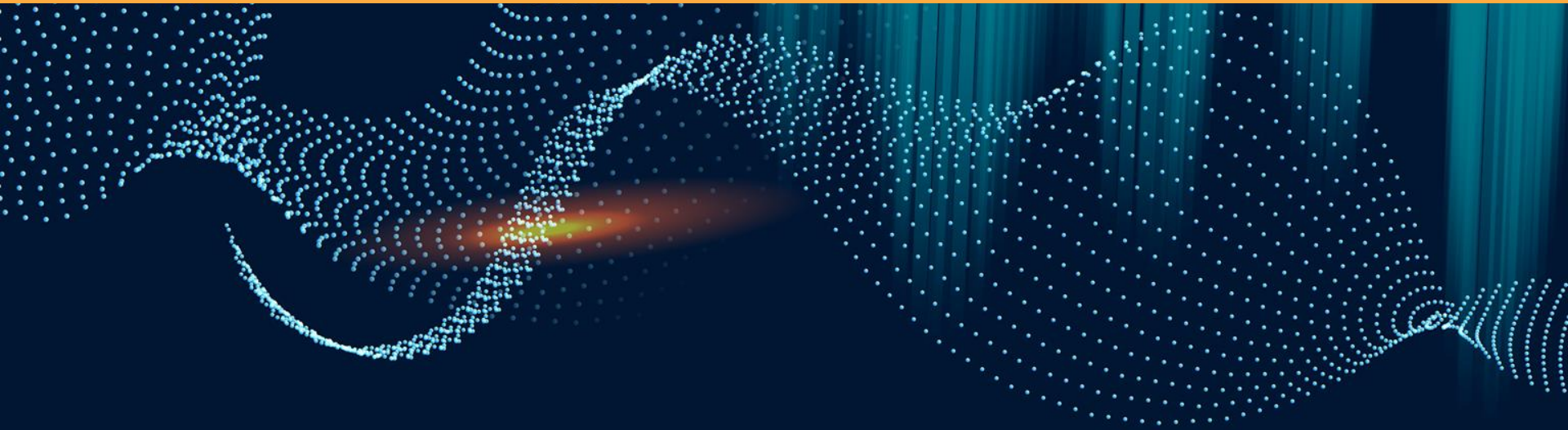




22M Questions

70% of which are answered as of July 2021 [1]

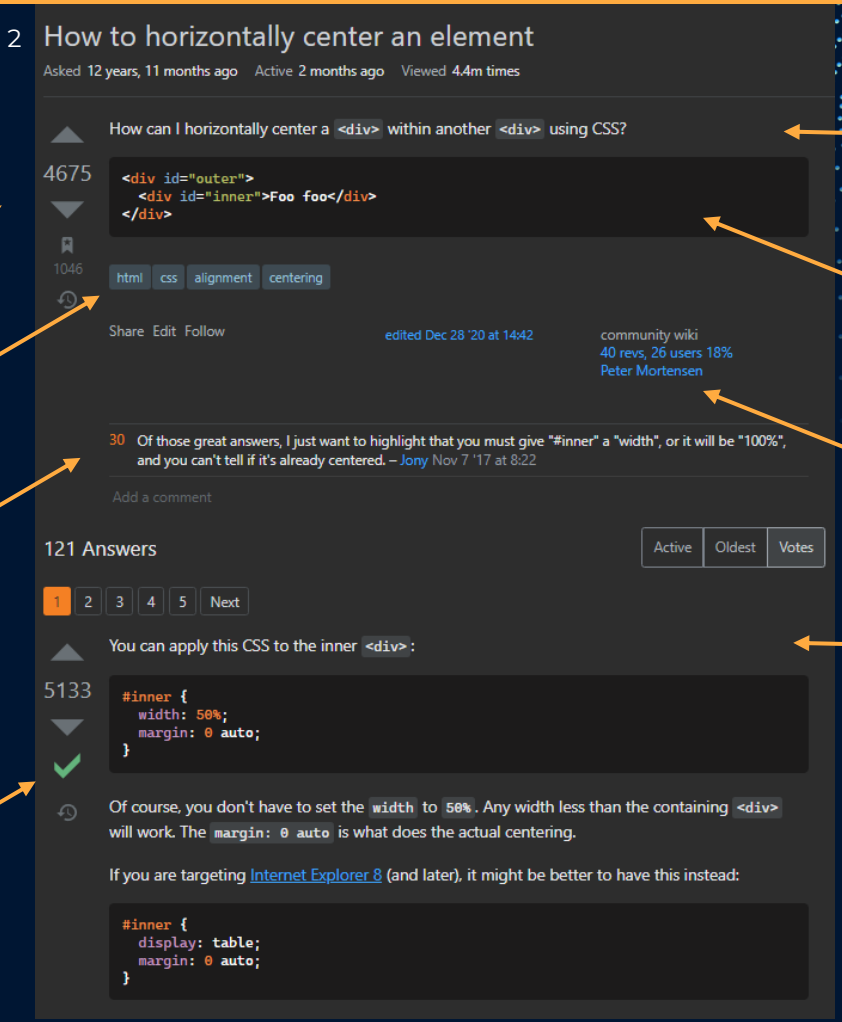
[1] https://stackexchange.com/sites?view=list#questions



15M | Users

As of July 2021 [1]

[1] https://stackexchange.com/sites?view=list#questions



Upvotes

Tags

Comment

Accepted answer

Question

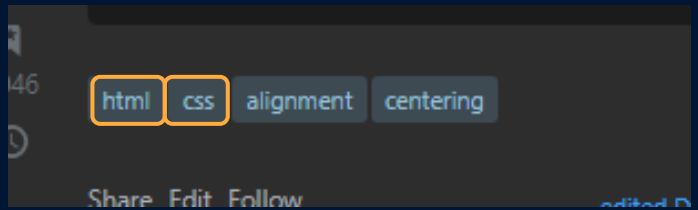
Code snippet

Author

Answer

[2] https://stackoverflow.com/questions/114543/how-to-horizontally-center-an-element

Tag Wiki



Similar to official documentations of programming technologies

[3] https://stackoverflow.com/tags/html/info

About html

Ask Question

Tag Info

Info | Newest | 11 Bountied | Frequent | Votes | Active | Unanswered

HTML (HyperText Markup Language) is the markup language for creating web pages and other information to be displayed in a web browser. Questions regarding HTML should include a minimal reproducible example and some idea of what you're trying to achieve. This tag is rarely used alone and is often paired with [CSS] and [javascript].

HTML (HyperText Markup Language) is the markup language used for structuring web pages and other information to be displayed in a web browser. HTML describes the structure of a web-page semantically along with cues for presentation, making it a markup language rather than a programming language. A browser 'renders' HTML in conjunction with CSS, which defines the 'style' (colors, fonts, layout, etc.) and JavaScript, which defines interactive and dynamic elements, adding style and behavior to the pages.

https://html.spec.whatwg.org/multipage/ is the canonical HTML specification.

HTML (HyperText Markup Language) is the main markup language for creating web pages and other information to be displayed to humans in a web-browser.

It was invented by Sir Tim Berners-Lee while developing the first Web browser at CERN to enable researchers to share their findings and formally released in June 1993. The original "HTML Tags" were first publicly mentioned by Berners-Lee in 1991 and borrowed the syntax from CERN's SGML-based documentation standard. The latest version for HTML is HTML5.2.

