#THE SIMDPLE DISCOURSE ATTRIBUTES ANALYSIS PROGRAM (SIMDAAP09) OPERATING INSTRUCTIONS
##Bernard Maskit

This document contains an overview of the current version (SimDAAP09) of the one speaker (simple) Discourse Attributes Analysis Program (DAAP) using Word Count as the independent variable, including transcribing rules, an outline of the SimDAAP procedures and the SimDAAP output. An overview of the DAAP software may be found at:
http://www.thereferentialprocess.org/the-discourse-attributes-analysis-program-daap

DAAP09 has two distinct tracks: DAAP09.6 for analyzing texts involving multiple speakers, and SimDAAP09 for analyzing texts with only one speaker or writer. These instructions are for SimDAAP09 for single speakers; there is a separate set of instructions for DAAP09.6.

Before running presimDAAP09, ensure that the text files to be processed have the correct syntax, as given below.
SimDAAP09 requires that you first run presimDAAP09; after each run, you need to check the LogFile (ProjectLOG.txt) and terminal window for transcription, syntax and other errors. These must be corrected before simDAAP09 itself is run.
In order to run presimDAAP09, the following files must be in your DAAP folder: ListL2, ListL8, ListL11, ListW2, TheDic. These files are needed for the presimDAAP09 function of disambiguation.

##CONTENTS

##A. STARTING PRESIMDAAP09 ON A MAC OSX PLATFORM

To run presimDAAP09, open a terminal window (applications -> Utilities -> Terminal). Inside the terminal window, use the Unix Change Directory (cd) command to navigate to the DAAP folder.
To see the list of folders in the current directory, type: ls <RETURN>.
To move to the folder DAAP in the current directory, type: cd DAAP <RETURN>.
To move up one level, type: cd .. <RETURN>.
Once you have navigated to the DAAP folder, type: perl presimDAAP09.pl <RETURN>.
The program will prompt you for the name of the folder containing the files to be processed.
Type the name of that folder followed by <RETURN>.

##B: STARTING PRESIMDAAP09 ON A WINDOWS PLATFORM

a: EASY ANSWER: Use the Windows navigation system to navigate to the DAAP folder. Double click on presimDAAP09.pl. (This will work only if perl has already been installed.) A command window will open with the request that you type the name of the projects folder to be analyzed. Type the name of the projects folder and hit <RETURN>.
CAUTION: THE COMMAND WINDOW WILL CLOSE, AND YOU WILL NOT SEE THE LIST OF ERRORS. YOU NEED TO OPEN THE FILE: DAAP\PROJECT\DATA\PROJECTLOG.TXT TO SEE THE LIST OF ERRORS.

b: BETTER ANSWER: Open a command window and use the DOS change directory (cd) command to navigate to the DAAP folder. At the prompt, type presimDAAP09.pl <RETURN>. You will be prompted to type the name of the subfolder (here called Project) containing the text files to be processed. In addition to the log file mentioned above, a list of errors will appear on the screen.
Note that the DOS command for the list of folders in the current directory is dir; the cd command works approximately the same way in both DOS and UNIX.

##C: SIMDAAP09 OPERATING INSTRUCTIONS

1. In order to run simDAAP09, the following files must be in the DAAP folder: TheDic, ZBad.txt, ZChange.txt, ZGood.txt. (There must be files with these names; they do not need to have any content.) Except for ZChange,txt, these files are all lists of words. ZChange.txt is a list of pairs of words separated by a comma (see below for explanation).

2. The DAAP folder must contain two subfolders, one labeled "Dics", and the other labeled "DATA". The Dics subfolder should contain whatever dictionaries, as above, are needed for the project. Both presimDAAP09 and simDAAP09 place all their output files in this DATA subfolder.

3. Before running simDAAP09, ensure that presimDAAP09 runs successfully with no errors.

4. simDAAP09 produces the file ProjectLOGF.txt in the DATA subfolder. This file contains a list of the text files processed, and, for each file, a list of transcription and other errors. These errors must be corrected and then both presimDAAP09 and simDAAP09 must be rerun.

##D: TRANSCRIBING RULES I: GENERAL INFORMATION

File Names. If this transcription is part of a larger project, such as several sessions or interviews or subject responses in an experiment, the transcript for each such session or interview or subject must be placed in a separate file, where the file name reflects the session or interview or subject number. These files are then all to be placed in a single Project folder.

