ALVES, Ubiratã Kickhöfel. Teoria da Otimidade Estocástica e Algoritmo de Aprendizagem Gradual: princípios de funcionamento e tutorial para simulação computacional. *ReVEL*, vol. 15, n. 28, 2017. [www.revel.inf.br]

TEORIA DA OTIMIDADE ESTOCÁSTICA E ALGORITMO DE APRENDIZAGEM GRADUAL: PRINCÍPIOS DE FUNCIONAMENTO E TUTORIAL PARA SIMULAÇÃO COMPUTACIONAL

Ubiratã Kickhöfel Alves[1]

ukalves@gmail.com

RESUMO: No presente trabalho, apresentamos um tutorial para pesquisadores interessados em utilizar o Algoritmo de Aprendizagem Gradual vinculado ao modelo da Teoria da Otimidade Estocástica (Boersma; Hayes, 2001). Tal modelo permite formalizar gramáticas a partir das quais emergem *outputs* tanto categóricos quanto variáveis, dando conta da formalização dos processos de aquisição de sistemas de língua materna (L1) e de língua estrangeira (L2). Este texto está organizado em duas partes: em um primeiro momento, apontamos os princípios de funcionamento que caracterizam o Algoritmo de Aprendizagem Gradual (GLA). Após isso, apresentamos um tutorial para a realização de simulações, com o algoritmo em questão, a partir do *software Praat* (Boersma; Weenink). Esperamos, com este texto, contribuir para uma maior disseminação do modelo, de modo que os pesquisadores possam realizar, de maneira eficiente e sem dificuldades, simulações dos processos desenvolvimentais das gramáticas de L1 e L2 por eles estudadas.
PALAVRAS-CHAVE: Teoria da Otimidade Estocástica; Algoritmo de Aprendizagem Gradual; simulações computacionais.

ABSTRACT: In this paper, we provide a tutorial for researchers interested in the Stochastic OT version of the Gradual Learning Algorithm (Boersma; Hayes, 2001). Stochastic OT allows the formalization of OT grammars resulting in variable output forms, making it possible for researchers to run computational simulations on First Language (L1) and Second Language (L2) developmental processes. This article is divided in two main parts. Firstly, we review the basic tenets of the Stochastic version of the Gradual Learning Algorithm. After this brief description, we provide some guidelines on how to run simulations with the Gradual Learning Algorithm on Praat (Boersma; Weenink). We hope this tutorial may contribute to the dissemination of Stochastic OT, making it easier for researchers to run learning simulations of L1 and L2 grammars.
KEYWORDS: Stochastic OT; Gradual Learning Algorithm; computational simulations.

INTRODUÇÃO

Considerando-se o contexto brasileiro de pesquisas sobre os processos de aquisição de linguagem (Alves, 2010; 2012; 2013; Quintanilha-Azevedo, 2011, 2016;

[1] Professor do Programa de Pós-Graduação em Letras da Universidade Federal do Rio Grande do Sul (UFRGS). Pesquisador do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Garcia, 2012; Schmitt; Alves, 2014; Alves; Lucena, 2014; Matzenauer *et al.*, 2015; Neuschrank *et al.*, 2015; Gutierres, 2016; Matzenauer; Quintanilha-Azevedo, 2016), encontramos um número cada vez maior de estudos que fazem uso de simulações computacionais, através do *Software Praat* (Boersma; Weenink)[2], com o Algoritmo de Aprendizagem Gradual (GLA), vinculado ao modelo da Teoria da Otimidade Estocástica (Boersma; Hayes, 2001).

Conforme podemos ver no próprio nome proposto, tal algoritmo, frente ao seu compromisso de expressar a gradualidade que caracteriza todo o processo de aquisição de linguagem, deverá ser capaz de formalizar, inerentemente ao modelo, a variação nas formas de *output*. É interessante mencionar, nesse sentido, que os *outputs* variáveis resultantes do algoritmo podem corresponder não somente a estágios intermediários do processo desenvolvimental, em direção a uma gramática final que resulte em *outputs* categóricos. De fato, tais *outputs* variáveis podem, também, ser resultado de casos de aquisição plena de um sistema-alvo adulto, caracterizado pela variação. Dado o mecanismo utilizado pelo Algoritmo para dar conta desses "dois tipos" de variação, podemos dizer que o GLA se mostra capaz de expressar uma forte relação, em termos de formalização, entre as áreas de aquisição e variação da linguagem.