HTML elements form the building blocks of all web-pages. HTML allows images and objects to be embedded in a page. It references styles and scripts and carries meta-data. It can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript, which affects the behavior of HTML web pages. Web pages written in different programming languages (PHP, JSP, VF, ASP.NET etc.) get rendered as HTML in a browser.

HTML is a hierarchical (tree-structured) markup language. That is, an item might be a descendant of another item, which is its ancestor. However, if item2 is a descendant of item1, then they have an additional special relation: item2 is *inside* of item1, or item1 is *wrapped around* item2.

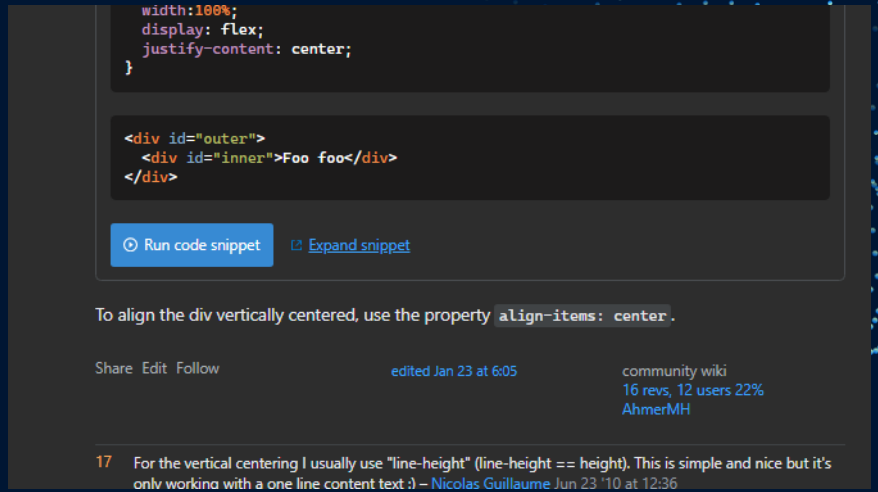
Syntax

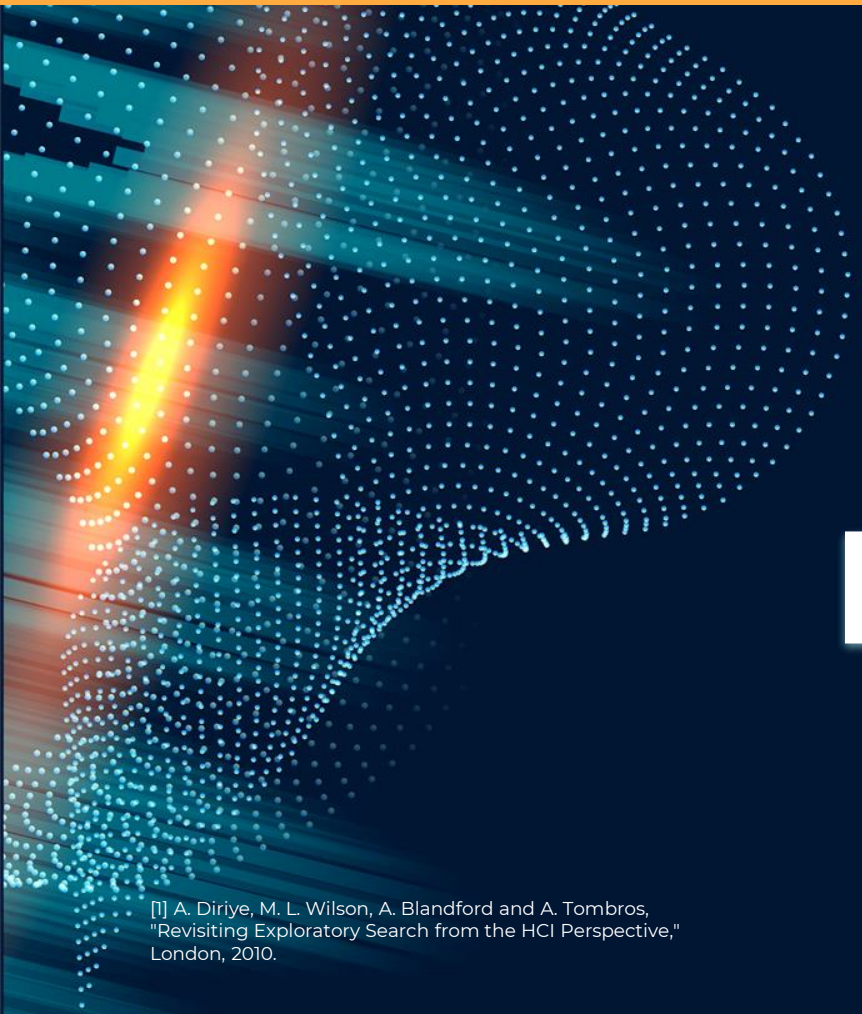
HTML is written in the form of elements consisting of tags (and their attributes) enclosed in angle brackets (e.g., `<html>`).

HTML tags most commonly come in pairs. The first is known as the *opening tag* and the second, which

Community Wiki Posts

Multiple collaborators can contribute to a question, answer, or tag wiki

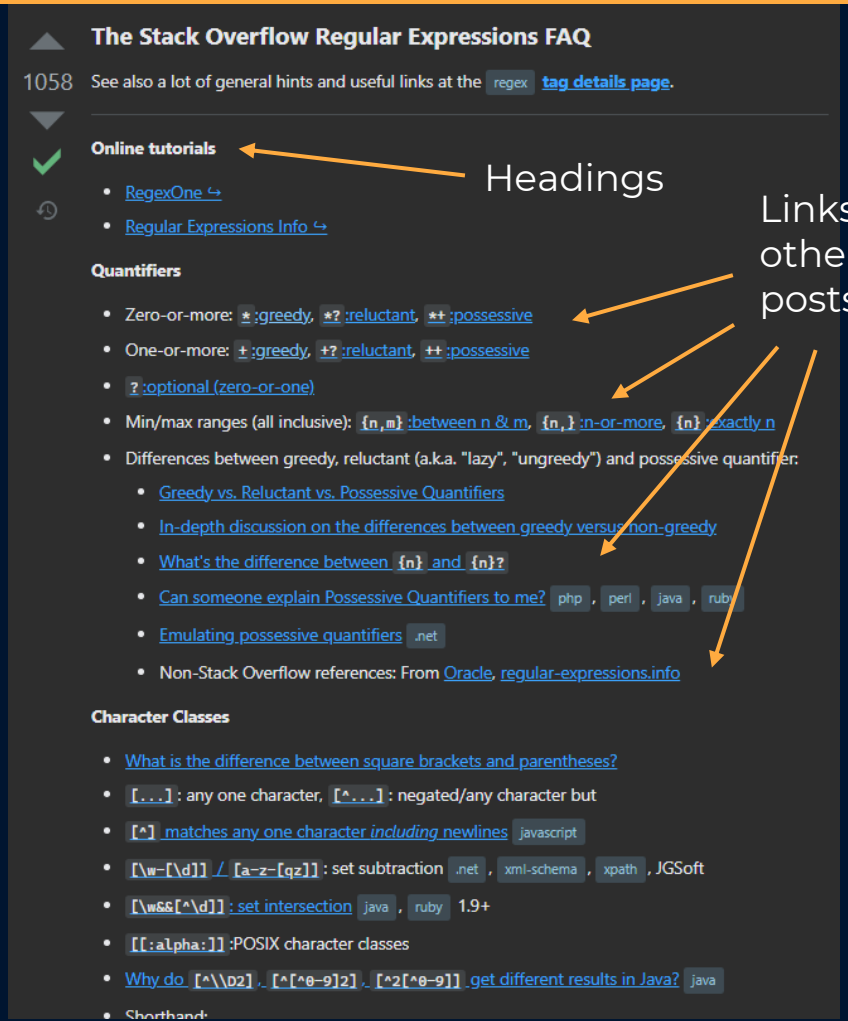




However...

the Q&A format makes it difficult to retrieve desired knowledge via unknown-item searches [1].