The text file must be in text (.txt) format. The file name must be of the form FileName.txt, where FileName contains only upper and lower case letters and numbers; no dots, dashes or commas.

Word processing format. If at all possible use a fairly low level word processor, such as TextEdit or WordPad. Many word processors automatically change single and double quotation marks into �smart� quotes; if at all possible, turn these off, so that you get "unsmart" quotes. (In MS Word, uncheck the appropriate boxes in Tools, AutoCorrect.) Do not use any of the typography or formatting tools of your word processor, such as bold, italics, justification, etc. Make sure you write or save (Save As) your final version as a text (.txt) file. Some word processors break words and insert hyphens. Make sure your word processor does not do this.

##E: TRANSCRIBING RULES II: NECESSARY SYNTAX

Parentheses. simDAAP09 does not process items within square brackets. Any identifying material at the beginning or end of each file must be placed within square brackets. At the beginning of each run, simDAAP09 asks if you wish it to also ignore material within round brackets. (There are different reasons why one may or may not wish such material to be processed.)

Please do not use either angle brackets (<>) or curly brackets ({}) for any purpose.

Backslashes. The backslash may only be used as indicated below. It must always be the first character of the line it is on.

Hyphens and dashes. See below.

##F: TRANSCRIBING RULES III: OPTIONAL SYNTAX

Top matter. If necessary, the transcript begins with a confidentiality and/or ownership statement provided by the project director. Place this confidentiality statement in square brackets at the top of the transcript.
Follow this with a statement, also in square brackets, telling: the transcriber�s identifier (i.e., name or initials, as directed by the project director), date of transcription, and name of study. Write the name or number of the subject on the next line. The file name should also identify the subject.

Optional function - Classification indicators. You can choose any number of classification terms depending on the organization of your study. For example, if your transcript is an interview with different overall categories of subject matter, and distinct questions within each category, your classification terms might be: 'time', 'person' and 'question'. If, for example, you questions are concerned with incidents at different times (far past, near past and future), different people (father, mother, spouse), you have a set of questions for each of these times and people, and you start with questions about mother in the far past, you might place the following before the transcription of the first speaker:

```
\t time:FarPast

\t person:Mother

\t question:1
```

Note that each of the above lines contains a space after \t, but no other spaces. Note also that the name of the classification indicator (time, person, etc.) and the name of the actualization (FarPast, Mother, 1) contain only upper and lower case letters and numbers.
If your transcript contains no other classification indicator, then you can proceed to the first question. If you will have other classification indicators in the transcript, they must all be identified as above before the first word of the text to be processed.
After you have completed the first question, if you move on to question 2, then you type on a line by itself:

```
\t question:2
```

However, if the next question asks question 1 about father, you next type on a line by itself:

```
\t person:Father
```

There is no need to repeat that we are still in the far past or that we are still concerned with question 1.

The point of these classification indicators is that they can be used to aggregate data. simDAAP09 prompts the operator for aggregation classifications. For example, if you want to see data for all responses concerning the mother, tell the operator to aggregate the data for time and question; that is, in response to the aggregation question, the operator types "time:question". If you have a batch of files in a folder to be simultaneously run, they must all use the same classification indicators, and, for each run, they will all be aggregated the same way.
All the classifications to be used in your study must be included in the first set of classification indicators in each file. You cannot add a new classification after the beginning of the text to be processed; and you must spell the classification term exactly the same way each time you use it (DAAP is case sensitive, spelling includes the use of capital letters).
The classification indicators, and the names of the actualizations of these indicators (e.g., FarPast, Mother, 1) must be made up of ordinary letters, either upper or lower case, and numbers; no spaces or other keyboard characters are allowed; also no accented letters are allowed. Again, the program is case sensitive, 'T' and 't' are different letters.

Confidentiality. For confidential material, the transcriber must consult with the research project director concerning the use of disguises. All proper names of persons, places and animals must be changed; they are usually changed into other names of persons, places and animals, of the same form, but some project directors prefer other codes. This replacement must be done by the transcriber; the software will not do it. These changes must be listed in a code book, and must not appear in any form in the transcript. The form and safety of the code book is the responsibility of the project director. NOTE: the "words": 'a', 'd', 'i', 'm', 's' and 't', are actual words according to DAAP, and must not be used as disguises.

