0.0.1 Question 1c

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

The ham email seems more personalized since it includes a name as well, whereas the spam email seems to contain a template consisting of texts like <\html>,<\head> etc. It also seems to use very general wording.

0.0.2 Question 3a

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 [13]:
```
train #= train.reset_index(drop=True)
#[train['email'][1]]
```

Out[13]:

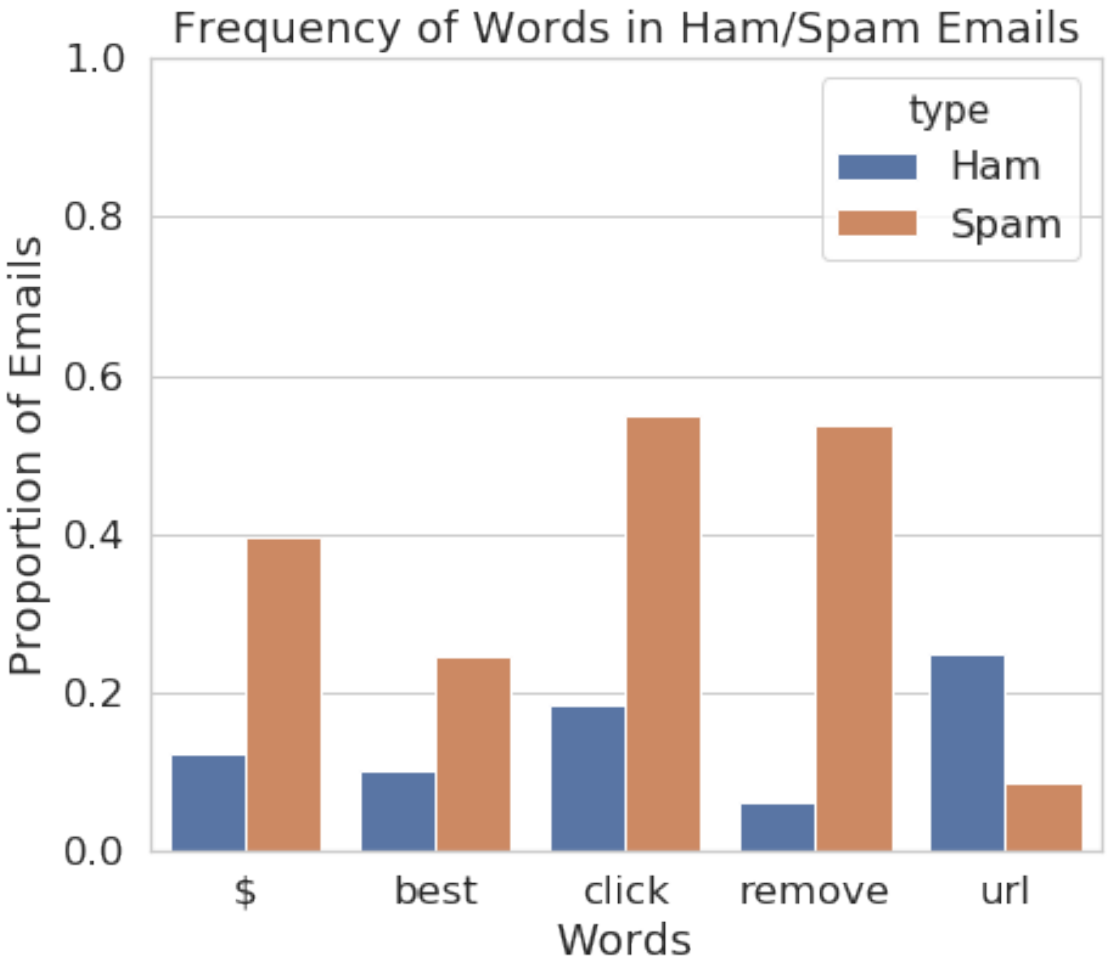
	id	subject	email	spam
7657	7657	Subject: Patch to enable/disable log\n	while i was playing with the past issues, it a…	0
6911	6911	Subject: When an engineer flaps his wings\n	url: http://diveintomark.org/archives/2002/10/…	0
6074	6074	Subject: Re: [Razor-users] razor plugins for m…	no, please post a link!\n \n fox\n ----- origi…	0
4376	4376	Subject: NYTimes.com Article: Stop Those Press…	this article from nytimes.com \n has been sent…	0
5766	5766	Subject: What's facing FBI's new CIO? (Tech Up…	<html>\n <head>\n <title>tech update today</ti…	0
…	…	…	…	…
5734	5734	Subject: [Spambayes] understanding high false …	>>>>> "tp" == tim peters <tim.one@comcast.net>…	0
5191	5191	Subject: Reach millions on the internet!!\n	\n dear consumers, increase your business sale…	1
5390	5390	Subject: Facts about sex.\n	\n forwarded-by: flower\n \n did you know that…	0
860	860	Subject: Re: Zoot apt/openssh & new DVD playin…	on tue, oct 08, 2002 at 04:36:13pm +0200, matt…	0
7270	7270	Subject: Re: Internet radio - example from a c…	chris haun wrote:\n > \n > we would need someo…	0

