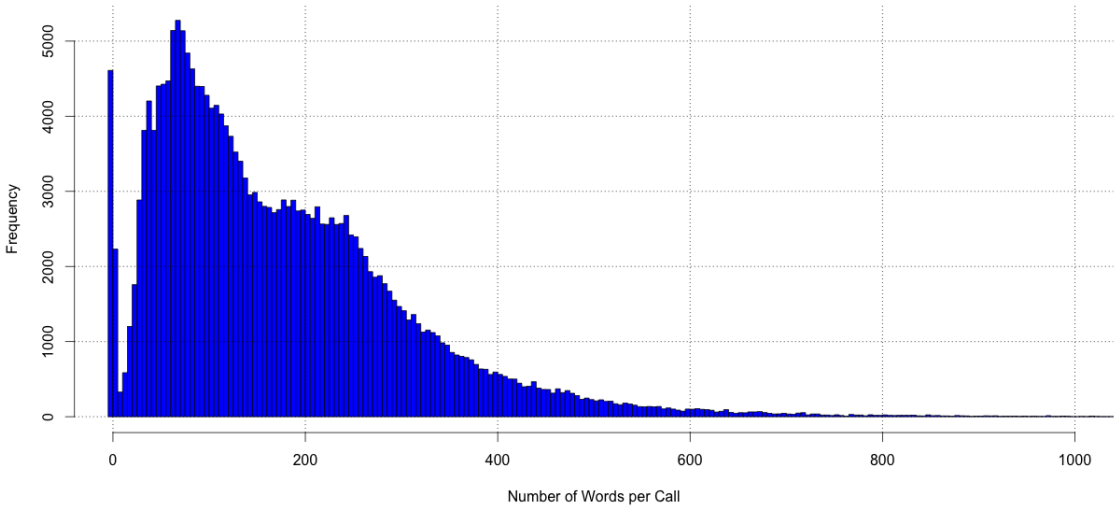
1 Project Scope

I analyzed 235,144 records from the WSH Phone Notes database table. As discussed with Hari, a number of these records spanned a signal phone conversation. Hence, I found that there were roughly 190,000 unique conversations and roughly 172,000 of those were populated with text appropriate for the classification. As Figure 1 below indicates, there are a total of eight class types contained in the database. For the purpose of the analysis, CONTINUED records were appended to their parent record and JUNK and OUTGOING records were ignored.

Table 1: **Summary of WSH Phone Call Data Table**

Category	Count	Status	Share
APPOINTMENTS	45,340	Classified	26.3
ASK_A_DOCTOR	29,607	Classified	17.2
CONTINUED	44,359	Classified	
JUNK	17,237	Ignored	
LAB	14,293	Classified	8.3
MISCELLANEOUS	34,384	Classified	19.9
OUTGOING	1,120	Ignored	
PRESCRIPTION	48,804	Classified	28.3
Total	235,144		100.0
w/o Continued	190,785		
w/o JUNK & OUTGOING	172,428		

Figure 1: **Word Distribution per Conversations**



2 Text Cleaning

The storage of the textual data in Rich Textual Format (RTF) presented some initial challenges, yet the conversion of the RTF into clean text in Python was fairly straightforward. However, I did make a number of adjustments, in order to make sure that the classification was being done in a manner consistent with the current WSH approach, including:

- Ignoring the *Reason for Call* data in the free text field.
- Limiting the classification to the initial portion of the text prior to the Follow-up sections.
- Client names and phone numbers were also excluded in the classification, as they did not represent desirable machine learning features.