Sounds other than spoken words. Events, or sounds other than words, should be noted in square brackets, as in [laughs], [coughs], or [telephone rings], etc.

##G: TRANSCRIBING RULES IV: RULES FOR WORDS

NOTE: These rules are for application of the Referential Process dictionaries, including WRAD, which must be separately downloaded. The rules concerning hyphens (dashes) are necessary for both presimDAAP09 and simDAAP09 to function correctly.
Rules for words. The following are intended to standardize the decisions that the transcriber will need to make. NOTE: These rules do not apply to items within square brackets, as simDAAP09 ignores all such items.

a) A word or phrase that is in the dictionary should be transcribed as written in the dictionary.

b) If there is a choice between a hyphenated or unhyphenated form, such as �goodbye� or �good-bye�, choose either one; DAAP automatically changes it into the unhyphenated form.

c) If the dictionary offers one word or two, such as �chickenpox� or �chicken pox�, choose one word; i.e., �chickenpox�.

d) If the item sounds as if it ought to be one word, but is not in the dictionary, such as �bookturner�, and there is no such item in the dictionary, and both parts are words in the dictionary, write it as two words; i.e., �book turner�.

e) If the item as spoken appears as one word, such as �nonbelief� and there is no such item in the dictionary, and the two separate parts are not words in the dictionary (�non� is not a word in the dictionary; it is listed as a prefix), then write this as one word, that is, �nonbelief.�

f) There are a few doubly hyphenated words, such as "mother-in-law". Please type such words as unhyphenated; that is, "motherinlaw".

g) Hyphens. Use hyphens only for compound words as above, and for marking incomplete words. Denote an incomplete word by ending it with exactly one hyphen followed by a space. For example, if the speaker stutters and says: "f f fail"; this should be transcribed as "f- f- fail"; note the spaces after the hyphens. If the speaker starts a word, hesitates, and then either completes the word or says another word, type the first partial word with a hyphen at the end, followed by a space. For example if the speaker says "some", then hesitates, then says "somewhat", transcribe it as "some- somewhat". The reason for this rule is that DAAP counts incomplete words as disfluencies. However, if the speaker says �I I I don�t know what to say�; these are not incomplete words, and this sentence should be transcribed as shown (DAAP also treats repeated words as disfluencies).
Do not use hyphens for purposes other than incomplete words, and binding parts of words, as in non-judgmental. Some word processors, including MS Word, sometimes change hyphens used in other contexts into em-dashes or en-dashes, which can cause difficulties.

h) Unclear words. These are noted in square brackets; if the speaker says "the" followed by one or more unclear words, type "the [unclear]". It is not necessary to try to preserve the number of unclear words.

i) Misspoken words. If the speaker misspeaks, or if you hear the speaker as misspeaking, and there is no doubt as to the correct meaning, type the correct word. For example, if the speaker says something that sounds like, �I want to Philadelphia yesterday, and walked on Market Street�, this is clearly a misspeaking, and the correct word is �went�, rather than �want�, so the transcription should read, �I went [want] to �� (one might need to code either "Philadelphia" or "Market Street").

j) Apostrophes. Use apostrophes as usual for contractions, such as "don't", "can't", "I'd", and for possessives.

k) Filled pauses. Sounds that have no meaning, such as �um�, 'hm', 'ah', etc., should be written as they sound, with the following exception. An elongated "hm" should be typed as "mmm" or "hmm", but not more than 3 successive letters 'm' should be used.

l) Numbers. Numbers should be written as numbers, but without commas or periods; that is, if the speaker says "two hundred thousand", you can type 200000. If the speaker says "four and a half", type, "4 and a half". For time, type for example, �8:45".

m) Amounts. Amounts, such as dollars and percents, should be transcribed as spoken; i.e., '100 dollars', or '10 percent'.