[7513 rows x 4 columns]

In [14]:
```
train=train.reset_index(drop=True) # We must do this in order to preserve the ordering of emai

ward = ['remove', 'best', 'url', '$', 'click']
intexts = words_in_texts(ward, train['email'])

dataf = pd.DataFrame(
    intexts, columns=ward
)
dataf['type'] = train['spam']


hams = dataf.query('type==0')
```

```
spams = dataf.query('type==1')

dataf = dataf.melt('type')

#hamprop = [np.mean(hams[y]) for y in ward]
#spamprop = [np.mean(spams[y]) for y in ward]


dataf = dataf.groupby(['type', 'variable']).mean().reset_index()
dataf = dataf.replace({0: 'Ham', 1: 'Spam'})

plt.figure(figsize=[7, 6])
plt.title('Frequency of Words in Ham/Spam Emails')
sns.barplot(data = dataf, x = 'variable', y = 'value', hue='type')
plt.ylim(0, 1)

plt.xlabel('Words')
plt.ylabel('Proportion of Emails')
```

```
Out[14]: Text(0, 0.5, 'Proportion of Emails')
```


Frequency of Words in Ham/Spam Emails
Proportion of Emails
1.0
0.8
0.6
0.4
0.2
0.0
type
Ham
Spam
$
best
click
remove
url
Words

0.0.3 Question 3b

Create a *class conditional density plot* like the one above (using `sns.distplot`), comparing the distribution of the length of spam emails to the distribution of the length of ham emails in the training set. Set the x-axis limit from 0 to 50000.

In [15]:
```
train
```

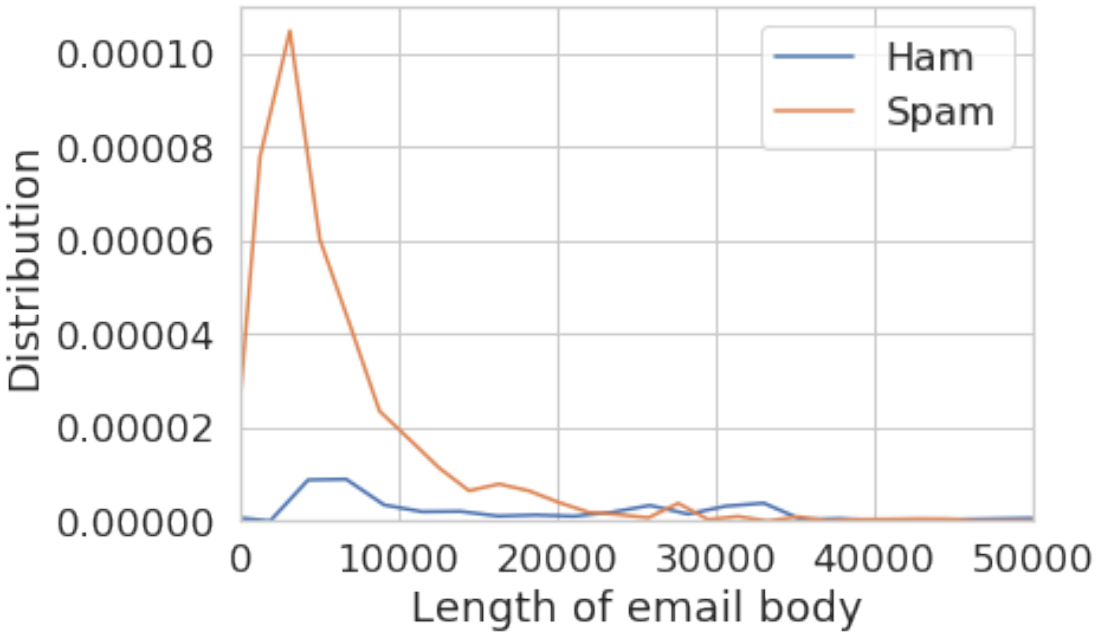
Out[15]:

	id	subject	email	spam
0	7657	Subject: Patch to enable/disable log\n	while i was playing with the past issues, it a…	0
1	6911	Subject: When an engineer flaps his wings\n	url: http://diveintomark.org/archives/2002/10/…	0
2	6074	Subject: Re: [Razor-users] razor plugins for m…	no, please post a link!\n \n fox\n ----- origi…	0
3	4376	Subject: NYTimes.com Article: Stop Those Press…	this article from nytimes.com \n has been sent…	0
4	5766	Subject: What's facing FBI's new CIO? (Tech Up…	<html>\n <head>\n <title>tech update today</ti…	0
…	…	…	…	…
7508	5734	Subject: [Spambayes] understanding high false …	>>>>> "tp" == tim peters <tim.one@comcast.net>…	0
7509	5191	Subject: Reach millions on the internet!!\n	\n dear consumers, increase your business sale…	1
7510	5390	Subject: Facts about sex.\n	\n forwarded-by: flower\n \n did you know that…	0
7511	860	Subject: Re: Zoot apt/openssh & new DVD playin…	on tue, oct 08, 2002 at 04:36:13pm +0200, matt…	0
7512	7270	Subject: Re: Internet radio - example from a c…	chris haun wrote:\n > \n > we would need someo…	0

