Notebook

April 19, 2020

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.

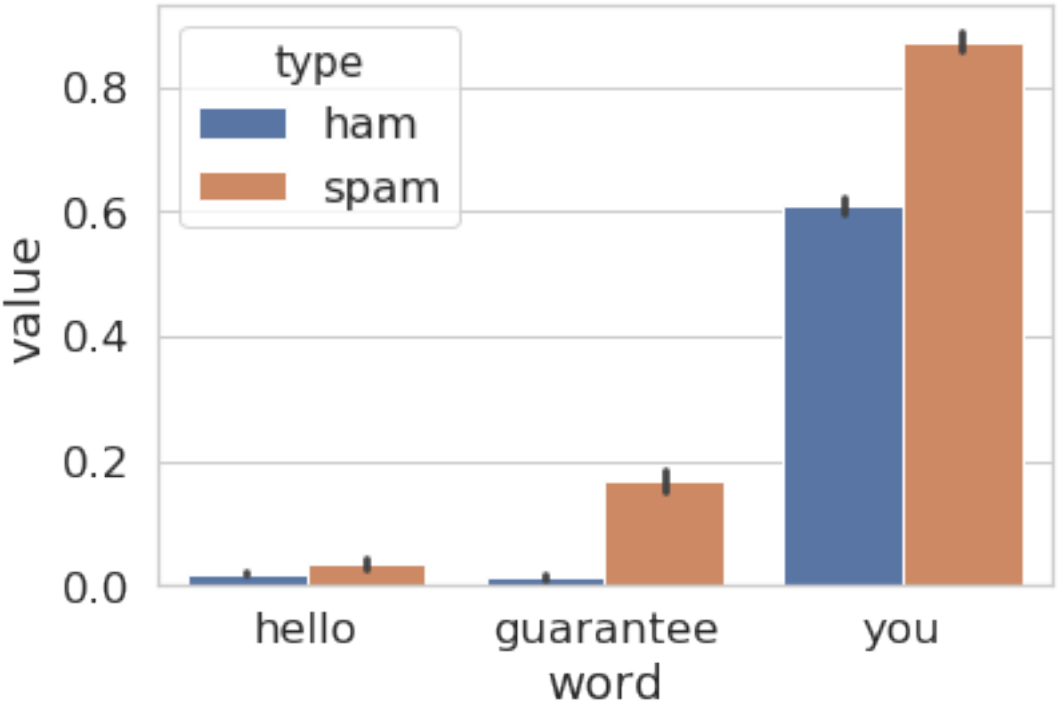
The spam email has html code in it? It also has a very suspicious looking link that starts with simply an IP address as opposed to a website.

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) # We must do this in order to preserve the ordering of emai
words = ['hello', 'guarantee', 'you']
new = pd.DataFrame(words_in_texts(words, train['email']), columns=words)
new['type'] = ['spam' if type == 1 else 'ham' for type in train['spam']]
new = new.melt('type')
new = new.rename(columns={'variable': 'word'})
sns.barplot(data=new, x='word', y='value', hue='type')
```

Out[13]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9dae691710>
```



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 [14]: d = {'type': ['spam' if type == 1 else 'ham' for type in train['spam']], 'length': train['emai
         new_2 = pd.DataFrame(data=d)
         ham = pd.Series(data=new_2.loc[new_2['type'] == 'ham']['length'])
         spam = pd.Series(data=new_2.loc[new_2['type'] == 'spam']['length'])
         sns.distplot(ham, hist=False, label = '')
         sns.distplot(spam, hist=False)
         plt.xlim(0, 50000)
         plt.show()
```