n) Disfluencies. The usual disfluencies are: 'well', 'like', you know', 'I mean' and 'kind of'. PreSimDAAP09 disambiguates the words 'kind', 'know', 'like', 'mean' and 'well', so that these have different forms for different purposes. Filled pauses, usually transcribed as 'hm', 'um', 'mm' or uhm', are also changed by preSIMDAAP09 into the standard 'mm', which is in the disfluency dictionary. (PreSIMDAAP09 does not have the facility to separately run its disambiguation and error checking functions.) Incomplete words, repeated words and repeated pairs of words are also marked by SimDAAP09 as disfluencies. You can mark a word as a disfluency by typing DX as the last two letters of the word. For example, the word 'oh' is sometimes used as a disfluency; in that case, it should be transcribed as 'ohDX'.

o) Slang. presimDAAP09 processes a few contractions without apostrophes, these include "gonna", "wanna" and "gotta". For all others, and for all slang expressions, please type the corresponding English words. For example, if the speaker says "yep", please type "yes"; if the speaker says "gotcha", please type "got you".
Punctuation Marks. Use punctuation marks, such as commas, periods, semicolons and question marks as in customary usage. presimDAAP09 uses these to decide if, for example, the word 'like' is being used as a disfluency. Do not use exclamation marks to denote emphasis.
Pauses. If there is need to keep track of time, there are separate instructions for inserting time markers. If the project manager wishes you to keep track of pauses, you may use slashes (/), to indicate pauses of up to five seconds (one slash for each second). Do not use longer dashes (em-dashes or en-dashes) or three dots (�) for this purpose. You should leave a space both before and after each of these slashes.
Unusual Symbols. Some word processors on occasion change hyphens to em-dashes or en-dashes. Please use hyphens only as listed above. Some word processors change ... into a special ellipses symbol; to avoid this, do not type three dots in a row.

##H: CONSTRUCTING NEW DICTIONARIES

Constructing Your Own Unweighted Dictionary An unweighted dictionary is simply a list of words, in alphabetical order, with each word on its own line, written in text format, where the file containing the dictionary has no suffix; for example, the name of the file containing the Disfluency dictionary is DF. If you are constructing your own dictionary, please do not use any of the names already in use: AffSen, AN, AND, ANF, ANH, ANP, AP, APA, APH, APL, APS, APW, AZ, CRF, DF, future, Logic, LOL, Neg, past, Present, R, SAS, SenS. SimDAAP09 requires that the names of all weighted dictionaries come alphabetically after the names of all unweighted dictionaries. Due to this, we require that the name of any unweighted dictionary start with one of the letters: A ... S.)

Constructing Your Own Weighted Dictionary This is a list of words, in alphabetical order where each word is followed by a space, and then a number between -1 and +1 (-1 and +1 are also allowed). Again, each word, with its number is on its own line. The weights in the dictionary are assumed to have meaning referring to a linguistic or psychological construct, where zero is the neutral value, positive weights are assigned to words connoting positive quality, and negative weights are assigned to words with a negative connotation. A weighted dictionary must be in text format in a file whose name has the suffix .Wt. For example, the WRAD file is named WRAD.Wt. simDAAP09 requires that the names of all weighted dictionaries come alphabetically after the names of all unweighted dictionaries. Due to this, we require that the name of any weighted dictionary start with one of the letters: T, U, V, W, X, Y.)

Constructing Your Own Weighted Dictionary Using Lemmas (Z-Dictionary) The initial letter Z is reserved for weighted dictionaries that use word stems (lemmas). The name of each such dictionary starts with the initial letter Z, and has the suffix .ZWt. These dictionaries have the same format as other weighted dictionaries; each line contains a word or word stem (lemma), followed by a space, followed by a weight lying between -1 and +1.
The current distribution of the Referential Process dictionaries does not include any Z-dictionaries.

##I: presimDAAP09 and simDAAP09 OUTPUT FILES

In order to explain the structure of the output files, we assume that the there are some

number of distinct texts, with file names: Text1.txt, Text2.txt, etc. in the folder labeled "Project" that simDAAP09 is processing. This folder contains a subfolder, Dics, that contains the weighted and unweighted dictionaries to be used (again, in alphabetical order, the names of the unweighted dictionaries precedes the names of the weighted dictionaries; the names of the weighted dictionaries consist only of alphanumeric characters and no dots (periods); the names of the weighted dictionaries end with .Wt) There is also another subfolder, DATA, that contains the output data, as given below.
A few of the output files are produced by presimDAAP09, and are marked as such. All the others are produced by simDAAP09.
LogFiles Before the final run, check the LogFiles (ProjectLOG,txt and ProjectLOGF.txt) for errors in the text. These should be corrected in the original text files, and both PresimDAAP09, and simDAAP09 rerun.