[7513 rows x 4 columns]

In [16]:
```
classcon = train.replace({0: 'ham', 1: 'spam'})#.groupby('spam').apply(len())['email']
classcon['charcount'] = [len(i) for i in classcon['email']]
#classcon = classcon.groupby(['spam', 'charcount']).reset_index()
#classcon
sns.distplot(classcon.loc[classcon['spam'] != 'spam']['charcount'], hist=False, label='Ham')
sns.distplot(classcon.loc[classcon['spam'] == 'spam']['charcount'], hist=False, label='Spam')

plt.xlim(0, 50000)
plt.xlabel('Length of email body')
plt.ylabel('Distribution')
```

Out[16]:
```
Text(0, 0.5, 'Distribution')
```


Ham
Spam
Distribution
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
0
10000
20000
30000
40000
50000
Length of email body

0.0.4 Question 6c

Provide brief explanations of the results from 6a and 6b. Why do we observe each of these values (FP, FN, accuracy, recall)?

Since our predictor always predicts 0 regardless, our false positive count is 0 because we aren't even bothering to predict anything as positive in the first place. As a result, our recall, which relies on our true positive count in the numerator, is also 0. Our false negative in this case would simply be the true number of Spam in our training set, which can be expressed as sum(Y_train) since 1's are spam. The accuracy would in this case simply be our true negative count divided by the total length of our data, since we have 0 true positives.

0.0.5 Question 6e

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

The logistic regression classifier provides more false positives than the zero predictor, but also provides less false negatives.

0.0.6 Question 6f

1. Our logistic regression classifier got 75.6% 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. This is not too far from our zero predictor accuracy; it differs by around 1%.

2. The given words are very specific and seem to adhere to a niche medical/professional related emails only. As a result, it creates a classifier that may not be strong in predicting whether general emails of any kind are spam or ham.

3. I would honestly prefer the zero predictor over the logistic classifer. In the case of predicting spam emails, I would want to minimize the number of false positives, because this will falsely filter out emails that could be important, making my life harder. Although the zero_predictor allows all emails to pass through as ham, it has 0 false positives whereas the logistic regression classifier has 122, so I would still be receiving all my emails and I could then manually sort it. The accuracy between the two are not too far, suggesting that most emails are ham in the first place anyway.

0.0.7 Question 7: Feature/Model Selection Process

In this following cell, describe the process of improving your model. You should use at least 2-3 sentences each to address the follow questions:

1. How did you find better features for your model?
2. What did you try that worked / didn't work?
3. What was surprising in your search for good features?

1. I took the route of honing in on particular, niche words that constitute a spam email. Although it may seem counterintuitive, I began by finding commonly used words in everyday language and conversation, and then finding the highest repeating words in spam emails that are not part of this set.
2. I tried to compare the frequencies and types of punctuation marks as well as small details in the subject line, but these methods proved to be less fruitful than I expected.
3. I was really surprised in finding that once I threw punctuation out of the door when analyzing the email texts, it proved to help a great deal in terms of organization and effectiveness of my classification.

Generate your visualization in the cell below and provide your description in a comment.

In [27]:
```
train
import re
```

In [28]:
```
wordy = ['a', 'an', 'the', 'how', 'i', 'but', 'be', 'can', 'cannot', "can't", 'we', 'you', 'us
wordy += ['have', 'had', 'which', 'who', 'why', 'with', 'ours', 'yours', 'our', 'he', 'her', '
wordy += ['if', 'is', 'it', 'from', 'had', 'their', 'them', 'where']
#wordy
```

In [29]:
```
classcon['no_pun'] = classcon['email'].str.replace(r'([^\w\s])', ' ')
classcon['word_by_word'] = classcon['no_pun'].str.split()
classcon['word_cont'] = classcon['word_by_word'].apply(lambda z: [i for i in z if i not in wor
classcon['word_num'] = classcon['word_by_word'].str.len()
#classcon
spam_stuff = classcon.query("spam=='spam'").word_cont.explode().value_counts()
spords = spam_stuff[:400].index.to_list()
#spords
```

In [30]:
```
X_train_cc = words_in_texts(spords, classcon['word_by_word'])
X_train_cc = np.append(X_train_cc, classcon[['word_num']], axis=1)
X_train_cc = np.append(X_train_cc, classcon[['charcount']], axis = 1)
X_train_cc.astype(int)
X_train_cc
```

Out[30]:
```
array([[   0,    0,    0, …,    0,  234, 1641],
       [   0,    0,    0, …,    0,  789, 4713],
       [   0,    0,    0, …,    0,  186, 1399],
       …,
       [   0,    0,    0, …,    0,  284, 1732],
       [   0,    0,    0, …,    0,  192, 1098],
       [   0,    0,    0, …,    0,  136,  812]])
```