[1] A. Diriye, M. L. Wilson, A. Blandford and A. Tombros, "Revisiting Exploratory Search from the HCI Perspective," London, 2010.



“Structured Posts”

Community posts which aim to organize the *Stack Overflow* website

[4] https://stackoverflow.com/a/22944075/10757178

5K/22M

questions are linked in community posts, indicating that *Stack Overflow* is largely unorganized [5]

[5] https://data.stackexchange.com/stackoverflow/query/new

Research Project Goals

Analysis

What are the characteristics of community and structured posts?

Prototyping

How can we create a tooling method that automatically generates structured posts?

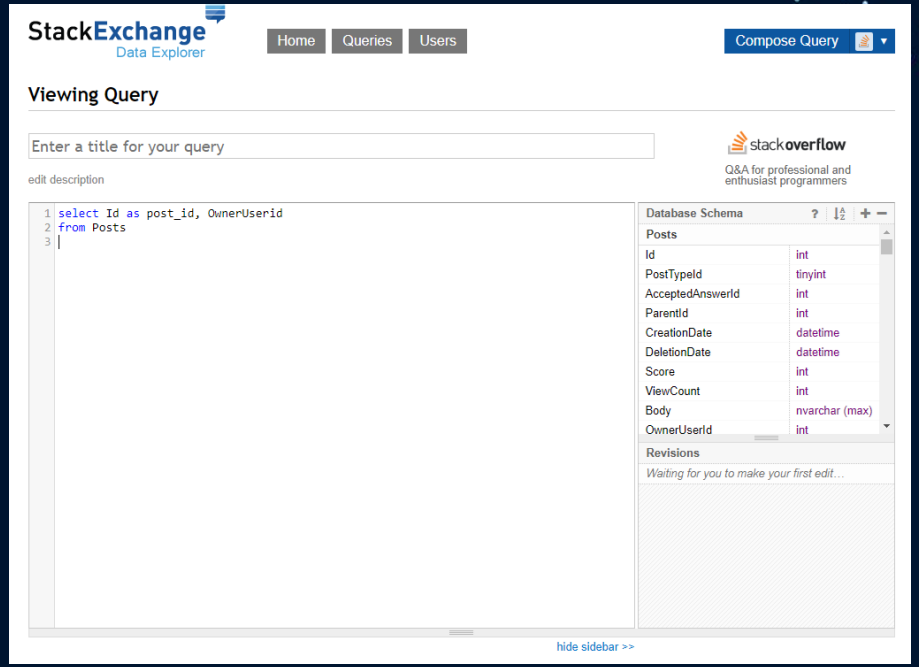
Hypothesis

It is expected that there are shared features among high-quality structured posts

Approach

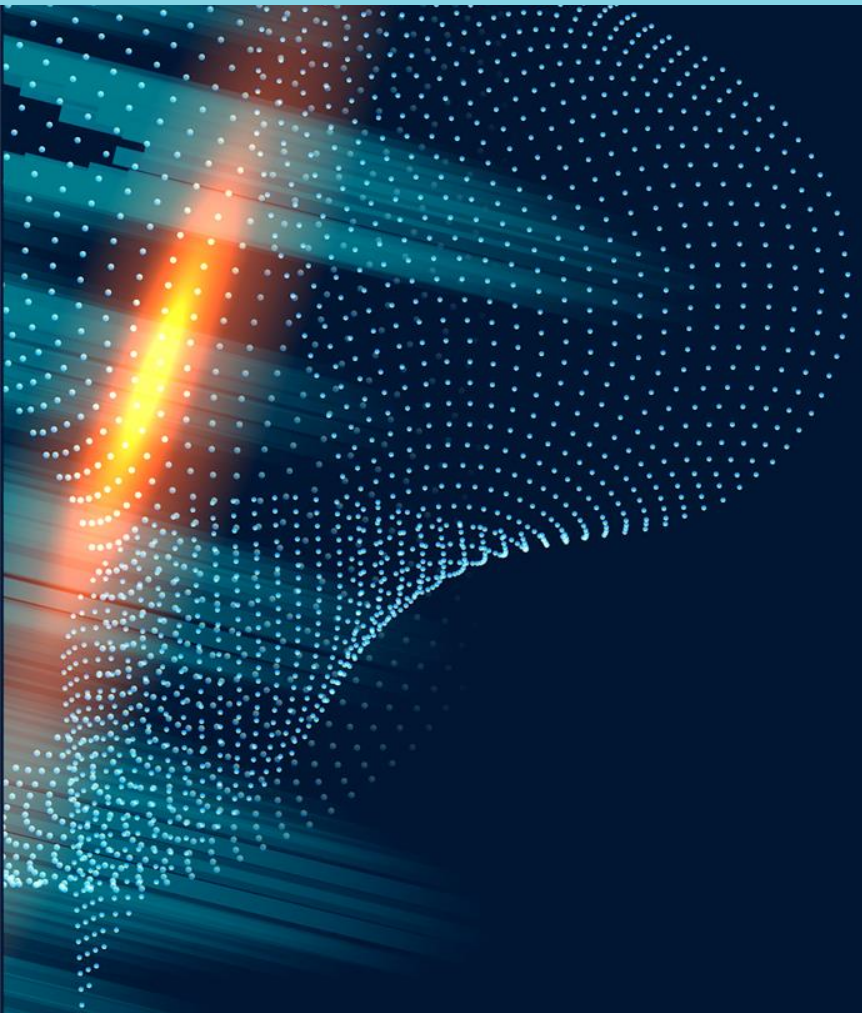
Unsupervised clustering groups similar posts together

Data Acquisition



Stack Overflow data dump since 2008 [5]

[5] https://data.stackexchange.com/stackoverflow/query/new



126K

community posts created since 2008

Post Properties and Features

Length

Text Length
No. of *STO* Links
No. of External Links
Total No. of Links
Ratio of *STO* Links

Activity

No. of Edits
Creation Date
Last Update Date
Inactivity Time
No. of Contributers

Popularity

No. of Upvotes

Structure

No. of Headings
Avg. No. of *STO* Links /Heading

Raw Data

	post_id	Text Length	No. of SO Links	No. of ALL Links	ALL - SO	SO / ALL	No. of Revisions	Creation Date	Last Update Time	No. of Contributers	Creation - Updated	No. of Upvotes	No. of Headings	Avg. No. of SO Links / Heading
0	1587.0	59.0	0.0	3.0	3.0	0.0	1.0	4760.0	4760.0	1.0	0.0	14.0	0.0	0.0
1	1660.0	7.0	0.0	8.0	8.0	0.0	5.0	4760.0	4728.0	2.0	31.0	91.0	3.0	0.0
2	1752.0	107.0	0.0	0.0	0.0	0.0	5.0	4756.0	4756.0	1.0	0.0	15.0	0.0	0.0
3	2376.0	40.0	0.0	0.0	0.0	0.0	2.0	4756.0	4756.0	2.0	0.0	0.0	0.0	0.0
4	2420.0	97.0	0.0	3.0	3.0	0.0	5.0	4756.0	4352.0	4.0	404.0	35.0	1.0	0.0