###OUTPUT FILES FOR EACH TEXT FILE

a. There is a separate **Marked Text** for each file (Text1MTT.txt, Text2MTT.txt, etc.). Each of these files reproduces the material in the original text, and adds some markers. There is a marker of the form [a] every 20 counted words (words in square brackets are not counted), where 'a' is the current word count starting at the beginning of the text. There is also some information at the end of the marked text concerning the total number of words. This file is produced by presimDAAP09.

b.There is a separate **Smooth File** for each text (Text1SMT.csv, Text2SMT.csv, etc.). This file is a comma-separated-value (csv) file that can be read by any spreadsheet, such as Excel, Lotus 123, Numbers (Mac), Calc (Open Office), etc. The file contains the data needed to make a chart of the smoothed dictionary density functions. The top line contains the names of the variables. After that, there is one line for each word of the text. The columns are ordered alphabetically by dictionary.

c. There is a separate **Word File** for each text (Text1RAW.csv, Text2RAW.csv, etc.). This file contains a list of all the (counted) words in the file; for each word, the file shows the dictionary values assigned to the word. This file is primarily for use by the DAAP operator in error checking.

###PROJECT OUTPUT FILES

a. There is an overall **Coverage File** (ProjectGLB.csv). This file contains information for each file in the folder that is independent of category marker: the number of words, the number of turns of speech (these are determined by the category markers), and then, for each dictionary, the coverage, which is the percent of words in the text that match the dictionary, and the number of dictionary matches. (Some of this is redundant: the coverage equals the number of dictionary matches divided by the total number of words.)

b. There is an overall **Turn File** (ProjectTRN.csv). This file contains a separate row for each change in category markers (these are labeled as turns of speech). The columns give the name of the particular text file; the turn number (this starts again at 1 for each file); the actualization of each category (there is a separate column for each category); the number of words in this turn of speech; the number of the word at which this turn of speech starts; the mean score for each unweighted dictionary; the mean score, the mean high score, and the high score proportion for each weighted dictionary; and (if requested) the covariations between each pair of dictionaries.

The *High Score Proportion* for a weighted dictionary is the proportion of words for which the smoothed dictionary density function is above the neutral value of .5. (For psychological or linguistic variables, negative quantities are sometimes difficult to interpret. Each weighted dictionary uses weights between -1 and +1 with 0 as neutral value, so words not in the dictionary are given dictionary score of 0, as they are with unweighted dictionaries. SIMDAAP09 linearly transforms these weights to lie between 0 and 1 with .5 as the neutral value.)

The *Mean High Score* for a weighted dictionary is the average value of the amount by which the smoothed dictionary density function for that dictionary is above the neutral value of .5. That is, we look only at the words for which the smoothed dictionary density function is above .5, and we take the average value, for those words only, of the difference between this smoothed dictionary function and .5.

The *Covariation* between a pair of dictionaries is a measure that is computationally related to the (Pearson) correlation coefficient, but does not have the same statistical meaning. The weighted dictionaries all have a natural neutral value of .5, the unweighted dictionaries are each given their mean over the entire text as a neutral value. The covariation is a measure of the extent to which the smoothed dictionary densities are simultaneously above, and simultaneously below, their neutral values. (NOTE: IT IS EXPECTED THAT THE NEUTRAL VALUE FOR UNWEIGHTED DICTIONARIES WILL BE SET AT 0 IN FUTURE DAAP RELEASES.)

CAUTION: The covariations are generally unreliable for texts with relatively few words, or for texts consisting in large part of segments consisting of relatively few words. They are generally reliable for segments containing at least 25 words.
There is an overall Aggregation File (ProjectAG2.csv). This file might contain fewer rows than the Turn File, in that some of the data has been aggregated. For example, if there are two categories, call them 'type' and 'time', and the operator has responded to the prompt calling for the data from 'time' to be aggregated, then there is one row for each actualization of 'type' for each file; all the data concerning 'time' has been aggregated. The columns in this file are the same as in the turn file.

c. There is an overall **Main Aggregation File** (ProjectAG0.csv). There is a separate row for each text file. The data from all the categories are aggregated. The columns are the same as the Turn File.