In [31]:
```
logr = LogisticRegression(fit_intercept=True, solver = 'lbfgs', max_iter=1000)
nmodel = logr.fit(X_train_cc, Y_train)


training_accuracy = np.mean(nmodel.predict(X_train_cc) == Y_train)
print("Training Accuracy: ", training_accuracy)
```

```
Training Accuracy:  0.9744442965526421
```

In [32]:
```
classcon
```

Out[32]:

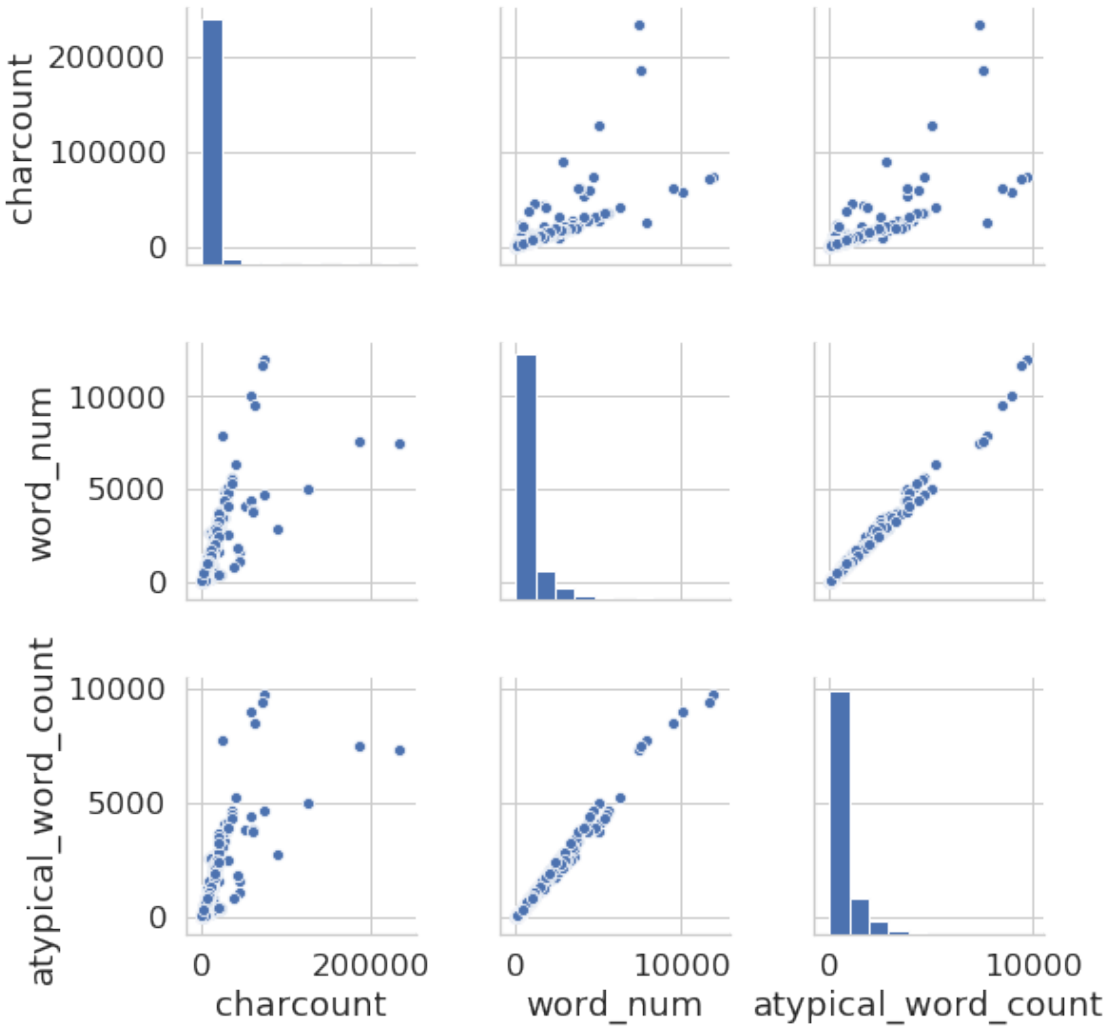
	id	subject	email	spam	charcount	no_pun	word_by_word	word_cont	word_num
0	7657	Subject: Patch to enable/disable log\n	while i was playing with the past issues, it a…	ham	1641	while i was playing with the past issues it a…	[while, i, was, playing, with, the, past, issu…	[while, was, playing, past, issues, annoyed, t…	234
1	6911	Subject: When an engineer flaps his wings\n	url: http://diveintomark.org/archives/2002/10/…	ham	4713	url http diveintomark org archives 2002 10 …	[url, http, diveintomark, org, archives, 2002,…		
2	6074	Subject: Re: [Razor-users] razor plugins for m…	no, please post a link!\n \n fox\n ----- origi…	ham	1399	no please post a link \n \n fox\n origi…	[no, please, post, a, link, fox, original, mes…		
3	4376	Subject: NYTimes.com Article: Stop Those Press…	this article from nytimes.com \n has been sent…	ham	4435	this article from nytimes com \n has been sent…	[this, article, from, nytimes, com, has, been,…		
4	5766	Subject: What's facing FBI's new CIO? (Tech Up…	<html>\n <head>\n <title>tech update today</ti…	ham	32857	html \n head \n title tech update today ti…	[html, head, title, tech, update, today, title…		
…	…	…	…	…	…	…	…		
7508	5734	Subject: [Spambayes] understanding high false …	>>>>> "tp" == tim peters <tim.one@comcast.net>…	ham	465	tp tim peters tim one comcast net …	[tp, tim, peters, tim, one, comcast, net, writ…		
7509	5191	Subject: Reach millions on the internet!!\n	\n dear consumers, increase your business sale…	spam	7054	\n dear consumers increase your business sale…	[dear, consumers, increase, your, business, sa…		
7510	5390	Subject: Facts about sex.\n	\n forwarded-by: flower\n \n did you know that…	ham	1732	\n forwarded by flower\n \n did you know that…	[forwarded, by, flower, did, you, know, that, …		
7511	860	Subject: Re: Zoot apt/openssh & new DVD playin…	on tue, oct 08, 2002 at 04:36:13pm +0200, matt…	ham	1098	on tue oct 08 2002 at 04 36 13pm 0200 matt…	[on, tue, oct, 08, 2002, at, 04, 36, 13pm, 020…		
7512	7270	Subject: Re: Internet radio - example from a c…	chris haun wrote:\n > \n > we would need someo…	ham	812	chris haun wrote \n \n we would need someo…	[chris, haun, wrote, we, would, need, someone,…		

```
1     [url, http, diveintomark, org, archives, 2002,…       789
2     [no, please, post, link, fox, original, messag…       186
3     [this, article, nytimes, com, has, been, sent,…       719
4     [html, head, title, tech, update, today, title…      5216
…                                                   …       …
7508  [tp, tim, peters, tim, one, comcast, net, writ…        61
7509  [dear, consumers, increase, business, sales, b…      1011
7510  [forwarded, by, flower, did, know, that, tell,…       284
7511  [on, tue, oct, 08, 2002, at, 04, 36, 13pm, 020…       192
7512  [chris, haun, wrote, would, need, someone, sit…       136

[7513 rows x 9 columns]
```

