Automatic Organization of Semantically Related Tags Using Topic Modelling

Iman Saleh[⋆] and Neamat El-Tazi

Faculty of Computers and Information, Cairo University
{iman.saleh,n.eltazi}@fci-cu.edu.eg

Abstract. The use of social media platforms such as social networks, weblogs and community question answering websites increased largely in the last few years. This increase in usage contributed to the vast explosion in available online content. Some of these platforms use social tagging, tags manually inserted by content authors, as a way to facilitate content description and discovery. In this paper, our goal is to automatically group semantically related tags in order to organize the large amount of tags contributed by various users. Our approach is based on using a topic model to discover topics of documents, and then grouping top tags related to documents assigned to each topic. We perform a set of experiments using different number of topics to provide different levels of details for the generated tag groups. The dataset used in our experiments is extracted from Stack Overflow, a community question answering website used by programming professionals. Tag groups generated by our technique are presented and evaluated.

Keywords: Stack Overflow, Tags, Topic Modelling, LDA, Organize Tags

1 Introduction

The amount of data available online in social media platforms is increasing largely. This large amount of data contains hidden information that need to be extracted. A way of organizing and categorizing this massive amount of data is to use tags inserted by users. However, the number of tags can increase largely over time. It is necessary in such a case to organize tags themselves to help users discover hidden topics in the data. In our work we are interested in Stack Overflow community question answering platform and how to organize its tags.
Stack Overflow is the largest online community question answering website designed to help developers share their programming knowledge. It allows them to post programming related questions and receive answers to them. The website contains[1] around 13.4 million questions tagged with around 48,228 tags. Questions are manually tagged by users and revised by privileged users. Tags in Stack Overflow are used in research studies to predict tags of posts [18,17,25,22,16,10],

[⋆] Corresponding author.
[1] All numbers mentioned in the paper date back to March 2017.

map tags to Wikipedia concepts [11], suggest and group tag synonyms [3,4], analyze trends of topics [2] and mine challenges encountered by developers [20]. Tags are important because they help categorize posts and hence do proper analysis of Stack Overflow content. The large number of tags in Stack Overflow makes it important to organize them into meaningful groups to be able to browse and analyze Stack Overflow content easily.
In this paper we introduce an automatic method to organize Stack Overflow tags. Our method is based on using Latent Dirichlet Allocation (LDA) topic model [5] to group semantically related tags. We first categorize a sample of posts in Stack Overflow into a number of topics. Each topic is assigned a set of key phrases that serve as a title for the topic using Open Calais tool[2] to provide a meaningful title for each group of words produced by LDA. Then we extract top tags assigned to the posts of each topic and consider them a group of related tags. We do some filtering on groups of tags to remove tags that are common among most of the topics. In order to evaluate how accurate tag groups are, we introduce a method inspired from word intrusion task in [6]: tag intrusion. This method is used to measure how semantically coherent tags are and whether they correspond to natural grouping for humans. We perform a set of experiments using different number of topics and provide proper evaluation of our technique. The following is a summary of the contributions introduced by this paper:

- An automatic method to group Stack Overflow tags into semantically meaningful topics using LDA topic model (Section3).
- A formal evaluation technique for our method (Section 3.4).

2 Related Work

Social tags in social media platforms were utilized in different contexts. [23] consider Twitter hashtags as topics and use them to automatically generate an overall sentiment polarity for a given hashtag. [7] utilize Twitter hashtags as an indicator of events. They use hashtags to discover breaking events. Hashtags are considered topics in Twitter as explained in [13] and used to predict the popularity of a topic. Semantic relationship between Twitter hashtags was explored in [8]. They construct a graph of hashtags and entities drawn from tweets such that edges capture semantic relatedness. A predictive model that incorporates the effect of time on individual hashtag reuse and social hashtag reuse was introduced in [12]. Their goal is to introduce a hashtag recommendation algorithm that considers time dimension in its recommendations. Stack Overflow and Twitter are similar in using tags to define topics. However, Twitter tags are much more diverse and changes frequently overtime unlike Stack Overflow tags which are more stable over time.
Grouping Stack Overflow tags was previously studied in [3]. Their approach was mainly focused on finding tags that are synonyms to each other. They used some strategies to achieve their goal based on studying the set of tag synonyms that

