

Analyzing and increasing the performance of a suite of 570+ help articles

Sarah Fierman

Role: Senior Content Strategist

Team: Technical Content

Company: Mailchimp

● Outline

1. Context
2. The problem
3. My approach
4. Recommendations
5. Impact

Context

- The company and team



What's Mailchimp?

Mailchimp is an email and marketing automations platform for growing businesses.

What's the Technical Content team?

The Technical Content team owns mailchimp.com/help, including a suite of 570+ help articles.

● My role and partners

I led this multi-phase project from discovery through delivery, while partnering with multiple roles across the support organization.

My role ★

Senior Content Strategist for the
Technical Content team
(DRI)

The team 🤝

8 writers
Content engineering
Support engineering

The problem

● The problem

The Technical Content team at Mailchimp used several templates to write help articles.

It wasn't clear how often writers used each template type and we were publishing articles without a clear way to show what content was most helpful.

Pain points for the customer 🥲

Customers can't find the answer to a question or problem quickly causing them to have to reach out to support via email or chat.

Impacts to the business 📉

Low self-serve rates and weighty and costly maintenance loads for managing the 570+ help articles.

My approach

● Approach

Goals

Understand **what content is most helpful** to our customers, so we can make sure we're giving our customers the most helpful experience.

Trim unnecessary content from the knowledge base to **reduce time and maintenance costs.**

Approach

1. Make the data accessible
2. Audit and tag the knowledge base
3. Analyze the article types

● Approach: Make the data accessible

As technical writer (before I moved into the content strategy role),
I noticed there was data about our help content that wasn't being used.

Information about how support agents and customers were interacting with the help articles documented, but wasn't directly available to writers.

● Approach: Make the data accessible

I reached out to the support engineers and started a conversation.

Together, we implemented a process where, via JIRA tickets, writers could request information about:

- How many views an article had received in a specific time frame
- How often it had been shared by support agents in a chat
- And more...

● Approach: Make the data accessible

This process worked well for awhile, but there were still problems.

Because the support engineering team needed time to fulfill each request, **the data was often slightly out-of-date or less relevant** to the technical writer's question.

When I moved into the content strategist role, my focus became partnering with a content engineer to pull our internal support data and customer interaction information from Google Analytics into a dashboard.

This meant writers could get information they needed in real time.

● Approach: Audit and tag

At the same time that we were building the dashboard, I started auditing and tagging each article in the knowledge base according to the template type.

I stored this information in the metadata in our CMS (Contentful).

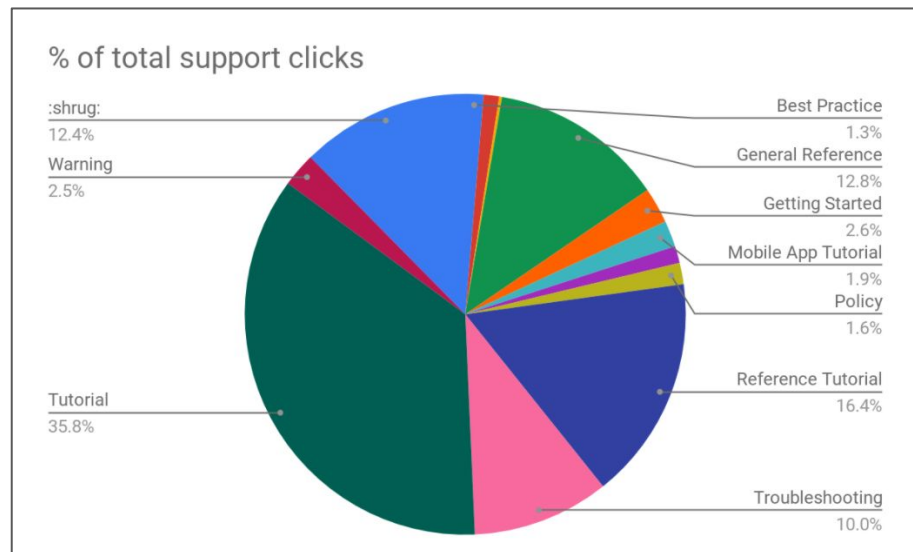
This is an example from <https://styleguide.mailchimp.com/> of the types of templates the team used to outline technical content articles.

Article Template	User Type	Goal
Pathfinder	prospective, new, intermediate	Orientation. Bundle topics and provide links to relevant tutorials or general reference.
General Reference	prospective, new, intermediate	Introduction. Provide a high-level explanation of what the feature is, how it works, and its benefit to the user. Include links to relevant tutorials.
Troubleshooting	new, intermediate, advanced	Support. Outline expected behavior and include potential causes of unexpected behavior. Group by cause or topic.
Tutorial	new, intermediate	Guidance. Briefly describe a task. Provide a roadmap and prerequisites, and clear step-by-step instructions.

● Approach: Analyze the types

With the powers of our dashboard and tagged content combined, I was able to group articles together by type and analyze their performance.

In the audit, I found that there were some articles that did not fit any of the article template types. I grouped these into the “shrug” category.



:shrug: was an inside joke with the team based on the emoji name in Slack. 🙄 It meant we didn't know where these articles fit.

● Approach: Analyze the types

When I compared “shrug” articles with the other categories, I found they had a much lower self-serve rate.