In [52]:
```
# Write your description (2-3 sentences) as a comment here:
# In the 4 cells below, I used a pairplot to visualize the trends between the character count
#the word count of the email body, and the 'atypical' word count of the email body as decided
# I supplemented this by plotting the heatmaps for the respective correlations. In doing so, w
#although there is generally a strong correlation between word count and atypical word count,
#there is less of a correlation between character count and the other two, and the effects are

#

# Write the code to generate your visualization here:
tab = classcon
tab['atypical_word_count'] = [len(tab['word_cont'][i]) for i in np.arange(len(tab['word_cont']]

spa = classcon.query("spam=='spam'")
s = spa.iloc[:, [4, 8, 9]].corr()
ha = classcon.query("spam=='ham'")
h = ha.iloc[:, [4, 8, 9]].corr()
sp = spa[['charcount', 'word_num', 'atypical_word_count']]
han = ha[['charcount', 'word_num', 'atypical_word_count']]

#spords[:10]
sns.pairplot(sp)
plt.title("Spam Emails")

#robust=True)
#train=train.reset_index(drop=True) # We must do this in order to preserve the ordering of ema


# hams = dataf.query('type==0')
# spams = dataf.query('type==1')

# dataf = dataf.melt('type')


# dataf = dataf.groupby(['type', 'variable']).mean().reset_index()

#sns.heatmap(dataf)

# plt.figure(figsize=[7, 6])
# plt.title('Frequency of Words in Ham/Spam Emails')
```

