**Transfer Learning on Stack Exchange Tags**

Made by: Ankit Sharma

Email id: [ankitactive@gmail.com](mailto:ankitactive@gmail.com)

**Index**

1. The Problem 3
2. My approach 4
3. Analysis 5
   1. Libraries used 5
   2. Reading data and processing 5
   3. Declaring custom built functions 5
   4. Clean, Process, and Print A Question

Sampled at “Random” 5

1. Matching tags to title and content 7
   1. Loop through empty character matrix 7
   2. Joining data back 7
2. Plotting the results 8
   1. Density overload: Percentage of tags matching title and content 8
   2. Histogram and density of tags matching titles 10
   3. Histogram and density of tags matching content 11

The Problem

We aim to predict tags from models on unrelated models. Stack Exchange wants a model to be able to take the title and content of a question, and recommend the best tags to use for that particular question. Ideally, this would be done with no prior knowledge of the most popular tags on the sight (specifically, on that particular category of stack exchange site).

My approach

The approach I will take in this report will be to first identify how many words in the tags match words in both the title and content for each question. I will make sure all words are converted to lower-case. I will not (yet) stem any of the words. By “stem”, I mean split a word such as “argued” / “arguing” / “argues” down into its root of “argu”. No, these stemmed words do not always represent a real word themselves, but they will help better identify patterns of commonly altered root words.

The tags in the Biology category don’t match the question Titles or Content well. The Cooking category matched their tags to titles and content the best

Analysis

**Libraries used:**

* *data.table*: The data table R package provides an enhanced version of data frame that allows you to do blazing fast data manipulations.
* *ggplot2:* The ggplot2 package, created by Hadley Wickham, offers a powerful graphics language for creating elegant and complex plots.
* *tm*: offers framework for text mining in R

**Reading Data and Processing**

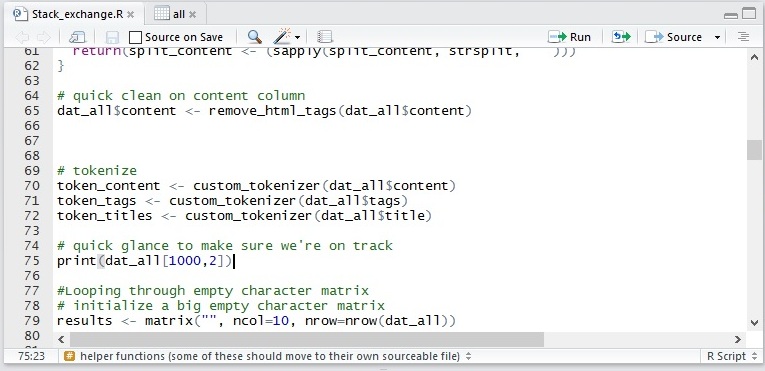
In this section, I will read in all of the data, attach a category label for each data set, and combine all of these data frames into a single data frame. I also noticed that IDs are recycled within different categories, so I elected to make a concatenated id field composed of the **id** and **category** fields (concatenated with a pip character between them).

Declaring My Custom Functions

I’m using two custom functions for this script so far:

* **remove\_html\_tags:** removes all html tags from a string
* **custom\_tokenizer:** takes in a string and makes it lowercase, removes punctuation, removes new line and return characters, removes any extra whitespace beyond a single space, then splits the string by spaces

Clean, Process, and Print A Question Sampled at “Random”



OUTPUT:

print(dat\_all[1000,2])

[1] "How do inward rectifier potassium channels work in the heart?"

Matching tags to title and content

**Quick and Dirty: Loop Through Empty Character Matrix**

I have found that working with a large character matrix is pretty computationally efficient. The key is to not use a solution based on c() or rbind() for capturing loop results. That requires a lot of RAM gymnastics which will slow the machine down. Having a matrix with predefined rows and columns allows for quick insertion by indices.

Within the actual loop, I am calculating the number of tags that can **exactly** match a word in the title, and then also in the content of the question. I also calculate a few other data points such as the percent of tags that are found in both the title and content.