[2] http://www.opencalais.com/

is provided by Stack Overflow. Their work focused only on android related tags. They further extended their work to group tag synonyms using graph clustering techniques [4]. In our work we are interested in similar tags in a broader sense and not only tags that are synonyms to each other. Working on tag synonyms only will result in a large number of tag groups that cannot be browsed easily. [9] used agglomorative clustering technique to group Stack Overflow tags. They only considered tags themselves and did not use Stack Overflow posts in their experiments. We introduce an alternative method that makes use of Stack Overflow posts and proper evaluation of our method. [21] also used k-medoids clustering in order to infer semantically related software tags. They measure the similarity of two tags based on the number of documents tagged by both of them and the similarity of textual content of documents tagged by them. However, the dataset they used is much smaller than our dataset and we believe their approach might not scale well to a large number of tags. [19] used Stack Overflow posts to build a software specific resource similar to WordNet. Their approach is based on the textual content of the posts and not their tags. A large taxonomy of subsumption relations between pairs of Stack Overflow tags was created using machine learning in [26]. They use several features such as lexical tags features and features extracted from tags wiki description. Their work focuses mainly on discovering relationships between tag pairs instead of creating tag groups. Finding trends of topics in Stack Overflow was explored in [2]. They also used LDA to find topics of Stack Overflow posts. Clusters of tags under each topic are manually created and referred to as technologies. Trends of topics and technologies are then extracted over time. In our work we want to automate the process of grouping tags under a certain topic. We also think that tag trends themselves carry important information in addition to topic or technology trends. Our analysis is based on tags themselves and how to compare between them. [20] also used LDA to find topics in Stack Overflow related to different Web API's. Their goal is to find common challenges encountered by developers. They investigate how topics related to different Web API's evolve over time. In our work we are interested in all topics in Stack Overflow rather than specific Web API's. [1] used LDA to mainly analyze problems faced by developers in Stack Overflow. They also provide an analysis of Stack Overflow usage in week days.

3 Grouping Stack Overflow Tags Using LDA

LDA is a generative probabilistic model that is used to model text corpora. It is based on (1) representing documents as a random mixture over k latent topics (2) modeling topics as distributions over words. In this model, topics are groups of semantically related words in a corpus of documents. For example, a group of words like "entity, model, database, query, mongodb" are semantically related and refer to concepts related to database and data management. Also, a document similar to the one shown in Figure 1 is clearly about git and version control systems. So it belongs to the topic represented by the following group of words: "git, branch, repository, svn, repo, github". We also notice that the document is labeled with the following tags: “git, alias, git-push”. Therefore these three tags can be grouped together since they are semantically related and refer to the same topic.

Title: Why does git push origin @ not work?
Post: We can push the head of a branch like below
$ git push origin HEAD
And, we can use @ for alias of HEAD.
$ git show @
Then why does the below command gives me an error?
$ git push origin @
fatal: remote part of refspec is not a valid name in @
Tags: git, alias, git-push

Fig. 1: An example of a Stack Overflow post

In our work we are interested in creating tag groups that are semantically related in order to organize Stack Overflow massive content. Working towards this end, we train LDA using a subset of Stack Overflow posts, extract tags assigned to each post, and then associate these tags with the most likely topic assigned to the post. Details of our method are explained in the following subsections.

