PROJECT REPORT

Aim of the Project

The aim of the project is to analyse the change in syntactic complexity of a person's sentence structure with increase in experience.

Data Collection

The data for this project is collected from Michael Collins's google scholar page with all the papers where he is either a first author or the only author dated from 1995 to 2012. For the sake of results to not be influenced by other authors, in this report, I focus only on papers where Michael Collins is the sole author.

Every paper was in a PDF format, so I used python's PDFminer library to convert the PDF document into text format. These text files were then converted to list of sentences using NLTK's sentence tokenizer.

Evaluating Measures

There are three measures used to calculate the syntactic complexity of a sentence.

1. Yngve scoring

   This scoring mechanism works by calculating the size of the stack when the parse tree is traversed from top-down and left-right. When a terminal node is reached, we add the score of all the branches from root to terminal node. To calculate the size of the stack, for each node in the tree, label all the children of the node from rightmost child to leftmost child starting with a score of zero and incrementing by one for every child node. Final score is the mean, which is calculated as sum of scores of all the terminal nodes divided by number of terminal nodes.

2. Frazier scoring

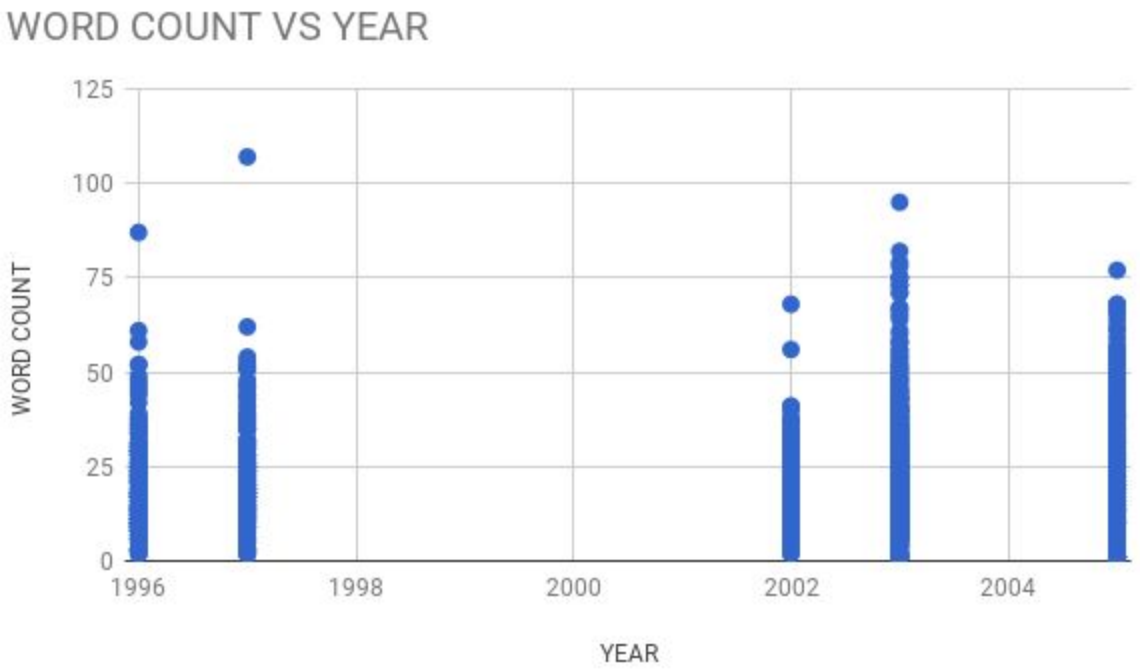
   This scoring mechanism follows a bottom-up approach. Every terminal node follows a path up to the root node or the lowest node which is not the leftmost child of the parent. Once each path is separated, a score of 1 is assigned to every non-terminal branch and a score of 1.5 for sentence node or sentence-complement nodes. The final score is the mean which is calculated as the sum of all the scores divided by the total number of terminal nodes.

