**I. Abstract**

Good-Turing is a method that’s used to estimate the probability of the unseen words or objects. At the lower frequency, Good-Turing is very good technique in estimating the unseen words or object. As the frequency is getting bigger, the probability of using Good-Turing method is not very accurate. However Simple Good-Turing’s estimate is a very good method when r is big. As the result, the Good-Turing will be used to estimate the unseen words/objects probability at the lower frequency because it is more accurate than SGT and at higher frequency, the SGT will be used. In order to determine when to switch between Good-Turing and SGT, Turing variance will be used to determine where the breakpoint will be.

**II. Introduction**

From a given words or speech, there can be infinite possibility number of combination from a given set of words or speech. By figuring out the probability of the unseen words/objects, the performance of the system can be significantly optimized. Good-Turing is a technique that used to estimate the probability of unseen words or objects. However, Good-Turing’s estimate is only accurate at the lower frequency. As the frequency increases, Good-Turing’s estimate is not a good method to use for estimating the probability when r is getting large. For all of Nr that are zero, they do contribute some to the over all probability, so we need to smooth out those regions and re-estimate the probability by using Simple-Good-Turing (SGT).

**III. Methods**

In order to re-estimate the raw r of the regions that Nr are zeros, the steps are as follow:

1. Re-estimate the Nr and call it Zr
2. Find a polynomial function (Linear regression) that fit with the data
3. Using the slope of polynomial function to estimate r for SGT and call it SGT\_r\*
4. Set up the threshold by using Turing variances
5. For lower r, the Good-Turing’s estimate will be used and for large r, SGT will be used
6. Switching from Good-Turing to SGT when | Turing\_r\* SGT\_r\* | < threshold, threshold is defined by 1.65 \* sqrt( Turing\_variance )

**IV. Results**

The data show that when r is small, the raw Nr appear to be in linear line. However, the raw Nr appear to be zero as r is getting larger. In figure 1, the graph is plotted in log scale, approximately right after the value log(r) = 2, the Nr values start to collapse into each other. At the value log(r) = 4.8, the Nr values are started to get flat and at the value log(r) = 5.5, all of the Nr are completely zero. In order to smooth out the regions of Nr that are zero, re-estimating the Nr is necessary. The Nr will be replace by the Zr and the Nr at the position of r will be divided by the average position of before r plus after r.

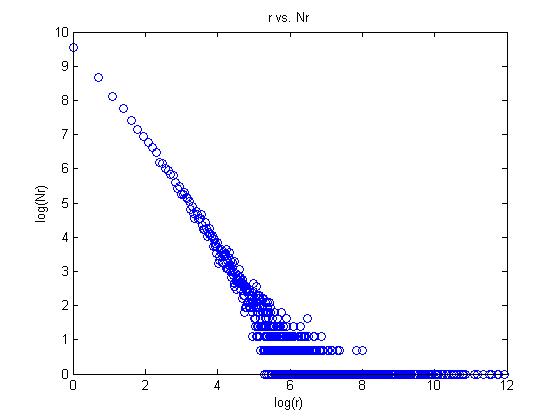


Figure 1: Raw r vs. Nr in log scale

The figure 2 shows r vs. Zr in log scale and the data almost fit into a linear polynomial function. The figure 2 also shows the regions where Nr were zeros in figure 1 are completely smoothed out by re-estimating the Nr or the Zr.

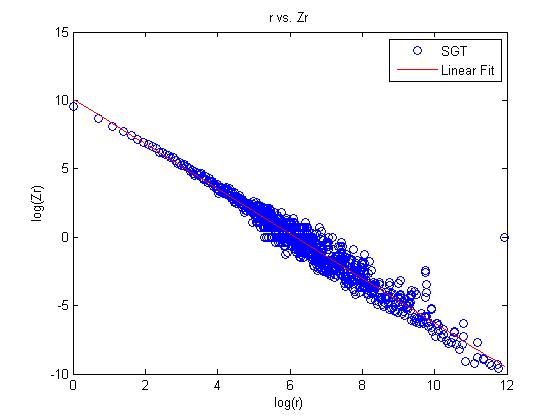


Figure 2: r vs Zr in log scale

From the polynomial fit function when the Nr were re-estimated, the slope can be obtained and used to estimate the r\* for SGT method. In figure 3, the graph shows that as r increases, the SGT\_r\* / r is approaching to 1 very fast. This is the second prior expectation that Gale is describing in the article, “Good-Turing Smoothing Without Tears”.

