

The Flesch Index: An Easily Programmable Readability Analysis Algorithm

JOHN TALBURT

University of Arkansas at Little Rock

This paper is an exposition of an algorithm for text analysis that can be of value to writers and documentalists. The simplicity of this algorithm allows it to be easily programmed on most computer systems. The author has successfully implemented this test as a function within a text editing system written in RPG II. Included in this paper is a sample program written for the VAX 11/780 in PL/I.

In 1949 Dr. Rudolph Flesch published a book titled "The Art of Readable Writing." In this book, he described a manual method of reading ease analysis. This method was to analyze text samples of about 100 words. Each sample is assigned a readability index based upon the average number of syllables per word and the average number of words per sentence. This Flesch Index is designed so that most scores range from 0 to 100. Only college graduages are supposed to follow prose in the 0 - 30 range. Scores of 50 -60 are high-school level and 90 - 100 should be readable by fourth graders.

Though crude, since it is designed simply to reward short words and sentences, the index is useful. It gives a basic, objective idea of how hard prose is to wade through. This test has been used by some state insurance commissions to enforce the readability of policies.

Flesch's algorithm was automated in the early 1970s by the Service Research

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1986 ACM 0-89791-186-5/86/0600/0114 \$00.75

Group of the General Motors Corporation. The program, called GM-STAR (General Motors Simple Test Approach for Readability) was used so that shop manuals could be made more readable. GM-STAR was originally written in BASIC language. The key to this program is a very simple algorithm to count the number of syllables in a word.

In general the text analysis portion of the program uses the following rules:

1. Periods, exclamation points, question marks, colons and semi-colons count as end-of-sentence marks.
2. Each group of continuous non-blank characters counts as a word.
3. Each vowel (a, e, i, o, u, y) in a word counts as one syllable subject to the following sub-rules:
 - A. Ignore final -ES, -ED, -E (except for -LE)
 - B. Words of three letters or less count as one syllable
 - C. Consecutive vowels count as one syllable.

Although there are many exceptions to these rules, it works in a remarkable number of cases.

The Flesch Index (F) for a given text sample is calculated from three statistics;

1. The total number of sentences (N),
2. The total number of words (W),
3. The total number of syllables (L),

according to the following formula:

$$F = 206.835 - 1.015 \times (W/N) - 84.6 \times (L/W).$$

The Grade Level Equivalent (G) of the Flesch Index is given by the following table:

If -50	F	50, then $G = (140 - F)/6.66$
If 50	F	60, then $G = (93 - F)/3.33$
If 60	F	70, then $G = (110 - F)/5.0$
If 70	F	, then $G = (150 - F)/10.0$

A PL/I program that implements this algorithm is listed below along with sample output. For simplicity, this program assumes all letters are in upper case. Processing text with lower case letters can be accomplished by either modifying the program to test for lower case as well as upper case, or by preprocessing the text sample to translate all letters to upper case. There are a multitude of other refinements and amenities that can be added to the basic analysis. Among these are:

1. Nothing which characters are considered sentence terminators.
2. Ignoring periods that are used for abbreviations rather than sentence

terminators.

3. Ignoring word connecting hyphens in compound words.
4. Noting which character groups should probably be spelled out, such as numerals and dollar amounts.
5. Sharpening the syllable counting routine to detect exceptional cases.

```

GMSTAR: PROCEDURE OPTIONS(MAIN);
  /* THIS PROGRAM READS A USER CREATED TEXT FILE AND */
  /* COMPUTES THE FLESCH READABILITY INDEX.          */
  DCL TEXT_IN   FILE INPUT  STREAM;
  DCL REPORT    FILE OUTPUT PRINT;
  /*** TEXT WORK VARIABLES ***/
  DCL LINE      CHAR(80);
  DCL WORD      CHAR(30);
  DCL TCHAR     CHAR(1);
  DCL SUFFIX    CHAR(2);
  DCL WORD_START BINARY;
  DCL WORD_END   BINARY;
  DCL WORD_LEN   BINARY;
  DCL V_COUNT    BINARY;
  DCL (J,K,L)    BINARY;
  /*** FLAGS ***/
  DCL MORE_LINES BIT(1) INIT('1'B);
  DCL MORE_WORDS BIT(1);
  DCL SEARCH_FLAG BIT(1);
  DCL VOWEL_FLAG  BIT(1);
  /*** STATISTICS ***/
  DCL FLESCH_INDEX  FLOAT;
  DCL GRADE_LEVEL   FLOAT INIT(0);
  DCL AVG_SENT      FLOAT;
  DCL AVG_SYLL      FLOAT;
  DCL SENT_COUNT    BINARY;
  DCL WORD_COUNT    BINARY;
  DCL SYLL_COUNT    BINARY;
  /*** CHARACTER CONSTANTS ***/
  DCL VOWELS        CHAR(6) INIT('AEIOUY');
  DCL TERMINATORS   CHAR(5) INIT('.,;!?'');
  DCL BLANK          CHAR(1) INIT(' ');
  DCL COMMA          CHAR(1) INIT(',');
  /*******/
  /*** BEGIN MAIN PROCEDURE ***/
  /*******/
  ON ENDFILE(TEXT_IN) MORE_LINES = '0'B;
  GET FILE(TEXT_IN) EDIT(LINE) (A(80));
  DO WHILE (MORE_LINES);
    J = 1;
    CALL NEXT_WORD;
    DO WHILE (MORE_WORDS);
      CALL TEST_TERM;
      WORD_LEN = WORD_END - WORD_START + 1;
      WORD = SUBSTR(LINE,WORD_START,WORD_LEN);
      WORD_COUNT = WORD_COUNT + 1;
      IF WORD_LEN > 3
        THEN DO;
          CALL TEST_SUFFIX;
          CALL VOWEL_COUNT;
        END;
      ELSE DO;
        V_COUNT = 1;
      END;
    END;
  END;