3.1 Training LDA model

We use MALLET [15] implementation of LDA to model a Stack Overflow posts dump[3]. When we investigated posts we found that some of them are of low quality and create noise in our model. Therefore we filtered data to include only a subset of posts that are assigned a score greater than or equal to 5. The score of a post is assigned by Stack Overflow community users as the difference between the number of upvotes and downvotes.
We split the dataset into 2 folds to make sure our technique is valid for Stack Overflow data. The first fold, DS_1, consists of 463,983 posts while the second, DS_2, contains 360,610 posts. Data in each fold was selected randomly first then filtered based on score. Code snippets were removed and tags of each post were included in training data. Best answer for a post is also included if available. We used Stanford parser [14] to stem data and use only noun words found in a post. LDA has three parameters: k, alpha and beta. The first parameter is the number of topics. The second parameter, alpha, controls the mixture of topics in a document; whether a document is likely to contain a mixture of most topics rather than only specific few topics. The third parameter, beta, controls the mixture of words in a topic; whether a topic is likely to contain most words rather than only specific words. We train LDA using 10, 30, 50 and 80 topics and compare between the four values. Alpha value used in our work is 50/k and

[3] Dump downloaded in October 2016

beta value is 200/V, as recommended in [24], where V is the vocabulary size. Vocabulary size for DS_1 is 184,147, and 143,023 for DS_2. The model is trained using 2000 iterations since this is a reasonable number for the model to converge and find good topics.

3.2 Assigning Titles to Each Topic

Topics generated by LDA are mainly groups of words that are semantically related. To generate topics that are meaningful while browsing tag groups, we need to have a title for each group of words; or a set of phrases that describe the topic represented by those words. In order to generate a title for each topic, we use the top 20 words of each topic as an input to Open Calais API, a tool used for tagging text. For each topic, a set of tags and their relevance to input text are produced (as shown in Table 1). These tags can be used as topics titles. Tags common among all topics such as "computing", "software" and "software engineering" are excluded from the set of title tags.

Topic index	Topic Top 20 Words	Open Calais Tags
15	css, html, div, tag, width, image, browser, style, height, content, javascript, jquery, bootstrap, size, font, svg, background, selector, chrome, border	JavaScript libraries 66%, HTML 66%, Ajax 66%, Bootstrap 66%, Open formats 66%, JQuery 66%, Cascading Style Sheets 66%, HTML element 66%, Web design 66%
27	table, sql, database, column, query, row, mysql, server, record, index, statement, oracle, postgresql, sqlite, procedure, key, insert, transaction, order, clause	Data management 100%, SQL keywords 66%, Database management systems 66%, Database trigger 66%, MySQL 66%, Databases 66%, PostgreSQL 66%, Join 66%, Insert 66%

Table 1: Open Calais tags of $topic_{15}$ and $topic_{27}$ and their relevance to top words

3.3 Creating Tag Groups Using LDA

LDA model assigns a mixture of topics to each document. For each topic, we collect all posts that most likely belong to it. Then we extract tags assigned to these posts. We refer to the tag group of $topic_i$ produced using an LDA model trained with k topics as TTG_i^k. For each TTG_i^k, we count how many times a tag appears associated with that $topic_i$. Each TTG_i^k is then assigned the top 5% tags according to frequency of occurrence in topic posts. An example of top tags associated with $topic_{15}$ (Table 1) is shown in Table 2.

However, grouping tags is not a straightforward task. We notice that some tags appear in all topics or most of them. For instance, javascript tag can appear

Tags in TTG^{30}_{15}	Frequency
css	10,462
html	7,292
javascript	3,645
jquery	3,079
css3	2,087
twitter-bootstrap	1,454
html5	1,271

Table 2: TTG^{30}_{15} top tags and their frequency of occurrence

Tags in UTG^{30}_{15}
css
css3
twitter-bootstrap
html5
svg
google-chrome
css-selectors

Table 3: $topic_{15}$ top tags that appear in 50% or less of TTG^{30}_{i}

in topics related to database, version control systems, android as well as regular expressions. Therefore, a tag that appears in most or all of the topics cannot be assigned to a specific group of tags since it is shared among topics. Therefore, we create a group of tags called "shared tags". We refer to shared tags group created using an LDA model trained with k topics as STG_k. An STG_k group contains tags that appear in 50% or more of all TTG^k_i. For instance, when k=10, if a tag_i is found in 5 or more TTG^k_i groups, it is a shared tag. Table 4 shows a sample of shared tags produced by LDA model with k = 10, 30, 50 and 80 and appeared in more than 50% of all TTG^k_i. We also create another group of tags for each topic that contains the remaining tags. These are tags that are either unique to a topic or appear in less than 50% of all TTG^k_i. We refer to the group of unique tags of a $topic_i$ created using an LDA model trained with k topics as UTG^k_i. Table 3 shows tags in UTG^{30}_{15}.