First five unscaled data points used after data analysis

	Text Length	No. of SO Links	No. of ALL Links	ALL - SO	SO / ALL	No. of Revisions	Creation Date	Last Update Time	No. of Contributers	Creation - Updated	No. of Upvotes	No. of Headings	Avg. No. of SO Links / Heading
0	0.010269	0.0	0.009583	0.013947	0.0	0.003731	1.000000	1.000000	0.008926	0.000212	0.002283	0.000000	0.0
1	0.001219	0.0	0.025558	0.037201	0.0	0.018646	1.000000	0.993652	0.017853	0.006779	0.005367	0.048401	0.0
2	0.018616	0.0	0.000000	0.000000	0.0	0.018646	0.999023	0.999023	0.008926	0.000212	0.002323	0.000000	0.0
3	0.006962	0.0	0.000000	0.000000	0.0	0.007462	0.999023	0.999023	0.017853	0.000212	0.001722	0.000000	0.0
4	0.016876	0.0	0.009583	0.013947	0.0	0.018646	0.999023	0.914062	0.035706	0.085754	0.003124	0.016129	0.0

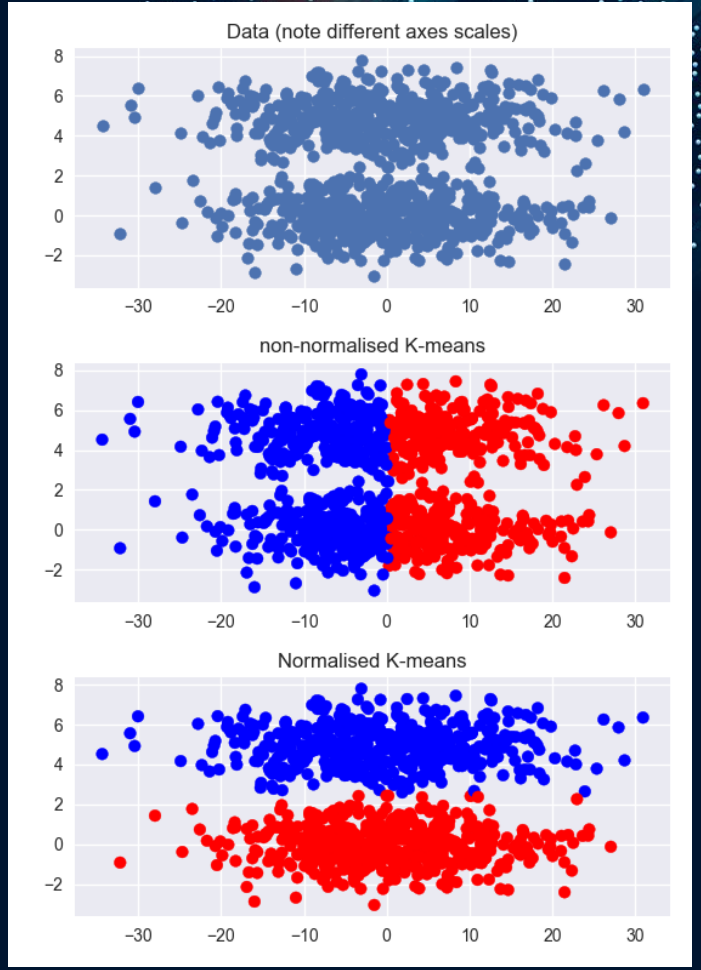
First five normalized data points used in clustering

Note: the following data is from a random sample of 6K posts since the script is very slow and we have yet to migrate all the data to our server

Data Normalization

Normalizing data equally weighs all features [2].

Don’t want distances to be a factor



[2] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft computing*, vol. 97, 2020.

[6] https://stats.stackexchange.com/a/283941

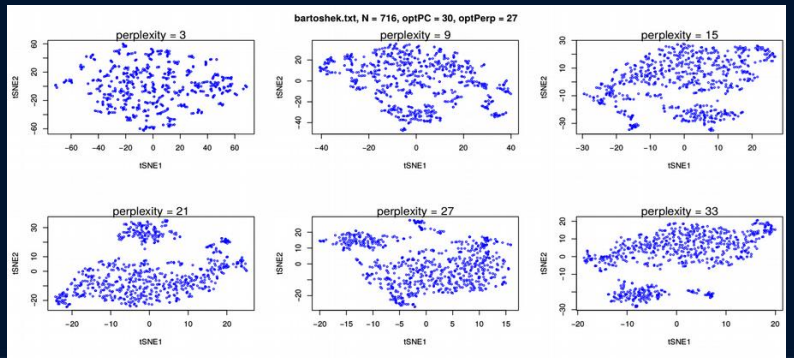
Visualizing High-Dimensional Data with T-Distributed Stochastic Neighbor Embedding (TSNE)

TSNE represents high-dimensional data in 2D by creating a probability distribution of how close data points are to each other [3].

Hyperparameter: Perplexity

Attention between local and global aspects of the data

Perplexity ~ $N^{0.5}$ [4].

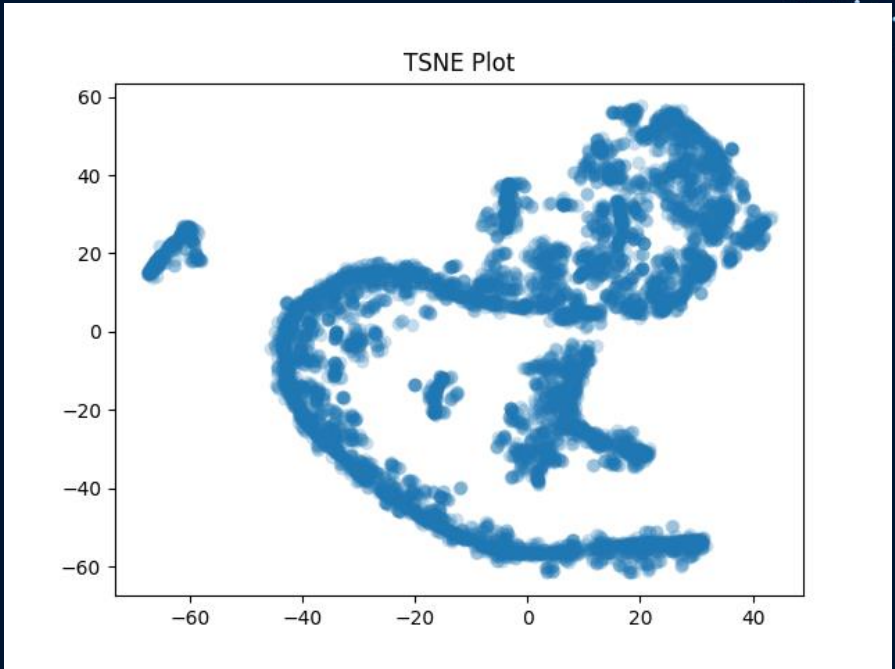


[3] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579-2605, 2008.

[4] N. Oskolkov, "How to tune hyperparameters of tSNE," Towards Data Science, 18 July 2019. [Online]. Available: https://towardsdatascience.com/how-to-tune-hyperparameters-of-tsne-7c0596a18868.