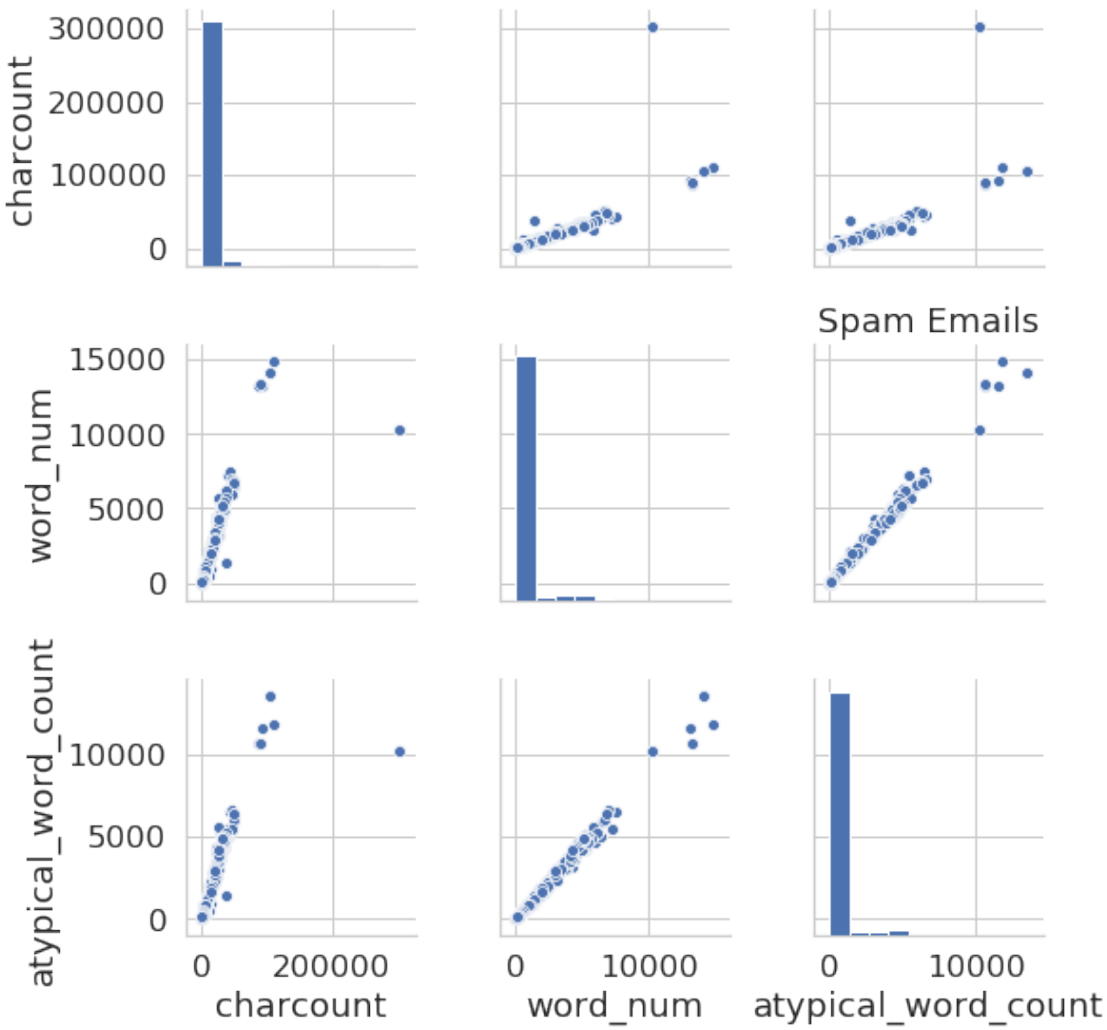
```
        #sns.barplot(data = dataf, x = 'variable', y = 'value', hue='type')
        # plt.ylim(0, 1)

        # plt.xlabel('Words')
        # plt.ylabel('Proportion of Emails')
```

Out[52]: <seaborn.axisgrid.PairGrid at 0x7ff1b1908450>



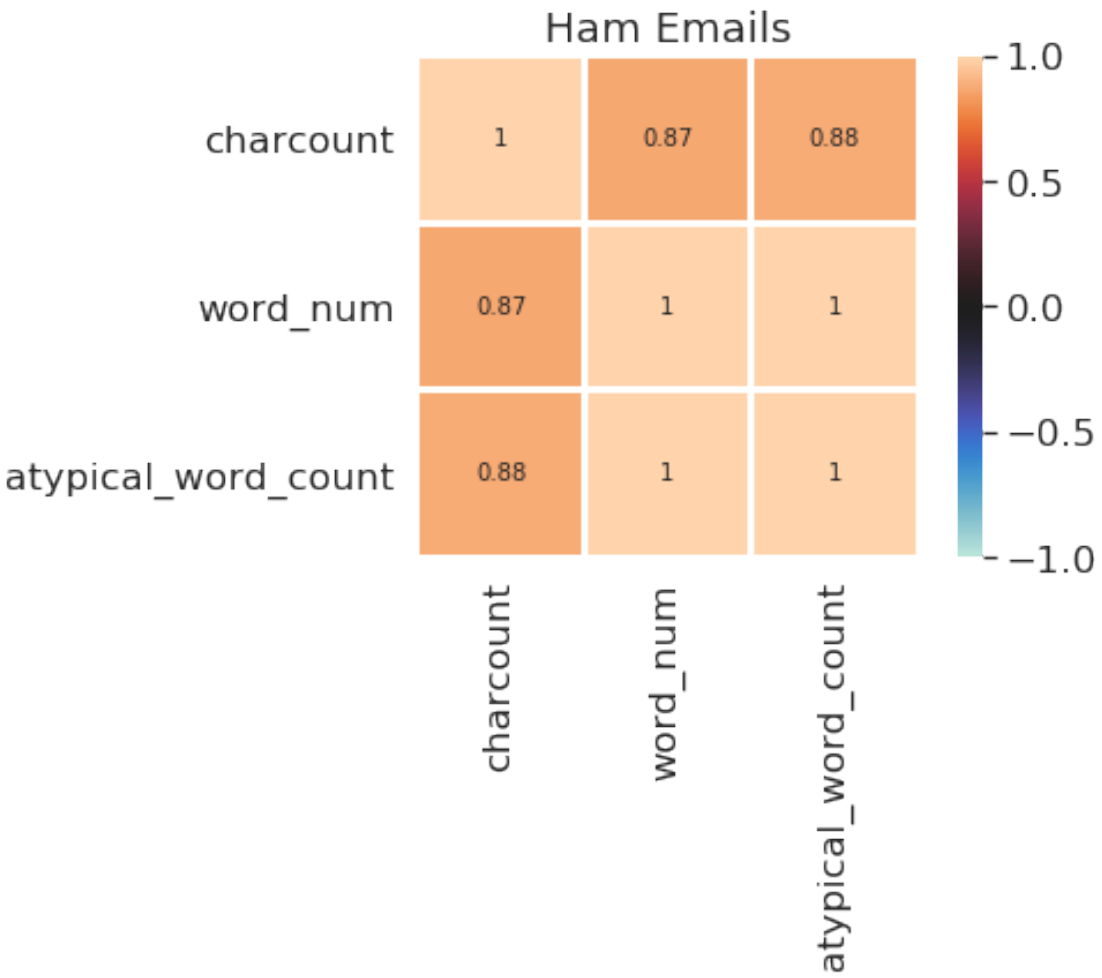
In [53]:
```
sns.pairplot(han)
plt.title("Spam Emails")
```