	100% of topics	90% or more and less than 100%
STG_{10}	android, asp.net, c++, c#, cocoa, file, ios, iphone, java, javascript, jquery, json, .net, objective-c, php, python, ruby, ruby-on-rails, string, visual-studio, performance, visual-studio-2010	arrays, angularjs, api, delphi, eclipse, c, css, html, json, node.js, osx, qt, r, unit-testing, vb.net, windows, winforms, wpf, xml
STG_{30}	c#, java, javascript, python, .net	android, c++, ios, php, python, ruby
STG_{50}	c#, java	javascript, .net, php, python
STG_{80}	N/A	c#, java, javascript, python

Table 4: Example of shared tags extracted using LDA model trained with k topics

3.4 Evaluating Tag Groups

An appropriate quantitative measure is needed to evaluate extracted tag groups. It should be able to measure the coherence of tags in a tag group and whether they are actually semantically related. In order to achieve this goal, we use tag intrusion, a measure inspired from word intrusion task explained in [6]. Word intrusion task measures semantic coherence of top words in an LDA topic. This measure was shown to capture aspects of LDA models that are not captured by other measures of model quality based on held out likelihood. So we believe that a similar measure can be useful in assessing semantic coherence of tag groups. In word intrusion task, top words of a $topic_i$ produced by an LDA model are selected. A word $w_{intruder}$ is then selected randomly from the top words of another $topic_j$ and included together with the set of top words of $topic_i$. $w_{intruder}$ must not be present in the top words of $topic_i$. Annotators are then asked to find the word that does not belong to the group. In our case we call the evaluation task tag intrusion. We aim at finding how cohesive tag groups are, i.e. degree of relatedness between tags in a tag group UTG_i^k. We create the task as follows: given an LDA model trained with k topics, for each UTG_i^k, the top five tags are selected. Then another random UTG_j^k is selected. Next, we randomly choose a tag $tag_{intruder}$ from the top 3 tags in UTG_j^k such that $tag_{intruder} \notin UTG_i^k$. We count how many times the intruder tag was selected correctly by 3 different annotators and use the percent as a measure of tag groups cohesion. Given that tag_{answer} is the intruder tag selected by annotator s and S is the total number of annotators, Equation 1 shows how we calculate tag intrusion. Table 5 shows the results of tag intrusion using tag groups extracted from LDA trained with k topics equal to 10, 30, 50 and 80.

$$TagIntrusion_k = \frac{(\sum_s(\sum_{i=1}^{k}(tag_{intruder} = tag_{answer})/k))}{S} \tag{1}$$

	k=10	k=30	k=50	k=80
DS_1	0.73	0.80	0.81	0.83
DS_2	0.77	0.79	0.76	0.80

Table 5: Tag intrusion results for datasets DS_1 and DS_2

4 Discussion

The results of tag intrusion (Table 5) show that our method achieves at least 73% score for tag intrusion. We used two different sets in order to validate our results and notice that the difference between scores for both is less than or equal to 5% which indicates our algorithm generalizes well on Stack Overflow data. Also, shared tags and tags under specific topic groups are almost the same for both datasets as shown in Table 6. Tag groups for "Regular Expressions and Character Encoding" and "Version control software" topics extracted from both datasets are almost the same. The table also shows that shared tags extracted from both datasets are very similar.