```

```

        SYLL_COUNT = SYLL_COUNT + 1;
    END;
    CALL NEXT_WORD;
    END;
    PUT SKIP EDIT(LINE) (A(80));
    GET FILE(TEXT_IN) EDIT(LINE) (A(80));
    END;
CALL SUMMARY;
/***** SUBROUTINE SECTION *****/
/**** ALL SUBROUTINES ARE CALLED FROM THE MAIN ****/
/**** PROCEDURE AND CALL NO OTHER PROCEDURES ****/
/**** NEXT_WORD SETS WORD_START AND WORD_END POINTERS ****/
/**** *****/
NEXT_WORD: PROCEDURE;
    MORE_WORDS = '0'B;
    SEARCH_FLAG = '1'B;
    DO WHILE (SEARCH_FLAG);
        IF SUBSTR(LINE,J,1) = BLANK
            THEN
                IF J = 80
                    THEN
                        SEARCH_FLAG = '0'B;
                    ELSE
                        J = J + 1;
                ELSE DO;
                    SEARCH_FLAG = '0'B;
                    MORE_WORDS = '1'B;
                    WORD_START = J;
                END;
            END;
        IF MORE_WORDS
            THEN
                SEARCH_FLAG = '1'B;
        DO WHILE (SEARCH_FLAG);
            IF SUBSTR(LINE,J,1) = BLANK
                THEN DO;
                    SEARCH_FLAG = '0'B;
                    WORD_END = J - 1;
                END;
            ELSE DO;
                IF J = 80
                    THEN DO;
                        SEARCH_FLAG = '0'B;
                        WORD_END = 80;
                    END;
                ELSE
                    J = J + 1;
            END;
        END;
    END NEXT_WORD;
/**** TEST_TERM LOCATES SENTENCE TERMINATORS ****/

```

```

/*****
TEST_TERM: PROCEDURE;
  TCHAR = SUBSTR(LINE,WORD_END,1);
  K = INDEX(TERMINATORS,TCHAR);
  IF K > 0
    THEN DO;
      SENT_COUNT = SENT_COUNT + 1;
      WORD_END = WORD_END - 1;
    END;
  IF TCHAR = COMMA
    THEN
      WORD_END = WORD_END - 1;
  END TEST_TERM;
/*****
/**** TEST_SUFFIX FINDS SUFFIXES -ED, -ES, AND -E NOT -LE ****
/****
TEST_SUFFIX: PROCEDURE;
  SUFFIX = SUBSTR(LINE,WORD_END-1,2);
  TCHAR = SUBSTR(LINE,WORD_END,1);
  IF SUFFIX = 'ED'
    THEN
      WORD_END = WORD_END - 2;
  IF SUFFIX = 'ES'
    THEN
      WORD_END = WORD_END - 2;
  IF TCHAR = 'E' & SUFFIX ^= 'LE'
    THEN
      WORD_END = WORD_END - 1;
  END TEST_SUFFIX;
/****
/**** VOWEL_COUNT EQUATES VOWELS AND SYLLABLES ****
/****
VOWEL_COUNT: PROCEDURE;
  K = WORD_START;
  VOWEL_FLAG = '0'B;
  V_COUNT = 0;
  DO WHILE (K <= WORD_END);
    TCHAR = SUBSTR(LINE,K,1);
    L = INDEX(VOWELS,TCHAR);
    IF L = 0
      THEN
        VOWEL_FLAG = '0'B;
      ELSE
        IF ^VOWEL_FLAG
          THEN DO;
            VOWEL_FLAG = '1'B;
            V_COUNT = V_COUNT + 1;
          END;
        K = K + 1;
      END;
    SYLL_COUNT = SYLL_COUNT + V_COUNT;
  END VOWEL_COUNT;
/****