[4

TSNE Plot

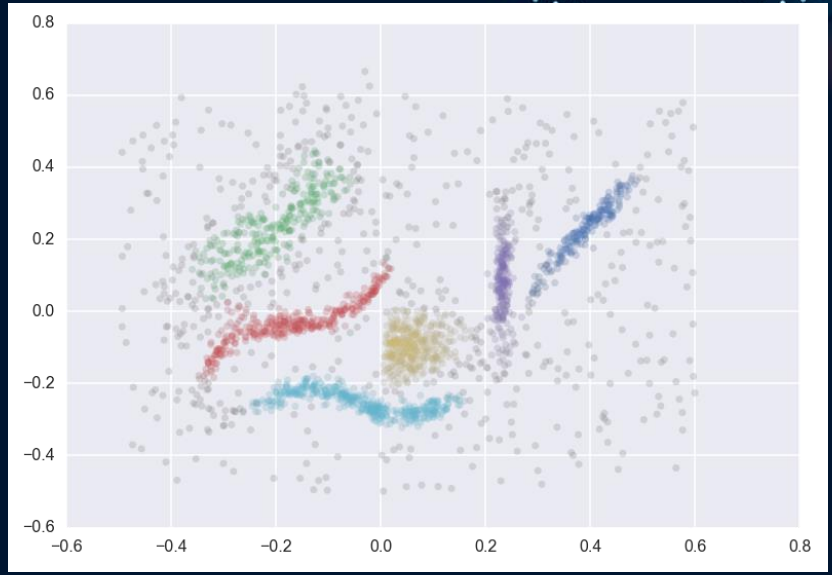


Unclustered, scaled, projected data

Hierarchical Density Based Spatial Clustering of Applications w/ Noise (HDBSCAN)

1. Generates sets of nested, related clusters
2. We don't know how many clusters we have
3. We don't know how big the clusters are
4. Indifferent to cluster shapes
5. Detects outliers
6. Easy and optimized parameter selection

[5]



7

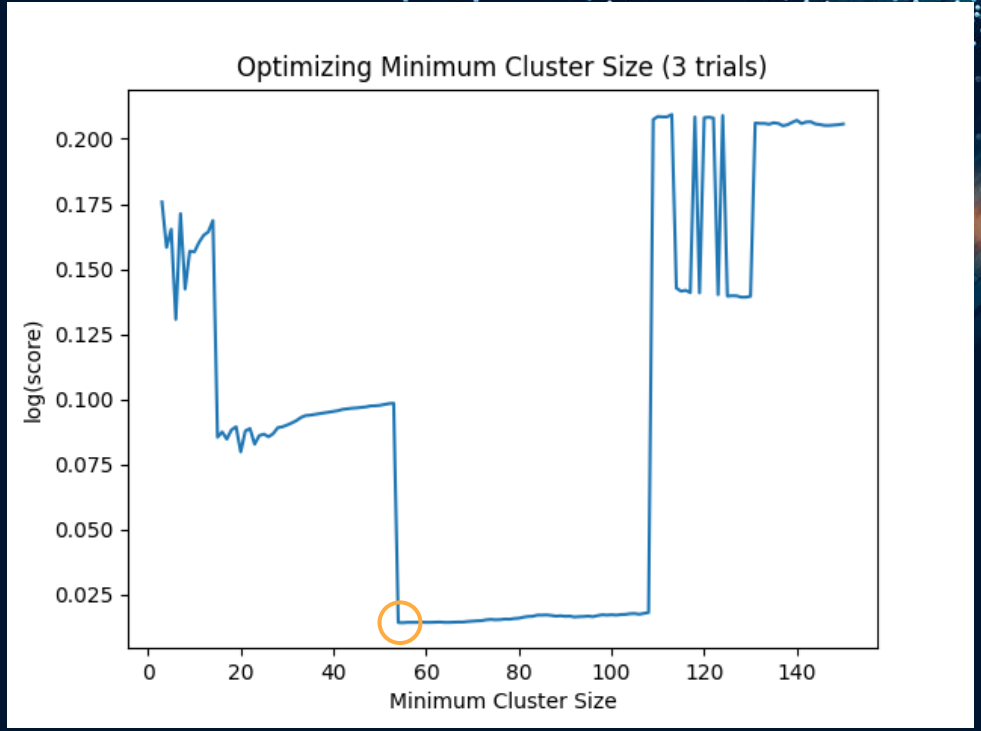
[5] L. Yang et al., "Semi-Supervised Log-Based Anomaly Detection via Probabilistic Label Estimation," 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 1448-1460, doi: 10.1109/ICSE43902.2021.00130.

[7] https://hdbscan.readthedocs.io/en/latest/advanced_hdbscan.html

Optimizing Min_Cluster_Size

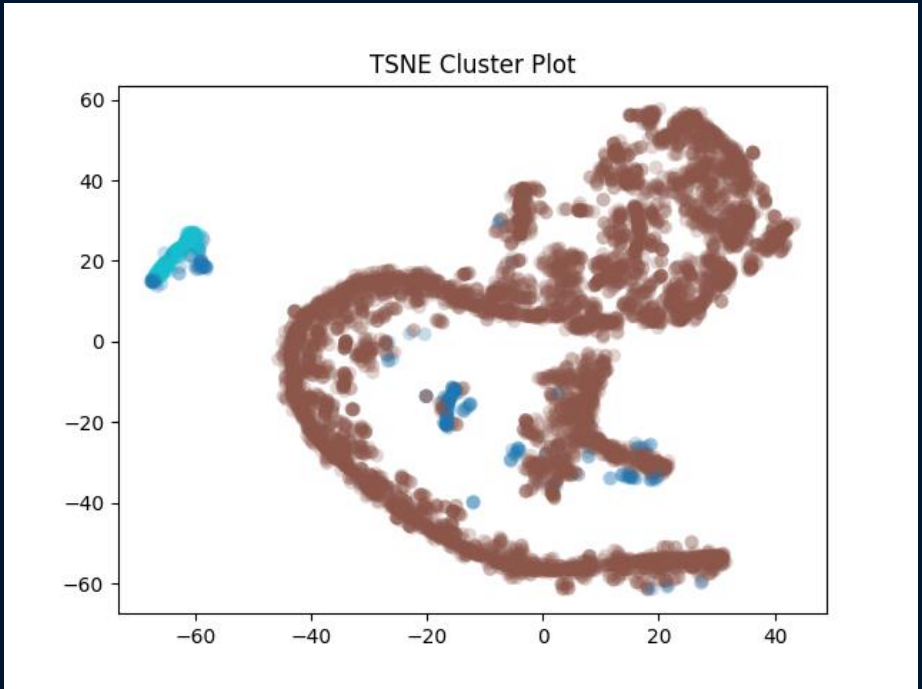
Want to maximize confidence that data points belong in their clusters

Find Min_Cluster_Size that minimizes number of points with <5% confidence [6]



[6] N. Oskolkov, "How to cluster in High Dimensions," Towards Data Science, 23 July 2019. [Online]. Available: https://towardsdatascience.com/how-to-cluster-in-high-dimensions-4ef693bacc6.