The number of topics does not have a significant effect on tag intrusion score. We think that the choice of the number of topics depends on whether

	DS_1	DS_2
Regular Expressions and character Encoding	regex, string, unicode, utf-8, parsing, perl, encoding, bash, character-encoding, split	regex, string, unicode, utf-8, perl, encoding, bash, character-encoding, split, replace
Version Control Software	git, svn, github, version-control, mercurial, jenkins, branch, merge, tortoisesvn, tfs	git, svn, github, version-control, mercurial, branch, merge, tortoisesvn, docker, jenkins
Tags in ST_{30} appearing in all topics	javascript, java, c#, python	Java, c#, .net, python

Table 6: Tag groups sorted based on frequency of appearance with topic, LDA k=30

we are interested in high level or low level topics. The lower the number of LDA topics, the more diverse tags will be grouped under a single topic. This is clear in Table 4 were the number of shared tags is inversely proportional to the number of LDA topics. For example, "regex" tag is considered as a shared tag for k=10. When k is increased to 30, the tag appears as the most frequent tag under a specific topic: "character encoding and regular expressions". Also, as the number of topics increases, some topics are close to each other and its hard to distinguish between them. For instance, when we set k=80, two topics produced very similar top tags as shown in Table 7. We think that organizing

Tag Group	Topic Titles	Top Tags
UTG_{64}^{80}	Smartphones, IOS SDK, Integrated development environments, Xcode, Tablet computers, IPad, Swift, GPS navigation devices, Cocoa	xcode, swift, cocoa, osx, xcode6, cocoa-touch, core-data, ipad, ios8, xcode4
UTG_{36}^{80}	IOS, IOS SDK, TvOS, Xcode, Swift, IPhone	uitableview, swift, cocoa-touch, xcode, ipad, ios8, ios7, cocoa, uiview

Table 7: Similar tag groups produced using LDA trained with k=80

tags into groups is particularly useful to explore tag trends and how they change over time. A user can first choose a topic of interest and then browse tags trends under a topic. This can also facilitate comparing trends of two tags over time. For example, frequent tags include "javascript, java, c#". Then "ios" comes next with lower frequency. Non-frequent tags, compared to the previous tags, include "objective-c", "scala". We find this result some how consistent with the frequency of questions in which these tags appear as shown in Table 8[4].

[4] Numbers obtained from http://Stack Overflow.com/tags

	javascript	java	c#	ios	objective-c	scala
Freq.	1.3m	1.2m	1.0m	481.3k	273.9k	63.1k

Table 8: Frequencies of top shared tags in Stack Overflow

5 Conclusion and Future Work

In this paper we introduced an automatic method to organize Stack Overflow tags. We use LDA to group tags that are semantically similar. We evaluated our method using tag intrusion score; a measure inspired from word intrusion that is used to evaluate LDA. We plan to extend our work by finding better methods to group Stack Overflow tags based on semantic similarity. A comparison between our method and other methods used to organize social tags needs to be performed as well. Finally, we plan to launch a demo to visualize our results.

Acknowledgements

This work is funded by Information Technology Academia Collaboration program organized by Information Technology Industry Development Agency in Cairo, Egypt. We thank Tarek Nabhan, Ahmed Hany and Noura Hassan for their guidance.

References

1. Allamanis, M., Sutton, C.: Why, when, and what: Analyzing stack overflow questions by topic, type, and code. In: Proceedings of the 10th Working Conference on Mining Software Repositories. pp. 53–56. IEEE Press (2013)
2. Barua, A., Thomas, S.W., Hassan, A.E.: What are developers talking about? an analysis of topics and trends in stack overflow. Empirical Software Engineering 19(3), 619–654 (2014)
3. Beyer, S., Pinzger, M.: Synonym suggestion for tags on stack overflow. In: Proceedings of the 2015 IEEE 23rd International Conference on Program Comprehension. pp. 94–103. ICPC '15, IEEE Press (2015)
4. Beyer, S., Pinzger, M.: Grouping android tag synonyms on stack overflow. In: Proceedings of the 13th International Conference on Mining Software Repositories. pp. 430–440. MSR '16, ACM (2016)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research 3(Jan), 993–1022 (2003)
6. Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J.L., Blei, D.M.: Reading tea leaves: How humans interpret topic models. In: Advances in neural information processing systems. pp. 288–296. Curran Associates, Inc. (2009)
7. Cui, A., Zhang, M., Liu, Y., Ma, S., Zhang, K.: Discover breaking events with popular hashtags in twitter. In: CIKM (2012)
8. Ferragina, P., Piccinno, F., Santoro, R.: On analyzing hashtags in twitter. In: ICWSM (2015)
9. Gajduk, A., Madjarov, G., Gjorgjevikj, D.: Intelligent tag grouping by using an aglomerative clustering algorithm. In: The 10th Conference for Informatics and Information Technology CIIT (2013)