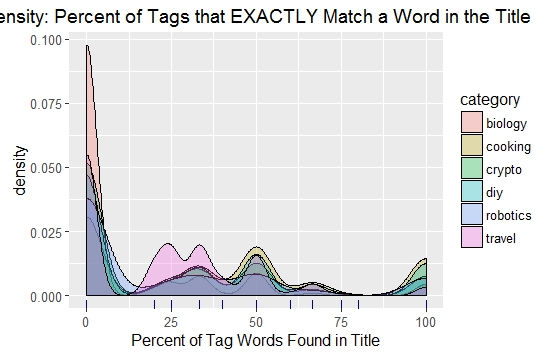
**Joining data back together:**

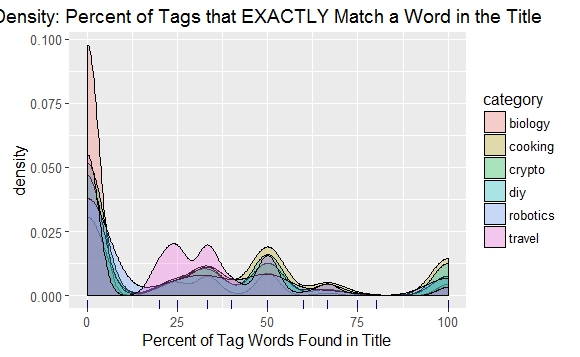
Now we have the results saved in a character matrix, and our initial data in a data.frame. I’ll turn them both into data.tables for a quick merge on the custom *id\_cat* field and then we will be ready for plotting.

Plotting the results

**Density Overlaid: Percent of tags matching words in title and content**

I wanted to first turn the transparency up and see if I could see some differences between the different categories of sites these questions came from (in terms of how well people’s tags match their questions).



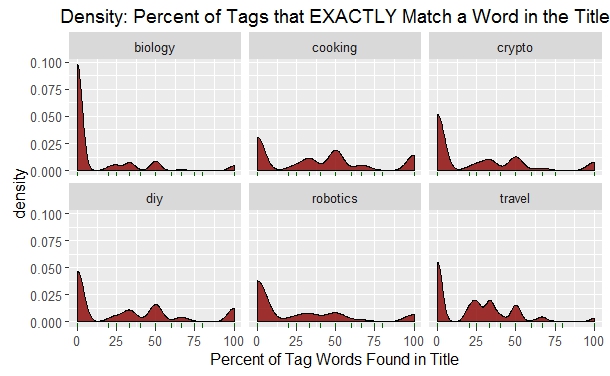


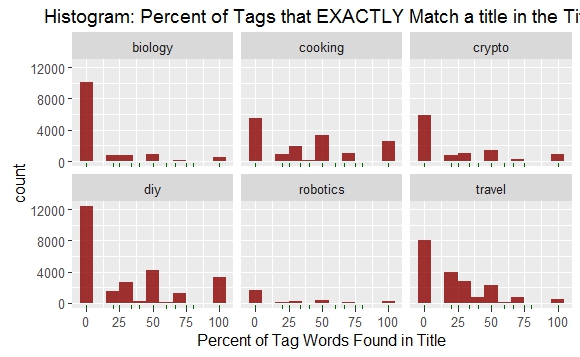
The only discernible patterns I’m seeing are that Biology questions are the most common types of questions to have tags that don’t match the title or the content very well. The “Travel” category doesn’t seem to do much better.

“Cooking” appears to be leading the pack. This makes intuitive sense, as the words used in a cooking question would be much simpler than some of the more technical categories such as “Biology” or “Robotics”.

In terms of plots, I think we can make this more readable:

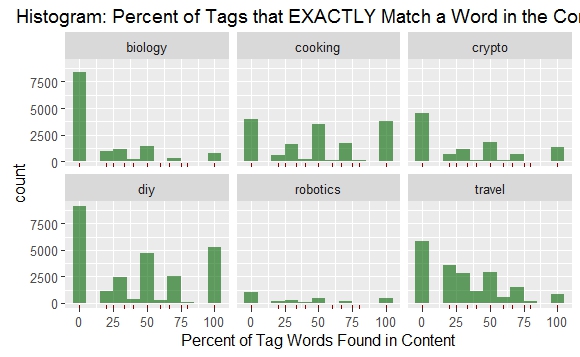
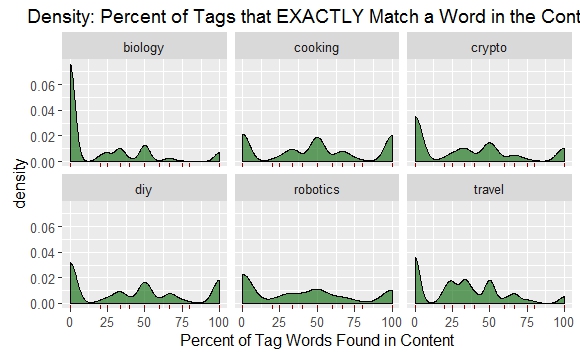
**Histogram and Density of Tags Matching Titles**

****

****

I would say that “Cooking” is in the lead, followed by “DIY”. “Biology” is clearly the worst in terms of tags appearing in the title.

**Histogram and Density of Tags Matching Content**

Trying the same plots for the content and see what we can see

Conclusions here look very similar to the same plots created for the titles.