3. Dependency distance

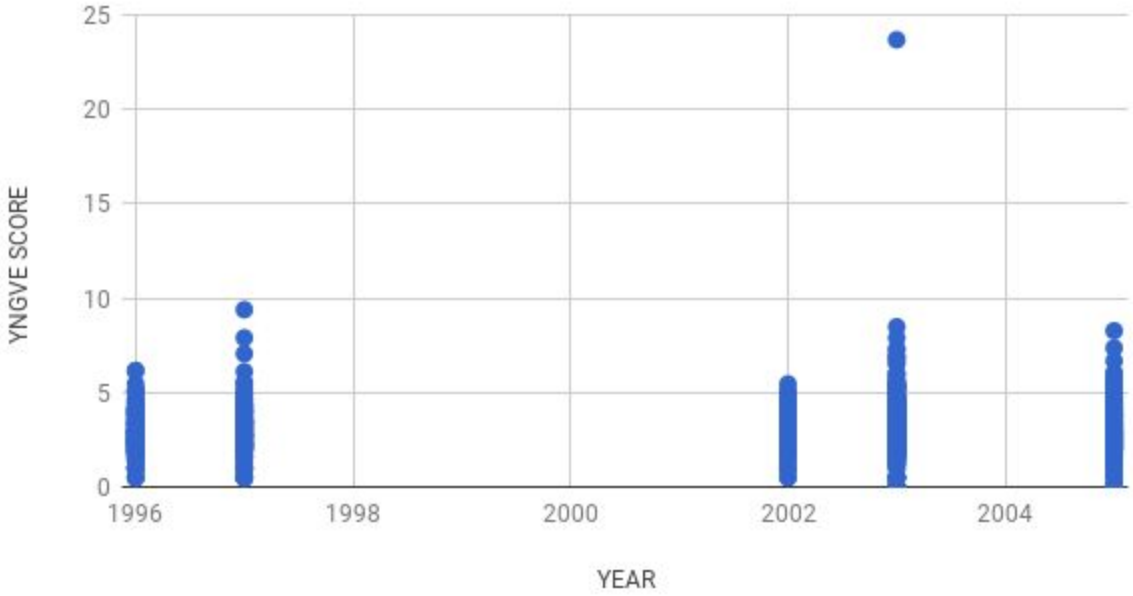
    To calculate the dependency distance for a sentence, we first find a dependency parse of the sentence. For every arc in the dependency parse tree, we calculate the distance from parent node to child node. The final score of the sentence is again, the mean value.

Results and Observations

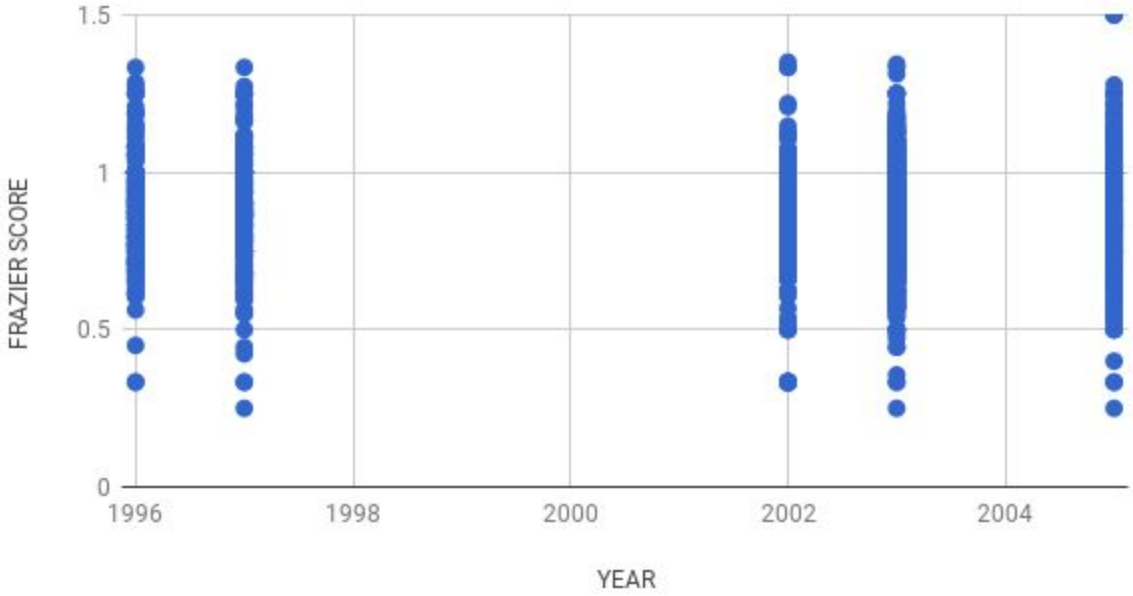
Following are the graphs plotting each complexity measure vs year :