The Clusters

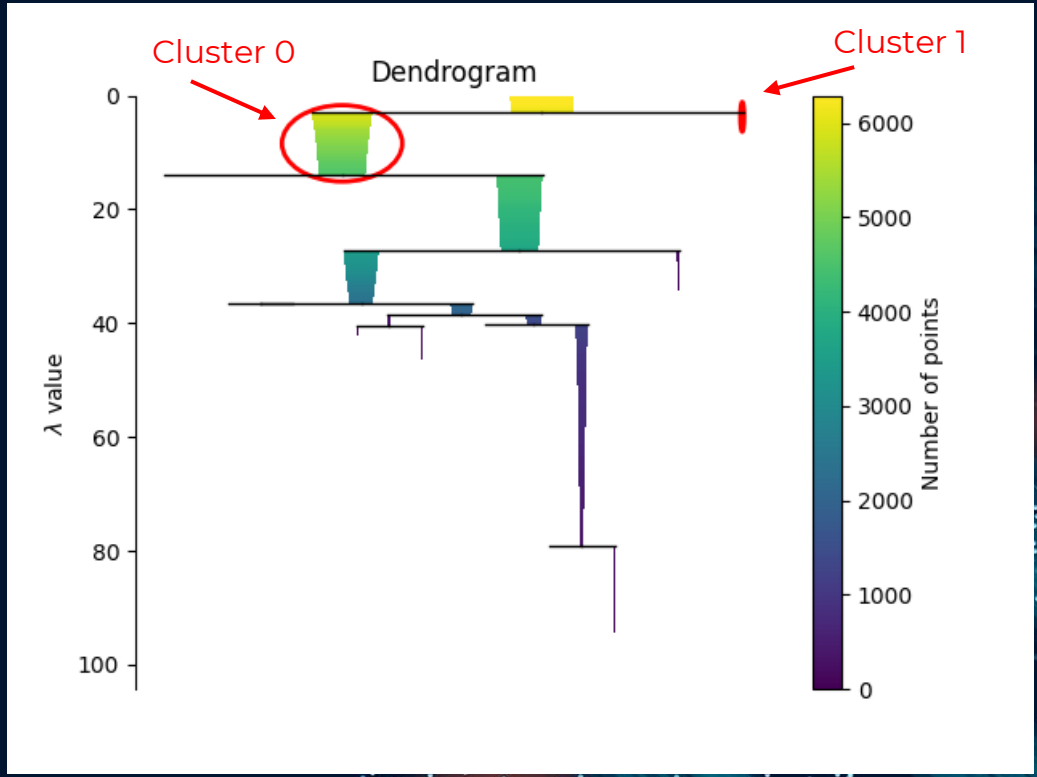


HDBSCAN TSNE plot with min_cluster_size = 55

- Cluster 0
- Cluster 1
- Noise

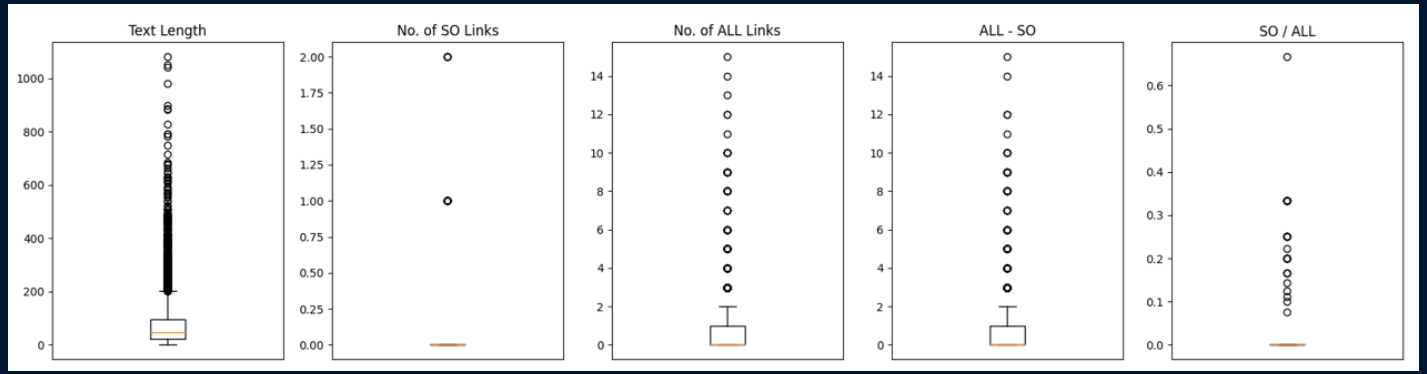
Dendrogram Analysis

HDBSCAN uses cluster hierarchy to maximize the clusters' lifespans without introducing too many artifacts [7].



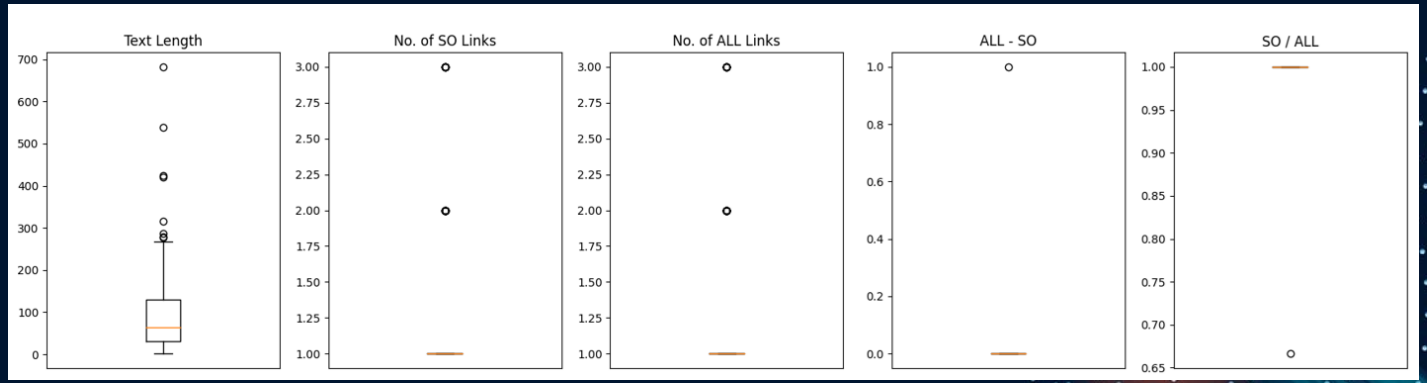
[7] L. McInnes, J. Healy, S. Astels, hdbscan: Hierarchical density based clustering In: Journal of Open Source Software, The Open Journal, volume 2, number 11. 2017

