

Praatを用いた知覚実験の方法

渡辺和希 (筑波大学大学院)

2012年4月22日 於: 筑波大学

Praatとは

アムステルダム大学の
Paul Boersma 氏 David Weenink氏が
開発した **無料**の音声分析ソフト

音声分析からアノテーション、そして最近では文章の読み上げまでいろいろできしまうすごいソフト!

<http://www.fon.hum.uva.nl/praat/>

ExperimentMFCとは

- Praat上で動く実験プログラムのことを
ExperimentMFCという。
- 詳しい内容は公式ページ参照 (英語)
<http://www.fon.hum.uva.nl/praat/manual/ExperimentMFC.html>

今回はその作り方を簡単に紹介します。

まずは簡単な[Demo](#)をご覧ください。

.....と、このように実験を組むことができます。

実験するだけでなく.....

- Reaction Timeが計測可能
- 保存形式によってそのままエクセルで集計可能

実験後の処理も簡単にできる。

非常に便利なプログラムです!

注意点

簡単なプログラムを組む感じなので...

- 大文字と小文字を間違えると動かない。
- 関係ないところにスペースがあったり、余計な文字が入っていたり、必要な文字がなかったりすると、動かない

といったように作成過程でエラーが頻発することもしばしば

プログラミング言語のような難解なものではないので、
マニュアルのテンプレートを
変更するだけで簡単に作成できます。

作成方法

- 基本的にtxtファイルでプログラムを組み、Praatiに読み込ませるだけです。
- 用意するもの
(音声ファイル、テキストエディタ、Praat)
他にソフトなどは一切いりません！

※PCをお持ちの方で今回実際に作成する方は、音声ファイルが必要となります。
適当で構わないので、Praat上で音声ファイルを3つほど録音してください。

土台の作成

まず、txtファイルに
"ooTextFile"
"ExperimentMFC 5""実験名"

と打ち込みます。
"実験名"は任意の実験名を入れます。

※文字コードはUTF-8で保存(文字化け防止)

実験の作成

それに続けて—

stimuliAreSounds? <yes>

と入力します。

これは刺激が音がそうでないかを示しています。
今回は知覚実験なので<yes>と入れます。

フォルダの指定

次に刺激音のフォルダの指定をします。
txtファイルと同じディレクトリのフォルダを指定します。

stimulusFileNameHead = "test/"

刺激音が置いてあるフォルダを指定します。

stimulusFileNameTail = ".wav"

刺激音の拡張子を指定します。

刺激音前後の音の指定

stimulusCarrierBefore = ""
stimulusCarrierAfter = ""

刺激音の直前or直後に音を流したいときに流したい音声ファイルを指定します。

- 例えば、開始の合図として純音を入れたい、とか説明を入れたいといったときに使います。

今回は使わないので何もいれません。

※用いる場合、刺激直後に提示される音声ファイルを拡張子つけずに指定します。

まとめ①

"ooTextFile"
"ExperimentMFC 5"
stimuliAreSounds? <yes>
stimulusFileNameHead = "test/"
stimulusFileNameTail = ".wav"
stimulusCarrierBefore = ""
stimulusCarrierAfter = ""

txtファイルはこのようになっていればokです。
このように続けて打ち込んでいきます。

Silenceの指定

stimulusInitialSilenceDuration = 0.5 seconds

刺激発出前の無音区間を指定します。
今回は0.5秒とします。

stimulusMedialSilenceDuration = 0

複数の刺激を連続で出すとき
その間の無音区間を指定します。
今回は刺激は1つなので、0 にします。

刺激の設定

numberOfDifferentStimuli = 3

"1" ""

"2" ""

"3" ""

刺激の数を指定し、音声のファイル名を拡張子無で指定します。
ファイル名のあと " " は 発出時のテキストを指定できます。

numberOfReplicationsPerStimulus = 1

一つの刺激音を何回流すかを指定します。

breakAfterEvery = 1

何回ごとに休止を置くかを指定します。

ランダム方法の設定

randomize = <*****>

<CyclicNonRandom>

ABCDEFGHIJ.....のように並べられた順で発する。

<WithReplacement>

ランダム。各刺激の出現回数と同じとは限らない。

<PermuteAll>

ランダム。各刺激の出現回数と同じ。

<PermuteBalancedNoDoublets>

ランダム。各刺激の出現回数と同じで同じ刺激が連続しない

今回は<PermuteBalancedNoDoublets>を指定します。

まとめ②

- まとめ①のあとに追加

stimulusInitialSilenceDuration = 0.5 seconds

stimulusMedialSilenceDuration = 0

numberOfDifferentStimuli = 3

"1" "" "2" "" "3" ""

numberOfReplicationsPerStimulus = 1

breakAfterEvery = 1

randomize = <PermuteBalancedNoDoublets>

説明文の作成

startText = " "
実験開始時の説明文を書きます。

runText = " "
刺激音が流れる場面=回答選択場面の説明文を書きます。

pauseText = " "
休止中の説明文を書きます。

endText = " "
実験終了時時の説明文を書きます。

※マウススクロールはできません

まとめ③

わかりやすくするために、
簡単に説明文を入れてみます。

startText = "開始です"

runText = "選択してください"

pauseText = "休憩です"

endText = "終了です"