10. Gruetze, T., Krestel, R., Naumann, F.: Topic shifts in stackoverflow: Ask it like socrates. In: International Conference on Applications of Natural Language to Information Systems. pp. 213–221. Springer (2016)
11. Joorabchi, A., English, M., Mahdi, A.E.: Automatic mapping of user tags to wikipedia concepts: The case of a q&a website–stackoverflow. Journal of Information Science (2015)
12. Kowald, D., Pujari, S.C., Lex, E.: Temporal effects on hashtag reuse in twitter: A cognitive-inspired hashtag recommendation approach. In: WWW (2017)
13. Ma, Z., Sun, A., Cong, G.: On predicting the popularity of newly emerging hashtags in twitter. JASIST 64, 1399–1410 (2013)
14. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: ACL (System Demonstrations). pp. 55–60 (2014)
15. McCallum, A.K.: Mallet: A machine learning for language toolkit (2002), `http://mallet.cs.umass.edu`
16. Romero, J.J.F., Guerrero, M.G., Calder, F.: Multi-class multi-tag classifier system for stackoverflow questions. In: 2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC). pp. 1–6. IEEE (2015)
17. Saha, A.K., Saha, R.K., Schneider, K.A.: A discriminative model approach for suggesting tags automatically for stack overflow questions. In: Proceedings of the 10th Working Conference on Mining Software Repositories. pp. 73–76. IEEE Press (2013)
18. Stanley, C., Byrne, M.D.: Predicting tags for stackoverflow posts. In: 12th International Conference on Cognitive Modelling. pp. 414–419 (2013)
19. Tian, Y., Lo, D., Lawall, J.: Automated construction of a software-specific word similarity database. In: 2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE). pp. 44–53 (2014)
20. Venkatesh, P.K., Wang, S., Zhang, F., Zou, Y., Hassan, A.E.: What do client developers concern when using web apis? an empirical study on developer forums and stack overflow. In: IEEE International Conference on Web Services (ICWS). pp. 131–138. IEEE (2016)
21. Wang, S., Lo, D., Jiang, L.: Inferring semantically related software terms and their taxonomy by leveraging collaborative tagging. In: International Conference on Software Maintenance. pp. 604–607. IEEE Computer Society (2012)
22. Wang, S., Lo, D., Vasilescu, B., Serebrenik, A.: Entagrec: An enhanced tag recommendation system for software information sites. In: Proceedings of the 2014 IEEE International Conference on Software Maintenance and Evolution. pp. 291–300. IEEE Computer Society (2014)
23. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: CIKM (2011)
24. Wood, J.: Source-lda: Enhancing probabilistic topic models using prior knowledge sources. CoRR abs/1606.00577 (2016)
25. Xia, X., Lo, D., Wang, X., Zhou, B.: Tag recommendation in software information sites. In: Proceedings of the 10th Working Conference on Mining Software Repositories. pp. 287–296. IEEE Press (2013)
26. Zhu, J., Shen, B., Cai, X., Wang, H.: Building a large-scale software programming taxonomy from stackoverflow. In: The 27th International Conference on Software Engineering and Knowledge Engineering. pp. 391–396. KSI Research Inc. and Knowledge Systems Institute Graduate School (2015)