Out[53]: Text(0.5, 1, 'Spam Emails')



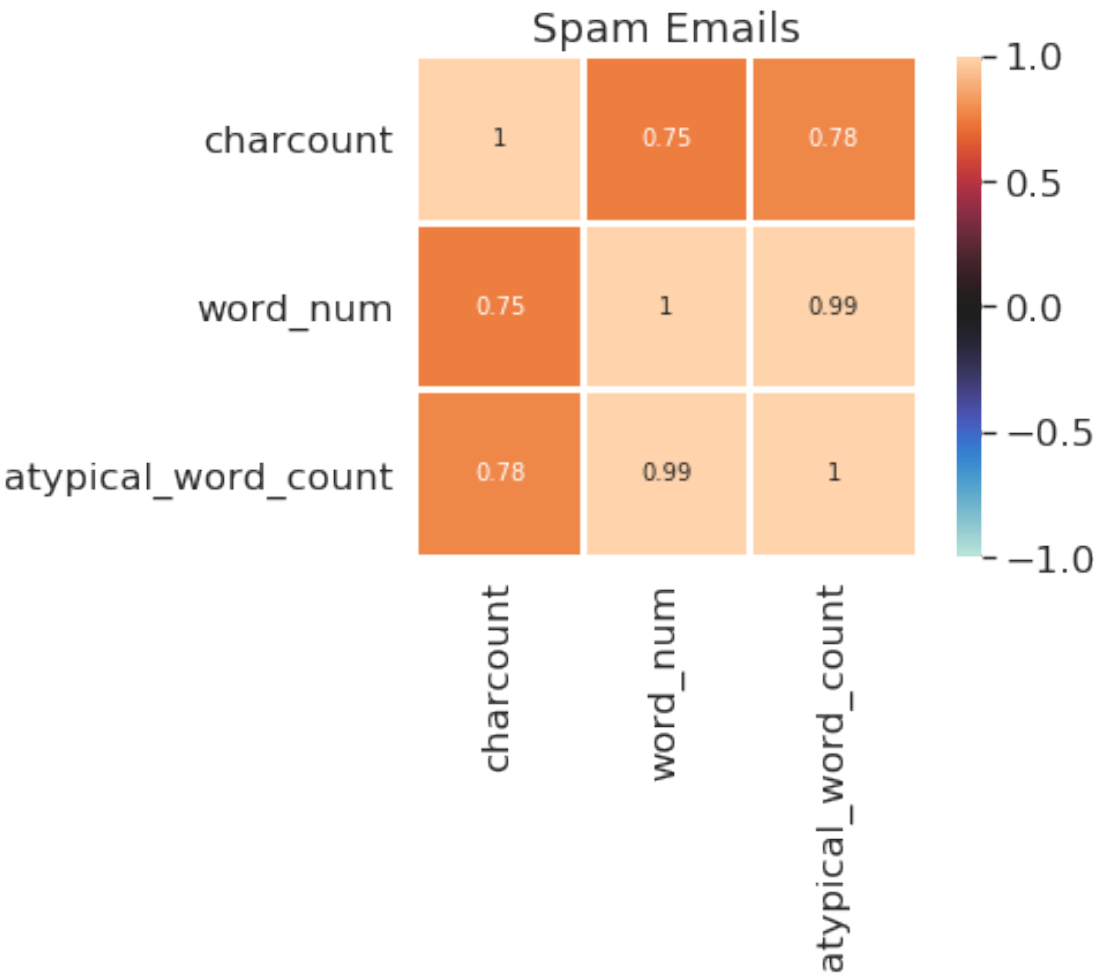

```
In [49]: sns.heatmap(h, vmin=-1, vmax=1, center=0, linewidths = 2, linecolor = 'white', annot = True, s
         plt.title('Ham Emails')
```

```
Out[49]: Text(0.5, 1, 'Ham Emails')
```