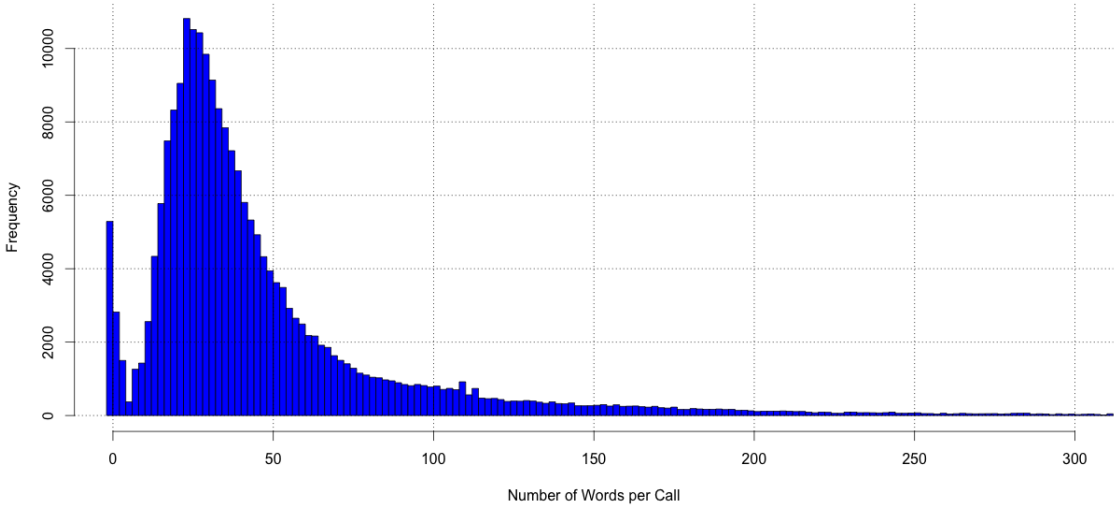
3 Feature Engineering

For the purpose of this Proof of Concept, the feature engineering of the textual data was rather limited using simple uni-gram and bi-gram approaches. Ideally, if this were to turn into a production system, I did notice a number of features, which could be valuable in improving accuracy further, including:

- Standardizing common abbreviations such as *pt* for *patient*.
- Expanding features such as date and time fields via regular expressions.
- Adding tags for when specific drugs and/or procedures are mentioned, in order to improve PRESCRIPTION accuracy.
- Adding special tags for LAB specific terms.

I examine making adjustments for the high frequency of documents with only a few words in them, as Figures 1 and 2 indicate, by excluding documents with less than 5 words, yet this did not make for a material difference in the classification accuracy.

Figure 2: **Word Distribution per Conversations - Cleaned Text**



4 Results

The data was broken into two datasets with roughly 84% of it for training and 16% for test purposes. Additionally, a number of tests were run to examine a number of two assumpitons with the data as detailed below in Table 2. The first two rows labelled as *Base Case* for Features represent my best understanding of the current text being used to classify the WSH data. As such, the classifier was able to achieve roughly 75% accuracy using either uni- or bi-grams in the Test dataset. As is typical with higher n-grams models, in this case with a bi-gram approach, a much higher accuracies were achieved in the Training dataset, yet with roughly the same results in Test, as overfitting tends to occur.

I then moved on to classify the data by testing two variations with the textual data. In the *with Summary* case, I have included the SUMMARY field into the classification text, as it would seem logical to add it to the classification, much like using article titles in the financial press to determine overall article content. Adding this text turns out to be quite useful in raising the Test accuracy to nearly 80%. Finally, I re-estimated the classifier allowing all text from the entry into the classification, even if it were of a follow-up nature. As one can see, adding more text actually worsens the results slightly, as the extra text does little to explain the reason for the call.

Table 2: **Summary of WSH Phone Call Classification Results**

		Accuracy (%)	
N-Gram	**Features**	**Train**	**Test**
1	Base Case	79.1	75.6
2	Base Case	96.2	74.4
1	with Summary	83.3	78.3
2	with Summary	98.4	78.0
1	with Summary & Full Text	84.1	78.4
2	with Summary & Full Text	99.0	77.5

Below, I provide the confusion matrix for the best model, namely the Uni-gram approach allowing the Summary Data to be used in the classification. As one can, there is strong clustering along the diagonal of the confusion matrix, which is to be expected for a robust model. The most accurate predictions are for Ask-A-Doctor and for Lab work, which seems intuitive. The Miscellaneous category seems to be causing the greatest number of false positives and false negatives especially with Appointments.

Table 3: **WSH Phone Call Confusion Matrix for Using Unigrams and Summary Field Data**

Class	**APPTS**	**ASK_A_DR**	**LAB**	**MISCELL.**	**PRESCR.**	**Total**
APPOINTMENTS	7,701	503	112	625	112	9,053
ASK_A_DOCTOR	282	3,918	86	433	344	5,063
LAB	267	180	3,129	427	160	4,163
MISCELLANEOUS	596	663	231	4,716	348	6,554
PRESCRIPTION	153	480	83	431	2,989	4,136
Class Accuracy	**85.1**	77.4	**87.5**	72.0	72.3	

5 Conclusion

This basic analysis has established that a high degree of accuracy can be achieved without manual curation of the call center data. If there were interest in putting this system into production, I recommend that features previously discussed in this write-up be added to the classification model. With further feature engineering and error analysis on the confusion matrix, I believe that automated accuracy would reach into the low 80% range. Accuracy could be raised further with further manual curation for low probability classifications, where there is confusion between the five classes. Further, a topic model to re-evaluate the classes choices could be examined, especially with respect to the sub-classifications, which I have left for future work.