d. There is an overall **New Left-Over-List File** (ProjectNewLOL.txt). This file contains a list of the words (types) in the data files that are not in any of the dictionaries being used and are not in the TheDic file. This list is for the entire project.

e. There is a Type-Token-Dictionary Match file (ProjectTTGlob.csv) in spreadsheet format showing all types, tokens and their dictionary matches for all files in the Project.

f. OUTPUT FILES FOR Z-DICTIONARIES. If you are using one or more Z-dictionaries (dictionaries that use word stems or lemmas), there are two separate ZMat output files, (ProjectZMat.txt and ZGoodQ). These files are designed for use with the ZGood, ZBad and ZChange files. The ZGoodQ file contains a list of words (types) that partially match a lemma in a Z-Dictionary; that is, the entire word does not match any word in the Z-Dictionary, but the first few characters do, and the word in the file does not appear in any of the files: ZGood, ZBad, or ZChange, (for example, "got" might be a word in one of the files, but not in a particular Z-Dictionary, while "go" is in that Z-Dictionary.) In this case, the word in the file appears in the ZGoodQ file, while both the word, and the lemma it partially matches are in the ZMat file. (In this example, the ZMat file contains a line with "go,got" in it.) The point of these files is to question the particular match. The operator needs to put this word ("got" in the example) either in the ZBad or ZGood files, or in the case of this example in the ZChange file as "got get"), so that, at the next run, preDAAP09 will know how to handle this word with respect to Z-Dictionaries.

##J: ERROR MESSAGES

Note that some of these errors are reported by PRESIMDAAP.

ERROR 1. Backslash not at first character in line. These can usually be corrected by searching for: \ (space followed by backslash), and replacing with just backslash.

ERROR 2. Unmatched parentheses. DAAP has completed its processing of a text file and ended with an unmatched open parenthesis, given as either a round or square bracket. DAAP does not process words in parentheses, so the word count given in square brackets in the marked text (MTT file), which is produced by preDAAP09, has stopped at the open parenthesis.

ERROR 3. Improper parentheses. A close parenthesis has occurred (either round or square bracket), with no open parenthesis preceding it. The LOGFile (either ProjectLog.txt or ProjectLogF.txt) repeats the line in which this error occurs.

ERROR 4. Unusual Word. DAAP only recognizes the usual printable characters; it does not recognize UTF characters. Many word processors use "smart quotes" or other special characters that are not part of the ASCII character set; this sometimes causes problems, especially if the text file was written on a different platform from the one that DAAP is running on. The unusual word is reported both on screen and in the ProjectLogF.txt file. A blank word may usually be safely ignored.

ERROR 6. Split Problem. PREDAAP09 has encountered an unusual problem. The LogFile, ProjectLog.txt shows the place at which the unusual problem has occurred; there is probably an unusual character here.

ERROR 7 (Fatal). Improper File Name. The file name contains an underscore or contains more than one period (dot). (However, note that PREDAAP09 starts with textfile.txt, and outputs textfile.c.txt.

ERROR 8. Improper Backslash t line. Category markers must be of the form:
\t Category:Actualization
Note that that the backslash is the first character in the line, but it may be preceded by a space. There is a space after the t. There may be spaces either before or after the colon. Both the category name and the name of the actualization contain ONLY upper and lower case letters and numbers; they may not contain slashes, dashes, dots, commas, spaces, etc. For each category, there must be a line at the beginning of the file, before the first word to be processed, with this category marker.

ERROR 10. Backslash Problem. PresimDAAP has encountered an unusual problem. Ensure that the \t lines are appropriately placed.
NOTE: For each text file PresimDAAP09 produces a file called TextFile.a.txt in the Projects Folder. On runs past the first, it reports an ERROR 14 for this TextFile.a.txt. This ERROR 14 message may be safely ignored.

ERROR 15. Undeclared Category. A new category marker has been introduced. The error message includes the word number at which this occurs. All category markers must be declared before the first line of text to be processed.

##FURTHER READING
For a detailed discussion of the mathematics that underlie the DAAP software please see:
Maskit, Bernard (2014): DAAPMath. figshare. http://dx.doi.org/10.6084/m9.figshare.928469
SIMDAAP09 Operating Instructions