Discuss one thing you notice that is different between the two emails that might relate to the identification of spam.

One thing I noticed that is different between that might help identify spam is that the spam email is a lot longer with a lot more jargon. The ham seems to be written well with more correct grammar and is more concise. The spam email also seems to be more of an advertisement. The spam email is also in HTML/CSS format.

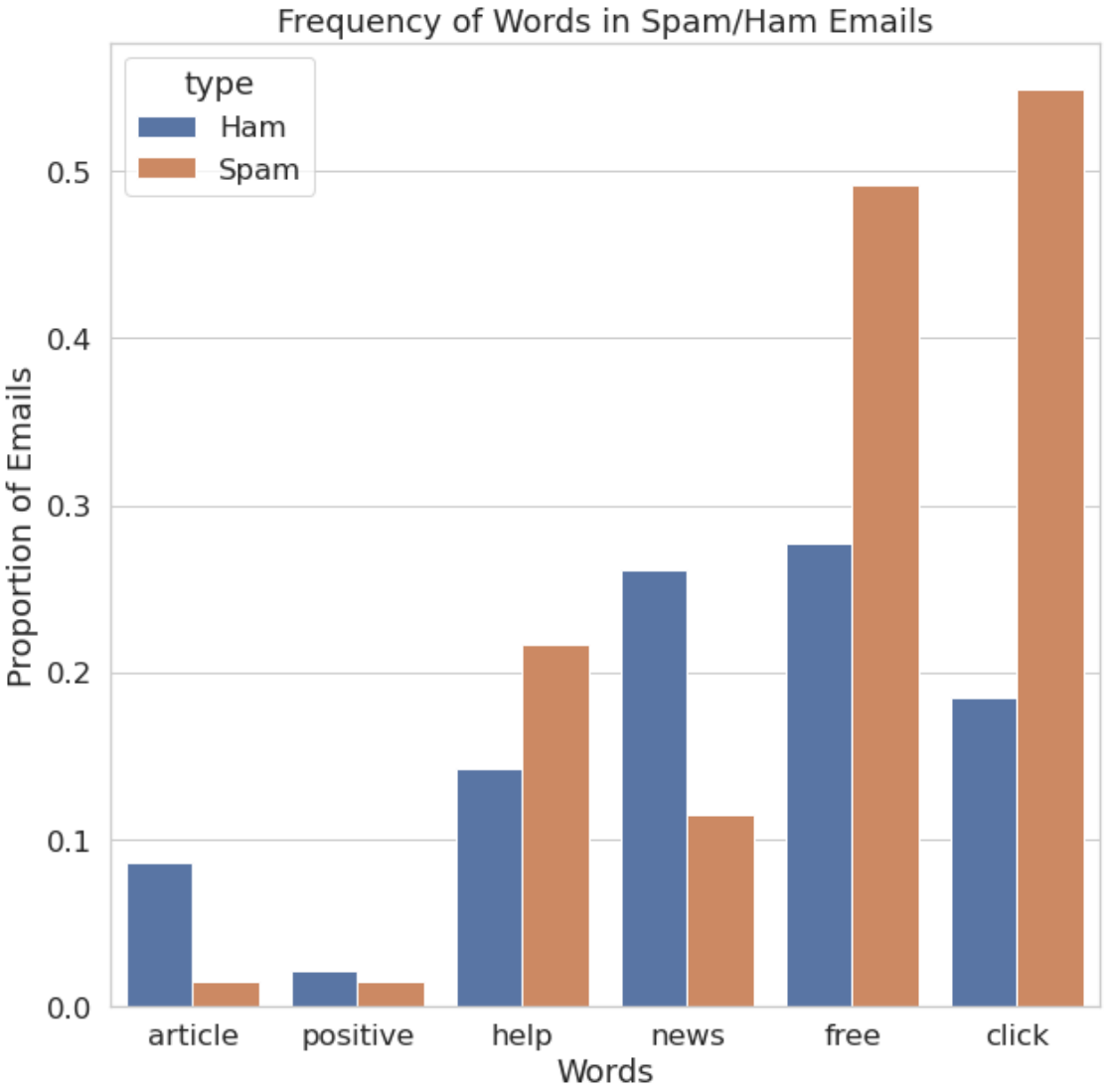
0.0.1 Question 3

Create a bar chart like the one above comparing the proportion of spam and ham emails containing certain words. Choose a set of words that are different from the ones above, but also have different proportions for the two classes. Make sure to only consider emails from `train`.