```

```
**** SUMMARY CALCULATES AND PRINTS ANALYSIS STATISTICS ****/  
/*****  
SUMMARY: PROCEDURE;  
  PUT SKIP(2) EDIT('NUMBER OF SENTENCES',SENT_COUNT) (R(FORM_1));  
  FORM_1: FORMAT(A(28),F(4,0));  
  PUT SKIP EDIT('NUMBER OF WORDS',WORD_COUNT) (R(FORM_1));  
  PUT SKIP EDIT('NUMBER OF SYLLABLES',SYLL_COUNT) (R(FORM_1));  
  AVG_SENT = FLOAT(WORD_COUNT,7)/FLOAT(SENT_COUNT,7);  
  PUT SKIP EDIT('AVERAGE SENTENCE LENGTH',AVG_SENT) (R(FORM_2));  
  FORM_2: FORMAT(A(28),F(5,1));  
  AVG_SYLL = FLOAT(SYLL_COUNT,7)/FLOAT(WORD_COUNT,7);  
  PUT SKIP EDIT('AVERAGE SYLLABLES PER WORD',AVG_SYLL) (R(FORM_2));  
  FLESCH_INDEX = 206.835 - 1.015 * AVG_SENT - 84.6 * AVG_SYLL;  
  PUT SKIP EDIT('FLESCH INDEX',FLESCH_INDEX) (R(FORM_2));  
  IF FLESCH_INDEX >= -50 & FLESCH_INDEX < 50  
  THEN  
    GRADE_LEVEL = (140 - FLESCH_INDEX)/6.66;  
  ELSE  
    IF FLESCH_INDEX < 60  
    THEN  
      GRADE_LEVEL = (93 - FLESCH_INDEX)/3.33;  
    ELSE  
      IF FLESCH_INDEX < 70  
      THEN  
        GRADE_LEVEL = (110 - FLESCH_INDEX)/5.0;  
      ELSE  
        GRADE_LEVEL = (150 - FLESCH_INDEX)/10.0;  
  PUT SKIP EDIT('GRADE LEVEL EQUIVALENT',GRADE_LEVEL) (R(FORM_2));  
  END SUMMARY;  
END GMSTAR;
```

\$ ty report.dat

***** SAMPLE TEXT ANALYSIS *****

MOST PEOPLE WHO BANK WITH US ARE COURTEOUS AND FRIENDLY BUT OCCASIONALLY A CUSTOMER MAY BE RUDE, OVERBEARING, EVEN UNKIND AND DISCOURTEOUS. WHEN THIS SITUATION ARISES, WE CAN BEST HANDLE IT BY FIRST, KEEPING CALM. A CUSTOMER MAY APPEAR TO BE ANGRY AT THE BANK BUT ODDS ARE HE IS ANGRY AT SOMEONE ELSE, A MEMBER OF HIS FAMILY OR SOMEONE AT WORK OR THE CUSTOMER MAY FEEL ILL OR SLIGHTLY UPTIGHT, SO KEEP CALM AND COLLECTED. TRY TO FIND OUT WHAT THE PROBLEM IS. USUALLY THE PROBLEM IS VERY SMALL BUT HAS BEEN MAGNIFIED OUT OF PROPORTION BY THE CUSTOMER. DO NOT ARGUE OR ENGAGE IN A VERBAL BATTLE. REMEMBER, THE WORST THING WE CAN SAY TO AN IRATE CUSTOMER IS YOU'RE WRONG. SINCERELY APOLOGIZE FOR ANY INCONVENIENCE CAUSED. THAT'S RIGHT! APOLOGIZE EVEN IF THE CUSTOMER IS CLEARLY AT FAULT. REMEMBER OUR JOB IS TO HELP PEOPLE EVEN THOSE WHO FEEL NEGATIVE TOWARD US.

NUMBER OF SENTENCES	11
NUMBER OF WORDS	155
NUMBER OF SYLLABLES	234
AVERAGE SENTENCE LENGTH	14.1
AVERAGE SYLLABLES PER WORD	1.5
FLESCH INDEX	64.8
GRADE LEVEL EQUIVALENT	9.0

REFERENCES

1. Flesch, Rudolph, "The Art of Readable Writing", Macmillan Publishing, 1949
2. General Motors Service Research Group, "GM-STAR" Computer Program, June 6, 1973
3. Consumer Reports, "A Bold Step Against too Much Fine Print", November, 1973