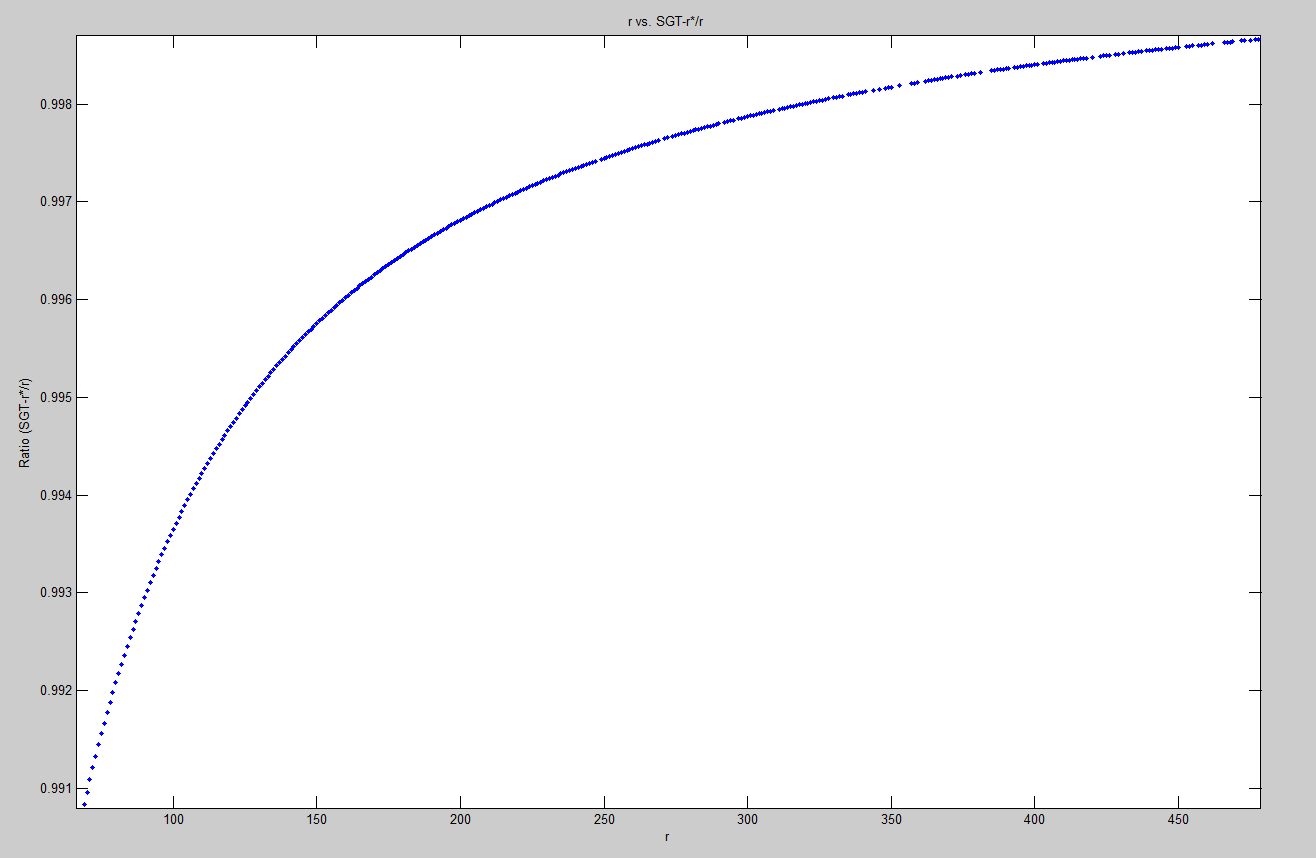


Figure 3: r vs. SGT\_r\* / r. zoomed in from 0 to 400

**V. Discussion**

After smoothed out the regions that Nr are zero, the probabilities in those region has been accounted toward the overall probability. The Good-Turing method that’s used to re-estimate the r is very accurate for lower r and when r gets large, the method is not a good candidate for estimating r\*. The Figure 4 shows the first 40 value of r plot vs (Turing\_r\* / r). When r is small, the data tend to stay together and converging to one. However, the data tend to scatter around as r is getting larger.

