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

Emails with words that are similar to “guaranteed success” and provides a link right before/after might be common for spam emails

0.0.1 Question 3

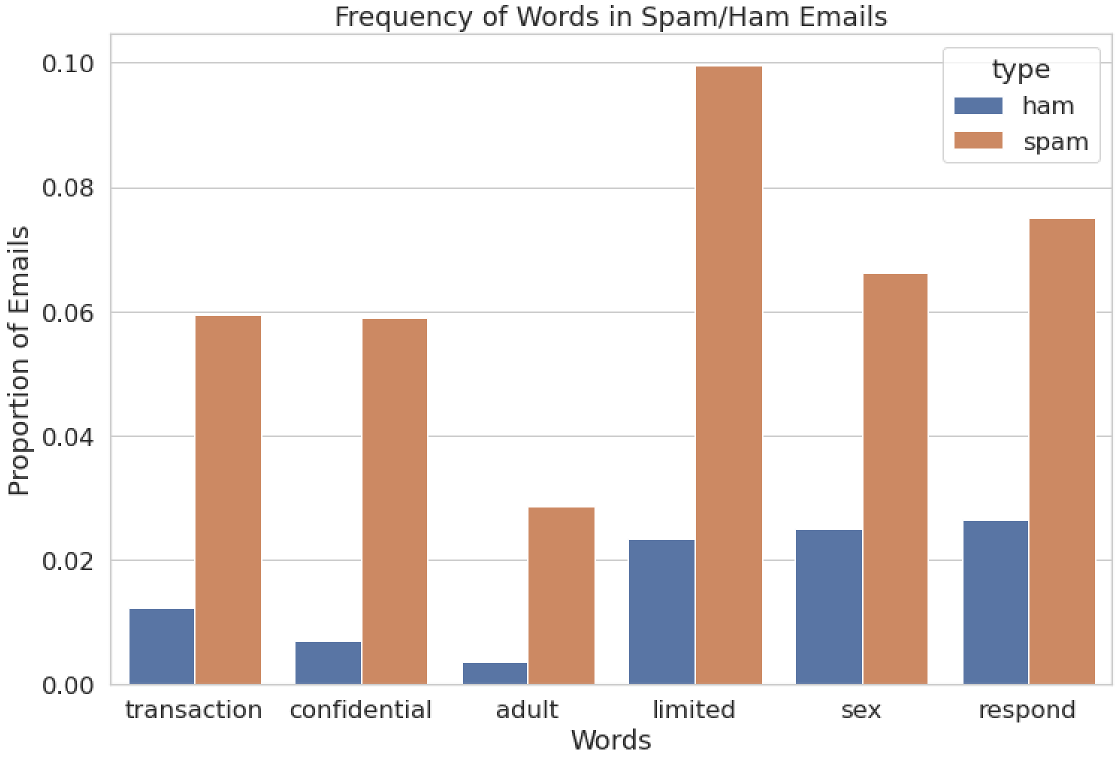
Create a bar chart like the one above comparing the proportion of spam and ham emails containing certain words. Choose a set of words that are different from the ones above, but also have different proportions for the two classes. Make sure to only consider emails from `train`.

```
In [12]: train = train.reset_index(drop=True) # We must do this in order to preserve the ordering of em

         words = ["transaction", "confidential", "adult", "limited", "sex", "respond"]

         words_in_train = pd.merge(left = pd.DataFrame(words_in_texts(words, train["email"]), columns =
                                   right = train["spam"].apply(lambda x: "spam" if x == 1 else "ham"),
                                   left_index = True, right_index = True). \
                         rename({"spam": "type"}, axis = 1)

         plt.figure(figsize = (12, 8))
         sns.barplot(data = words_in_train.melt("type"),
                     x = "variable",
                     y = "value",
                     hue = "type",
                     ci = None)
         plt.xlabel("Words")
         plt.ylabel("Proportion of Emails")
         plt.title("Frequency of Words in Spam/Ham Emails");
```



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.

FP: is 0 since the zero predictor never predicts positive, therefore can never have any false positives
FN: is the correct number of positives in the Y_train. This is because if we predict everything as negative, all of the “truly positives” will become false negatives
Accuracy: is number of true negatives divided by the size of Y_train since the only correct (i.e. accurate) predictions are the true negatives as there is no true positive
Recall: is 0 since there is no true positive as we only ever predict negative

0.0.3 Question 6e

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

fp = np.sum((predictions_train != Y_train) & (predictions_train == 1)) = 122
fn = np.sum((predictions_train != Y_train) & (predictions_train == 0)) = 1699
false negatives

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. It is sligthly higher (75.76% vs 74.47%)
2. Since there the frequency never exceeds 5% of any of the words, it is likely that there are many emails (i.e. rows) in X_train that is all zero, which means that they do not contain any of the words. Therefore, even if an email is spam, as long as it doesn't contain any of the words, it would be labeled as ham.
3. Since the whole point is to filter out spams, I would prefer the logistic model as it has a higher precision albeit the relatively low recall.