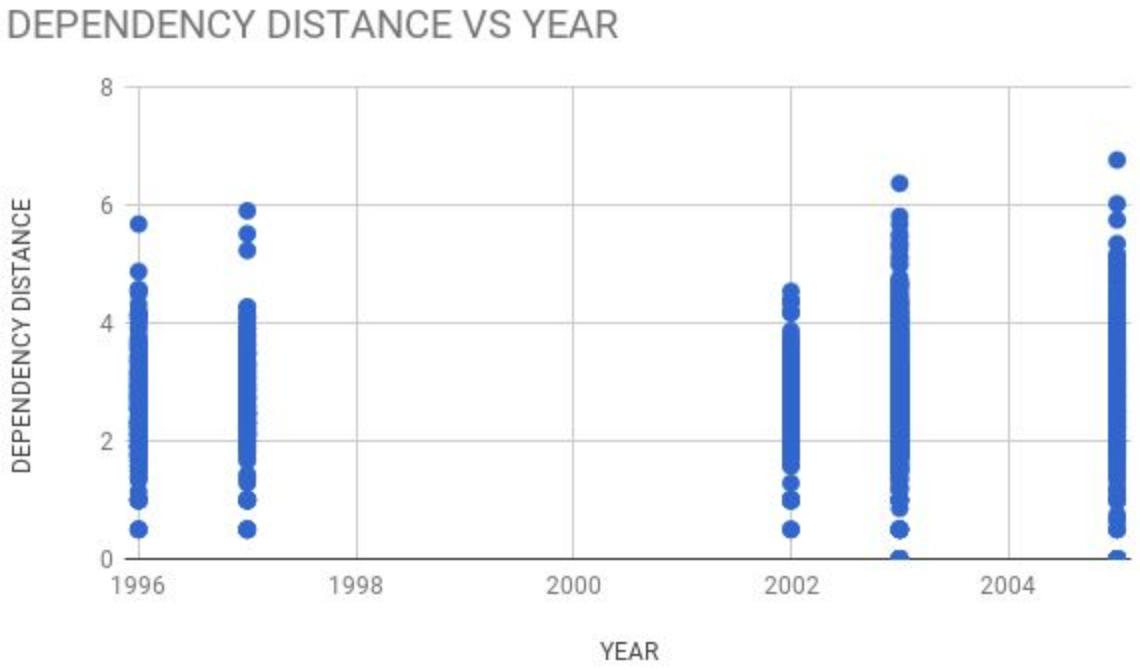


YNGVE SCORE VS YEAR



FRAZIER SCORE VS YEAR





The graphs shows the distribution of sentence's syntactic complexity over the years (specifically, 1996, 1997, 2002, 2003 and 2005). As we can see, the complexity of sentences increases slightly in 2003 and 2005 for all the scoring measures.

Following tables show the distribution of count of scores :

WORD COUNT					
	1996	1997	2002	2003	2005
0-15	90	55	56	241	147
15-25	70	64	48	258	148
25-50	59	55	39	242	184
50-75	4	7	2	27	24
75 Above	1	1	0	7	1

YNGVE SCORE					
	1996	1997	2002	2003	2005
0-2	37	22	24	65	75
2-4	159	130	107	593	359
4-6	26	26	14	106	66
6-8	2	3	0	9	3
8 Above	0	1	0	2	1

FRAZIER SCORE					
	1996	1997	2002	2003	2005
0-0.5	3	5	8	8	4
0.5-0.75	47	34	35	168	81
0.75-1	119	91	71	435	276
1-1.25	47	45	28	157	137
1.25 Above	8	7	3	7	6

DEPENDENCY DISTANCE					
	1996	1997	2002	2003	2005
0-1	3	5	2	16	46
1-2	45	21	25	94	55
2-4	159	146	111	598	350
4-6	17	10	7	66	52
6 Above	0	0	0	1	2

Here, word count works as a baseline measure. It is observed that, the trend of all the scoring mechanism is similar to that of the baseline. As the number of words increase, the complexity score also increases.

Challenges and Limitations

1. Some data is lost when converting from PDF file to text file.
2. Many sentences in the paper include use of equations, formulas or numbers which were marked as seperate terminal nodes. This causes the increase in noisy data because, these equations do not parse properly into a tree.
3. There were lack of papers year wise.
4. The length of the paper seems to play an important role in the scores. It would make sense to perform this analysis when we can collect data which is uniform and consistent.

References

1. Brian Roark, Margaret Mitchell and Kristy Hollingshead. “Syntactic complexity measures for detecting Mild Cognitive Impairment”.
2. https://github.com/euske/pdfminer
3. http://www.nltk.org/api/nltk.tokenize.html
4. https://github.com/neubg/util-scripts
5. https://stanfordnlp.github.io/CoreNLP/
6. https://github.com/smilli/py-corenlp
7. https://spacy.io/models/