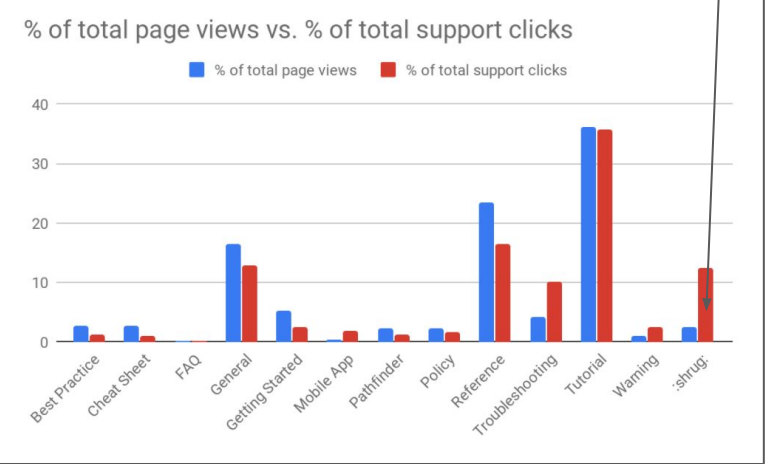
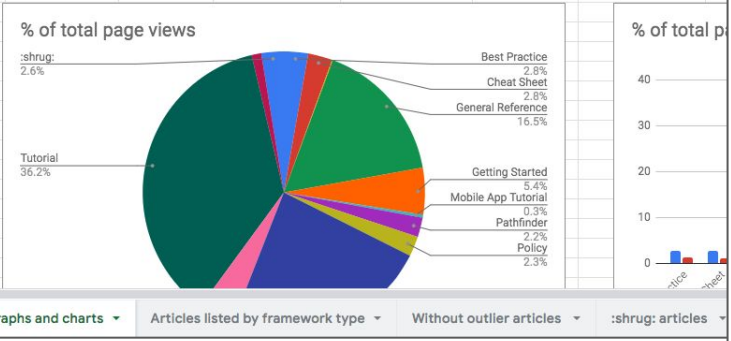
Customers were much more likely to contact technical support from “shrug” articles.

I also looked at average time on the page and user satisfaction rates from the feedback form on mailchimp.com/help/ pages. People tended to spend less time on the page and were less satisfied over all in articles that fit the “shrug” category.

Approach: Analyze the types

		Average # page views (for 1 article, based on framework type)	% of total page views	Support clicks by framework	Average # support clicks (for 1 article, based on framework type)	% of total support clicks	Self-serve rate average (%)	Framework type as percentage of all articles
15	Best Practice	56,538	3769.2	2.773595652	52	3.466666667	1.336073998	2.782931354
8	Cheat Sheet	56,763	7095.375	2.784633521	44	5.5	1.130524152	1.484230056
3	FAQ	1,963	654.3333333	0.09629927243	8	2.666666667	0.2055498458	0.5565862709
66	General Reference	336,640	5100.606061	16.51461389	498	7.545454545	12.7954779	12.24489796
10	Getting Started	109,214	10921.4	5.357732419	102	10.2	2.620760534	1.85528757
18	Mobile App Tutorial	6,952	386.2222222	0.3410456149	75	4.166666667	1.927029805	3.339517625
18	Pathfinder	45,318	2517.666667	2.223173932	48	2.666666667	1.232399075	3.339517625
15	Policy	46,909	3127.266667	2.301223928	61	4.066666667	1.567317575	2.782931354
87	Reference Tutorial	480,203	5519.574713	23.55741188	640	7.356321839	16.44398767	16.14100186
32	Troubleshooting	84,091	2627.84375	4.125268527	391	12.21875	10.04624872	5.936920223
229	Tutorial	738,923	3226.737991	36.24948919	1,394	6.087336245	35.81706064	42.48608534
17	Warning	22,511	1324.176471	1.104326501	97	5.705882353	2.492291881	3.153988668
21	:shrug:	52,412	2495.809524	2.571185668	482	22.95238095	12.38437821	3.896103896
539				100				
		2,038,437			3,892			

:shrug: articles are a small percentage of the articles, but are where the majority of support clicks were coming from



Recommendations

● Recommendations

I presented my findings to the Technical Content team and recommended restructuring each of the articles in the *:shrug:* category into alignment with a template.

● Recommendations

Recommendation:

Let's clean up the :shrug: articles.

People are more likely to contact support from a **:shrug:** article.

Some of this could be the subject matter of those articles, a.k.a. troubleshooting in nature.

But it's safe to say, **patterns and structure help our users.**

We should bring these articles into alignment with our frameworks.

● Recommendations

Recommendation:

Let's add the Reference Tutorial framework to our standard frameworks.

Here's how we're currently describing this framework: "Combines older reference and tutorial articles as part of content audits".

We have **88** Reference Tutorial articles, that make up **16.3%** of the Knowledge Base.

So, we're using this framework a lot more than we thought! And not just as a way to combine older articles.

Impact

● Impact: Higher self-serve rates

After a “shrug” article was restructured to fit a template, I circled back on a regular cadence to assess if the changes had any impact to the self-serve rate or user satisfaction rate.

I found that both metrics increased to the level of the articles in other template types when articles were in alignment with the typical patterns and structure.

Articles that were brought into alignment reached a 99.2% self-service rate. Industry standard at the time was closer to 75%.

● Impact: Low-lift maintenance strategy

Because of this, I was able to verify the value of the consistent patterns in help content and make the content more helpful for customers.

I was also able to identify the articles in each template category that had a low self-serve rate and low views.

We cut 50+ articles from the knowledge base without impacting customer satisfaction scores. This became a successful, repeatable maintenance strategy that helped us reduce costs and effort.

Thank you for your time!