Length Comparison



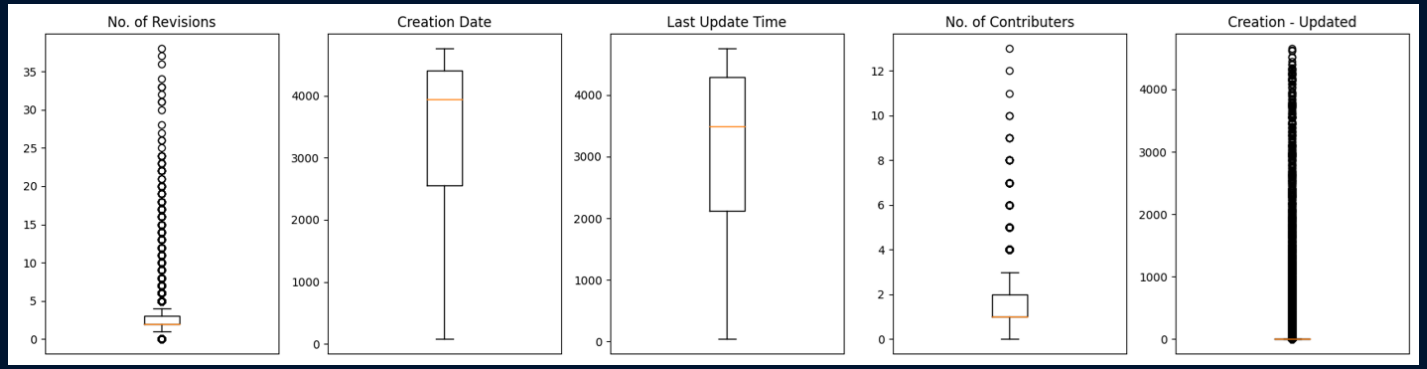
Cluster 0

Longer but fewer *STO* links



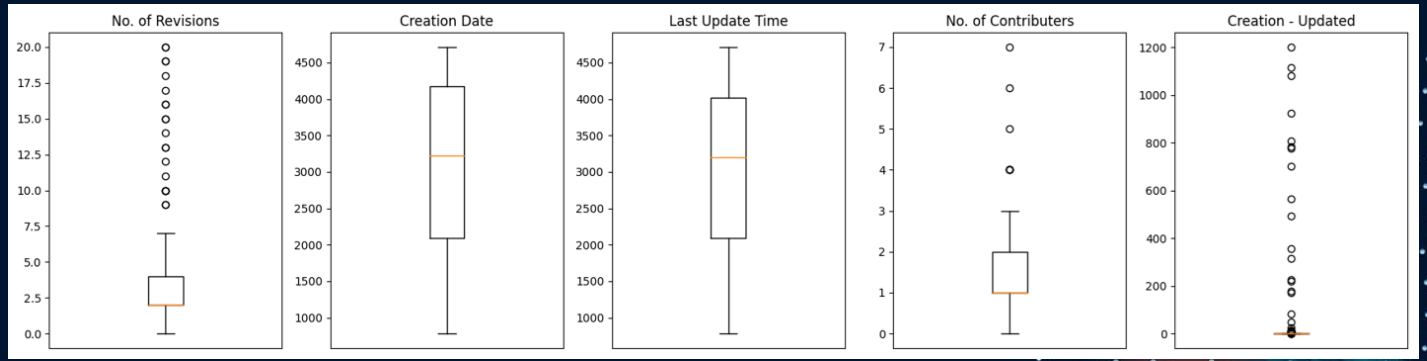
Cluster 1

Activity Comparison



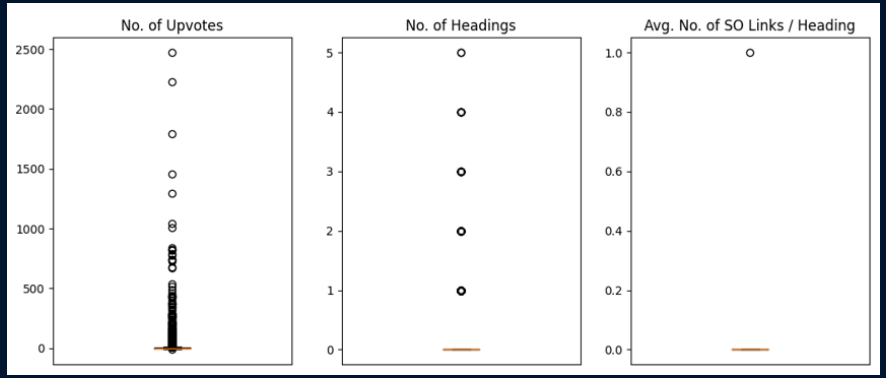
Cluster 0

More contributions but older



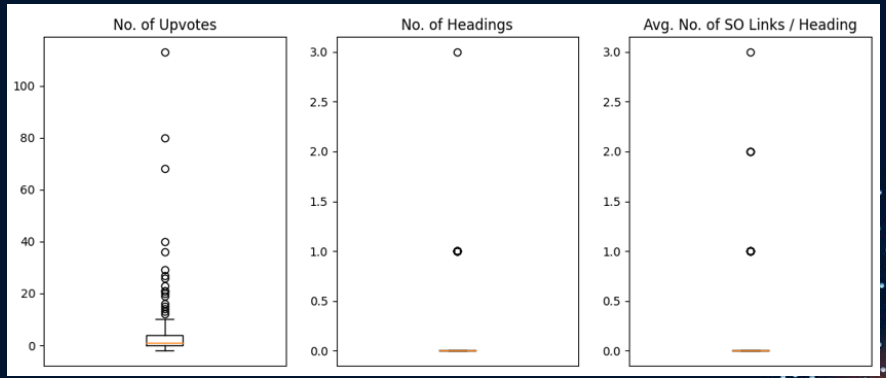
Cluster 1

Popularity and Structure Comparison



Cluster 0

Lower median upvotes and *STO* links per heading



Cluster 1

Stratified Sampling: Cluster 0

Pretty long, but is not a collaborative “structured post” that serves to organize *Stack Overflow*

6

Heres a complete function for adding and removing parameters based on this question and this github gist: https://gist.github.com/excalq/2961415

```
var updateQueryStringParam = function (key, value) {

    var baseUrl = [location.protocol, '//', location.host, location.pathname].join('
        urlQueryString = document.location.search,
        newParam = key + '=' + value,
        params = '?' + newParam;

    // If the "search" string exists, then build params from it
    if (urlQueryString) {

        updateRegex = new RegExp('([\?&])' + key + '[^&]*');
        removeRegex = new RegExp('([\?&])' + key + '=[^&;]+[&;]?');

        if( typeof value == 'undefined' || value == null || value == '' ) { // Remove

            params = urlQueryString.replace(removeRegex, "$1");
            params = params.replace( /[&;]$/, "" );

        } else if (urlQueryString.match(updateRegex) !== null) { // If param exists a

            params = urlQueryString.replace(updateRegex, "$1" + newParam);

        } else { // Otherwise, add it to end of query string

            params = urlQueryString + '&' + newParam;

        }

    }
    window.history.replaceState({}, "", baseUrl + params);
};
```

You can add parameters like this:

```
updateQueryStringParam( 'myparam', 'true' );
```

And remove it like this:

```
updateQueryStringParam( 'myparam', null );
```

In this thread many said that the regex is probably not the best/stable solution ... so im not 100% sure if this thing has some flaws but as far as i tested it it works pretty fine.

Share Edit Follow

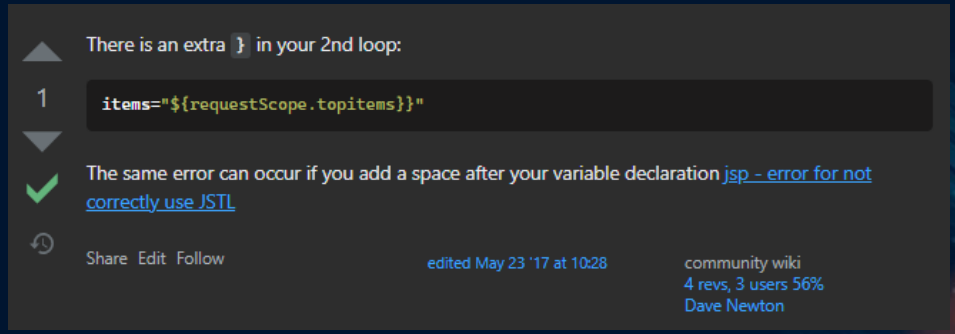
answered Nov 17 '16 at 6:59

community wiki
GDY

[8] https://stackoverflow.com/questions/40648561

Stratified Sampling: Cluster 1

Too short to be considered “structured,” but is a collaborative effort and facilitates finding other information on *Stack Overflow*


There is an extra } in your 2nd loop:

1

```
items="${requestScope.topitems}}"
```

The same error can occur if you add a space after your variable declaration jsp - error for not correctly use JSTL

Share Edit Follow

edited May 23 '17 at 10:28

community wiki
4 revs, 3 users 56%
Dave Newton

[9] https://stackoverflow.com/questions/24247862

Stratified Sampling: Noise

10

649

+400

One word answer: **asynchronicity**.