In [264]:
```
train = train.reset_index(drop=True) # We must do this in order to preserve the ordering of e
my_words = words_in_texts(['article', 'positive', 'help', 'news', 'free', 'click' ], train['e
word_1 = [x[0] for x in my_words]
word_2 = [x[1] for x in my_words]
word_3 = [x[2] for x in my_words]
word_4 = [x[3] for x in my_words]
word_5 = [x[4] for x in my_words]
word_6 = [x[5] for x in my_words]

my_df = pd.DataFrame(data= {'article': word_1, 'positive': word_2, 'help': word_3, 'news': wo
my_df['type'] = my_df['type'].replace({0:'Ham',1:'Spam'})
my_df_melted = my_df.melt('type')

plt.figure(figsize=(10,10))
sns.barplot(data = my_df_melted, x = 'variable', y = 'value', hue = 'type', ci = False)
plt.xlabel('Words')
plt.ylabel('Proportion of Emails')
plt.title('Frequency of Words in Spam/Ham Emails');
```


Frequency of Words in Spam/Ham Emails
type
Ham
Spam
Proportion of Emails
0.0
0.1
0.2
0.3
0.4
0.5
article
positive
help
news
free
click
Words

0.0.2 Question 6c

Comment on the results from 6a and 6b. For **each** of FP, FN, accuracy, and recall, briefly explain why we see the result that we do.

For the zero_predictor_fp, we see 0 occurrences of this because there is no positive value in our predictor since our predictor is all zeros. For the zero_predictor_fn, we see 1918 occurrences of this because there are 1918 times when the results in Y_train are 1s and but out predictor gets 0s. Our zero_predictor_acc is 0.744 because the (TP + TN)/n is 0.744. The zero predictor is 74.4% accurate. Our recall is 0 because it is TP/(TP+FN) and our number of TP is 0 since our predictor is all zeros.

0.0.3 Question 6e

Are there more false positives or false negatives when using the logistic regression classifier from Question 5?

There are 122 occurrences of false postives and 1699 occurrences of false negatives when using the logistic regression classifier from Question 5. There are a lot more false negatives when using the logistic regression classifier from Question 5.

0.0.4 Question 6f

1. Our logistic regression classifier got 75.76% prediction accuracy (number of correct predictions / total). How does this compare with predicting 0 for every email?
2. Given the word features we gave you above, name one reason this classifier is performing poorly. Hint: Think about how prevalent these words are in the email set.
3. Which of these two classifiers would you prefer for a spam filter and why? Describe your reasoning and relate it to at least one of the evaluation metrics you have computed so far.

1) Compared the the zero_predictor which predicted 0 for every email which had an accuracy of 74.47%, the logistic regression classifier got 75.76% prediction accuracy which is a little bit higher than the zero predictor.
2) One reason why the given classifier is performing so badly is because ['drug', 'bank', 'prescription', 'memo', 'private'] is not a good enough representation of the many spam/ham emails. These words are very spcific to personal information and can be ham emails. We should add a lot more words and also words that are closely correlated with both spam and ham emails.
3) I would prefer the given logistic regression classifier for spam filter because it is more accurate than the zero predictor and also there is a value for precision and recall. The zero predictor has zero precision and recall because there are no TP. Having the precision and recall gives us another perspective on the accuracy and how precise/good at classifying postives is our chosen classifier. It also tells us how much our chosen classifier is penalizing false positives and false negatives.

To double-check your work, the cell below will rerun all of the autograder tests.

```
In [275]: grader.check_all()
```

```
Out[275]: q2 results: All test cases passed!

          q4 results: All test cases passed!

          q5 results: All test cases passed!

          q6a results: All test cases passed!

          q6b results: All test cases passed!

          q6d results: All test cases passed!
```

0.1 Submission

Make sure you have run all cells in your notebook in order before running the cell below, so that all images/graphs appear in the output. The cell below will generate a zip file for you to submit. **Please save**

before exporting!

In [276]:
```
# Save your notebook first, then run this cell to export your submission.
grader.export()
```

```
<IPython.core.display.HTML object>
```