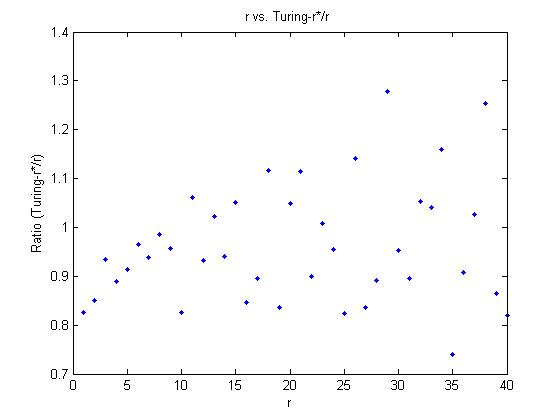


Figure 4: r vs. Turing\_r\* for first 40 r

On the other hand, Simple Good-Turing is a great method in estimating large r. The figure 4 shows the first 40 values of r plot vs. (SGT\_r\*/r) and the graph is look completely different than figure 3. Notice that the data tend to converge to 1 as r is getting larger.

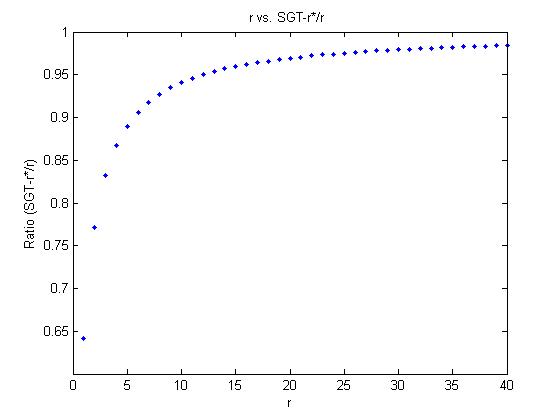


Figure 5: r vs. SGT\_r\* / r for first 40 value of r

Since Good-Turing’s estimate is great in estimating the low r and SGT method is good in estimating when r is big, the hybrid of the two will be used. In order to determine where to switch from using Good-Turing to SGT, a threshold will determine where the break point will be. The threshold is defined as Turing variances and the difference of Turing\_r\* and SGT\_r\*. If the difference of | Turing\_r\* and SGT\_r\* is less than the 1.65\*sqrt(Turing variance), then the SGT will be used to estimate for the rest of the r. Once the method switches, the r\* is getting bigger; the figure shows r\* / r is approach to 1 as r is getting bigger. The figure 3 also shows that when the r is small, the r\* is much smaller than r. For the first few points of r, the ratio of r\*/r are below 0.9; this shows that r\* is only 90% the value of r or less. The discounted probability (r\*) are used to estimate N-Gram because the unseen words/objects have some non-zero probabilities and it must be accounted for in the overall probability; therefore, we need to reduce the probability of all of the words/objects have been seen and leaving some probabilities for unseen words/objects.

**VI. Summary**

In using the raw values of r and Nr, there are many regions that Nr got truncated. In order to smooth out those regions, the Nr need to get re-estimated. Once the Nr got re-estimated, the slope can be obtained by using the polynomial fit function (Linear Regression for this case). The slope be used to re-estimate the r\* for Simple Good-Turing (SGT). Between Good-Turing’s estimate and SGT’s estimate, there are good and bad side of both in term of accuracy, Good-Turing estimate are great for small r, but not for the big r; SGT is a great estimate to use for big r but not as accurate as Good-Turing for small r. Therefore, the combination of both estimates will be used and the threshold between switching from one to another is using the Turing variance and the different between Good-Turing and SGT.