Forewords

This topic has been iterated at least a couple of thousands of times, here, in Stack Overflow. Hence, first off I'd like to point out some extremely useful resources:

- @Felix Kling's answer to "How do I return the response from an asynchronous call?". See his excellent answer explaining synchronous and asynchronous flows, as well as the "Restructure code" section.
  @Benjamin Gruenbaum has also put a lot of effort explaining asynchronicity in the same thread.
- @Matt Esch's answer to "Get data from fs.readFile" also explains asynchronicity extremely well in a simple manner.

The answer to the question at hand

Let's trace the common behavior first. In all examples, the `outerScopeVar` is modified inside of a *function*. That function is clearly not executed immediately, it is being assigned or passed as an argument. That is what we call a ***callback***.

Now the question is, when is that callback called?

It depends on the case. Let's try to trace some common behavior again:

- `img.onload` may be called *sometime in the future*, when (and if) the image has successfully loaded.
- `setTimeout` may be called *sometime in the future*, after the delay has expired and the timeout hasn't been canceled by `clearTimeout`. Note: even when using `0` as delay, all browsers have a minimum timeout delay cap (specified to be 4ms in the HTML5 spec).
- jQuery `$.post`'s callback may be called *sometime in the future*, when (and if) the Ajax request has been completed successfully.
- Node.js's `fs.readFile` may be called *sometime in the future*, when the file has been read successfully or thrown an error.

In all cases, we have a callback which may run *sometime in the future*. This "sometime in the future" is what we refer to as **asynchronous flow**.

Asynchronous execution is pushed out of the synchronous flow. That is, the asynchronous code will **never** execute while the synchronous code stack is executing. This is the meaning of JavaScript

```
        //    call the callback passing the result as argument
        callback('Nya');
    }, Math.random() * 2000);
}
```

Run code snippet Expand snippet

Most often in real use cases, the DOM API and most libraries already provide the callback functionality (the `helloCatAsync` implementation in this demonstrative example). You only need to pass the callback function and understand that it will execute out of the synchronous flow, and restructure your code to accommodate for that.

You will also notice that due to the asynchronous nature, it is impossible to `return` a value from an asynchronous flow back to the synchronous flow where the callback was defined, as the asynchronous callbacks are executed long after the synchronous code has already finished executing.

Instead of `return`ing a value from an asynchronous callback, you will have to make use of the callback pattern, or... Promises.

Promises

Although there are ways to keep the callback hell at bay with vanilla JS, promises are growing in popularity and are currently being standardized in ES6 (see Promise - MDN).

Promises (a.k.a. Futures) provide a more linear, and thus pleasant, reading of the asynchronous code, but explaining their entire functionality is out of the scope of this question. Instead, I'll leave these excellent resources for the interested:

- JavaScript Promises - HTML5 Rocks
- You're Missing the Point of Promises - domenic.me

More reading material about JavaScript asynchronicity

- The Art of Node - Callbacks explains asynchronous code and callbacks very well with vanilla JS examples and Node.js code as well.

A structured post! However, noise is inconsistent.

[10] https://stackoverflow.com/questions/23667087

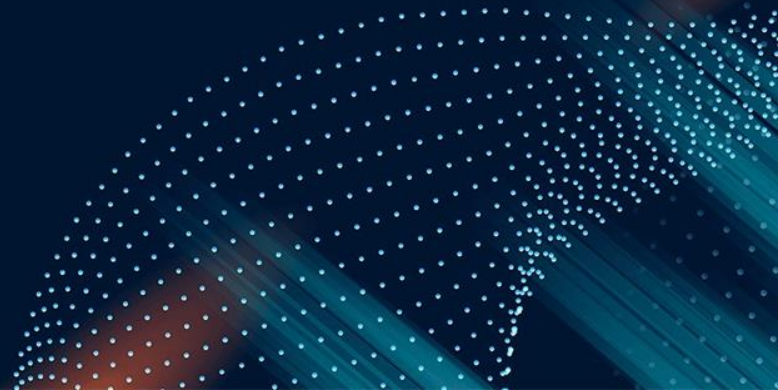
Inference

Structured posts are very rare:

- Time consuming to make
- Difficult to find (ironic!)

How can we better reveal structured posts?

- Cluster by property rather than all properties at the same time
- Introduce more features (e.g., no. of comments)
- Cluster latent variables



Future Work

01

Explore how structured posts are organized and managed

02

Use BERT to detect type-III code clones between structured posts and documentation [7]

03

Use NLP to develop an automated tool that generates structured posts

[7] Saini, Vaibhav, et al. "Oreo: Detection of clones in the twilight zone." Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (FSE), 2018.

Research Impact

Collaboration

Encourages innovation and idea-sharing among developers

Education

Increases accessibility to information for new and experienced developers

Efficiency

Saves developers' time to increase innovation

References

[1] https://stackexchange.com/sites?view=list#questions

[2] https://stackoverflow.com/questions/114543/how-to-horizontally-center-an-element

[3] https://stackoverflow.com/tags/html/info

[4] https://stackoverflow.com/a/22944075/10757178

[5] https://data.stackexchange.com/stackoverflow/query/new

[6] https://stats.stackexchange.com/a/283941

[7] https://hdbscan.readthedocs.io/en/latest/advanced_hdbscan.html

[8] https://stackoverflow.com/questions/40648561

[9] https://stackoverflow.com/questions/24247862

[10] https://stackoverflow.com/questions/23667087

References

[1] A. Diriye, M. L. Wilson, A. Blandford and A. Tombros, "Revisiting Exploratory Search from the HCI Perspective," London, 2010.

[2] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft computing,* vol. 97, 2020.

[3] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research,* vol. 9, pp. 2579-2605, 2008.

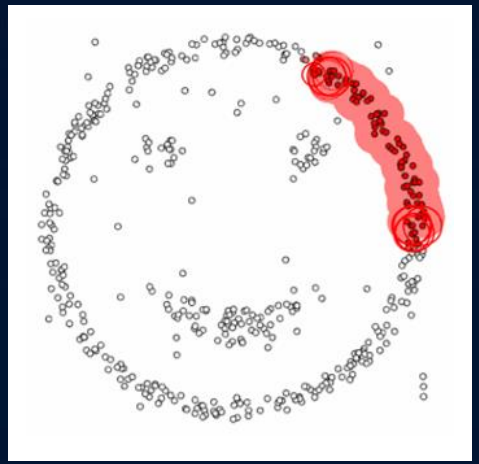
[4] N. Oskolkov, "How to tune hyperparameters of tSNE," Towards Data Science, 18 July 2019. [Online]. Available: https://towardsdatascience.com/how-to-tune-hyperparameters-of-tsne-7c0596a18868.

[5] L. Yang et al., "Semi-Supervised Log-Based Anomaly Detection via Probabilistic Label Estimation," 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 1448-1460, doi: 10.1109/ICSE43902.2021.00130.

[6] N. Oskolkov, "How to cluster in High Dimensions," Towards Data Science, 23 July 2019. [Online]. Available: https://towardsdatascience.com/how-to-cluster-in-high-dimensions-4ef693bacc6.

[7] Saini, Vaibhav, et al. "Oreo: Detection of clones in the twilight zone." Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (FSE). 2018.

Thanks for listening!



[6]