回答画面の作成

- 次に回答画面の作成をします。

刺激音の聞き直しができるかどうか設定します。

`maximumNumberOfReplays = 2`

この場合2回まで聞き直しができます。
聞き直しをさせない場合は 0 にします。

ボタンの設定

`replayButton = 0.80 0.90 0.40 0.50 "再生" "r"`

聞き直しボタンを作成します。
小数点で並んだ数字は回答ボタンの画面上での位置を指定します。
ボタンの 左 右 下 上 の位置を示します。
つまり、左辺が画面上の左から0.8の場所、右辺が左から0.9の位置、
上辺が上から0.4の位置、下辺が上から0.5の位置ということです。

画面の中心が0.5ですから、0.8 0.9というのは、画面の右側で、
0.4と0.5というのは中心あたりに位置していることになります。(詳細は口頭で図示します)

"再生"の部分は、文字列を入力し、聞き直しボタン上に表示します。
"r"は、クリックの代わりにキーボードのRキーでも聞き直すことができます。指定しない場合は何も入力しません。

※なお、このボタンの設定方法は他の回答ボタンとも共通です。

特殊ボタンの作成

`okButton = 0.80 0.90 0.40 0.50 "ok" ""`

okボタンを設定します。
設定しない場合、回答を選択すると、次の刺激に移行しますが
okボタンを設定すると、
okボタンを押すまで次の画面には移行しません。
okボタンを押すまで、回答の再選択が可能です。

`oopsButton = 0.80 0.90 0.10 0.20 "戻る" ""`

戻るボタンを設定します。
設定すると、直前の刺激の画面に戻ることができます。
つまり、前の刺激に戻って再選択ができるようになります。

まとめ③

- まとめ①②のあとにボタンの設定を追加します。

`maximumNumberOfReplays = 2`
`replayButton = 0.80 0.90 0.80 0.90 "再生" "r"`
`okButton = 0.80 0.90 0.40 0.50 "ok" ""`
`oopsButton = 0.80 0.90 0.10 0.20 "戻る" ""`

回答ボタンの設定

`responsesAreSounds? <no> "" "" "" "" 0 0`
回答が音声であるかどうかを設定します。
<yes>にすると、回答ボタンをクリックすると音声流れます。今回は<no>にします。

`numberOfDifferentResponses = 2`
回答数を選択します。今回は2つにします。

回答ボタンを作ります。先ほどのボタンと同じ設定方法でボタンを配置します。
`0.25 0.35 0.40 0.50 "YES" 20 "" "2"`
`0.45 0.55 0.40 0.50 "NO" 20 "" "2"`

`numberOfGoodnessCategories = 0`
Goodnessを計測する場合(とても良い・良い・普通・悪い・とても悪い)は上と同様に
設定します。上と同時に計測することができます。

まとめ④

まとめ③に追加してください。

`numberOfDifferentResponses = 2`
`0.25 0.35 0.40 0.50 "1" 10 "" "1"`
`0.45 0.55 0.40 0.50 "2" 10 "" "2"`
`numberOfGoodnessCategories = 0`

完成

- まとめ①～④のテキストをまとめて、ひとつのtxtファイルとして保存します。
- それをPraat上で開きます。
- 読み込まれたファイルを Run すると、実験が始まります。

実際に動かしてみる

エラーが出る場合

- ▶ 大文字・小文字が間違っている。
- ▶ 余計なところに文字・スペースなどがある。
- ▶ 必要なところに文字・記号が書かれていない。
- ▶ フォルダなどの指定先が間違っている

などなど、融通が利かないので、
細かなミスがないようにチェックしてください。

最後に実験結果の保存方法です

実験結果の保存方法

- 実験を最後まで進めて、「終了です」の表示がでる画面までいきます。

- 実験画面は消しても構わないが、念のためそのまま、Praat Objects > Extract results

するとResult MFCファイルができあがるので
Collect to Table→View & Edit で結果を確認できます。

結果ファイルは、Result MFCでもTableでも用途に応じて保存してください。
Tab区切り、comma区切りにするとエクセルでも読み込むことができます。

おわりに....

今回は簡単な実験でしたが、

複数の実験ファイルを埋め込んだり、
画像を使った実験も可能であったり、

カスタマイズすることでいろいろな実験が可能です。

複雑になればなるほど、当然エラーも起きやすくなる
ので、訓練が必要です....。