In [34]:
```
sns.heatmap(s, vmin=-1, vmax=1, center=0, linewidths = 2, linecolor = 'white', annot = True, s
plt.title('Spam Emails')
```

Out[34]:
```
Text(0.5, 1, 'Spam Emails')
```


Spam Emails
charcount
word_num
atypical_word_count
1
0.75
0.78
0.75
1
0.99
0.78
0.99
1
charcount
word_num
atypical_word_count
1.0
0.5
0.0
−0.5
−1.0

In [ ]:

0.0.8 Question 9: ROC Curve

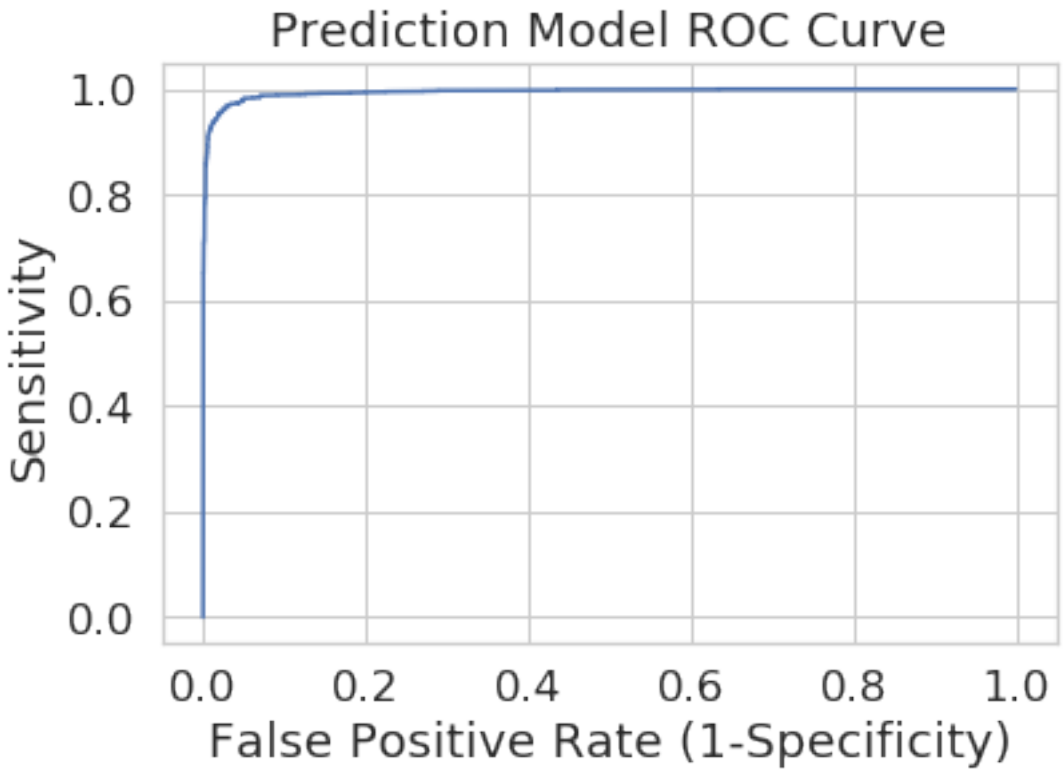
In most cases we won't be able to get no false positives and no false negatives, so we have to compromise. For example, in the case of cancer screenings, false negatives are comparatively worse than false positives — a false negative means that a patient might not discover a disease until it's too late to treat, while a false positive means that a patient will probably have to take another screening.

Recall that logistic regression calculates the probability that an example belongs to a certain class. Then, to classify an example we say that an email is spam if our classifier gives it ≥ 0.5 probability of being spam. However, *we can adjust that cutoff*: we can say that an email is spam only if our classifier gives it ≥ 0.7 probability of being spam, for example. This is how we can trade off false positives and false negatives.

The ROC curve shows this trade off for each possible cutoff probability. In the cell below, plot a ROC curve for your final classifier (the one you use to make predictions for Gradescope) on the training data. Refer to Lecture 19 or Section 17.7 of the course text to see how to plot an ROC curve.

In [35]:
```
from sklearn.metrics import roc_curve

# Note that you'll want to use the .predict_proba(...) method for your classifier
# instead of .predict(...) so you get probabilities, not classes

X_trainp = nmodel.predict_proba(X_train_cc)[:, 1]
fpr, sensitivity, threshold = roc_curve(Y_train, X_trainp, pos_label=1)
plt.plot(fpr, sensitivity)
plt.xlabel('False Positive Rate (1-Specificity)')
plt.ylabel('Sensitivity')
plt.title('Prediction Model ROC Curve')
```

Out[35]:
```
Text(0.5, 1.0, 'Prediction Model ROC Curve')
```


Prediction Model ROC Curve
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (1-Specificity)