O presente trabalho, portanto, tem como objetivo principal apresentar um tutorial para a realização, a partir do *Software Praat* (Boersma; Weenink), de simulações computacionais com o GLA. Para isso, o presente texto será organizado em dois momentos principais. Primeiramente, explicaremos o princípio de funcionamento e as premissas básicas de operação do algoritmo. Após esta explanação inicial, apresentaremos instruções para a realização das simulações computacionais com o GLA, a partir da simulação de uma gramática categórica e, posteriormente, de um sistema que resulte em *outputs* variáveis. Esperamos que esta apresentação venha a motivar um número ainda maior de pesquisadores a adotarem o referido algoritmo e a realizarem simulações de processos de aquisição de hierarquias de restrições à luz da OT Estocástica.

[2] Cabe mencionar que simulações computacionais com o GLA podem, também, ser realizadas por meio do programa *OT-Soft* (Hayes; Tesar; Zuraw). Para simulações com o GLA através do *OT-Soft*, veja-se Ferreira-Gonçalves (2010).

2. O ALGORITMO DE APRENDIZAGEM GRADUAL: PRINCÍPIO DE FUNCIONAMENTO

Ao falarmos em GLA, é preciso deixar claro que não estamos mais operando com uma concepção *Standard* de OT. Na OT Estocástica (Boersma; Hayes, 2001), as restrições recebem valores numéricos, para atuarem ao longo de uma escala. Cada vez em que há avaliação de candidatos, tais valores são convertidos em um ranqueamento correspondente. A avaliação do candidato ótimo a partir do ranqueamento em questão (*ranking* esse resultante da conversão dos valores numéricos em hierarquia) segue as mesmas premissas de avaliação do modelo *Standard* da OT, a partir do qual a forma de saída é aquela que obedece às restrições mais altamente ranqueadas, independentemente do número de violações incorridas por tal candidato às restrições de *status* mais baixo na gramática. Vejamos tal noção de dominância estrita no *tableau* que segue:

(01)

	A	B
[*Output1*]	*!	
☞ [*Output2*]		***

De acordo com o *tableau* em (01), o candidato ótimo é [*Output2*], pois ele não viola o candidato mais altamente ranqueado na hierarquia, ainda que tenha desrespeitado a restrição mais baixa três vezes. Tanto na OT *Standard* quanto na Estocástica, a eliminação dos candidatos se dá a partir da restrição mais alta: assim, [*Output 1*] já é eliminado pela restrição A, o que fica claro pela marca de violação fatal "!", que simboliza a exclusão do candidato.

Conforme explicam Coetzee e Pater (2009), um aspecto importante da OT Estocástica diz respeito ao fato de que ela é acompanhada de uma teoria de aprendizagem, sendo vinculada a um algoritmo chamado de Algoritmo de Aprendizagem Gradual (GLA, do inglês *Gradual Learning Algorithm*). De acordo com os princípios de funcionamento do algoritmo em questão, o aprendiz recebe um mapeamento *input-output* de cada vez, e o estado corrente da gramática determina o *output* ótimo. Quando o *output* gerado difere dos dados da evidência positiva, o aprendizado acontece. O GLA atualiza o valor das restrições, de modo a subtrair um

valor *x*[3] dos valores das restrições que são mais violadas na forma correta do que no "erro" do aprendiz, além de adicionar um valor *x* a todas as restrições que são violadas no candidato com erro. A hierarquia de restrições, conforme vimos, é estabelecida em função dos valores a serem assumidos pelas restrições de tal escala numérica, valores esses que serão chamados de "valores centrais".

Ao nos referirmos ao GLA, julgamos fundamental ressaltar o seu caráter estocástico, de acordo com o qual o ranqueamento é afetado por um dado valor de ruído (*noise*) estatístico[4] a cada momento de avaliação de candidatos. O valor numérico das restrições, a ser promovido ou demovido pelo algoritmo, corresponde ao ponto central de uma faixa ou gama de valores probabilísticos que podem vir a ser assumidos pela restrição em questão, em um dado momento de produção. Em função do ruído, a cada momento de fala as restrições podem assumir um índice numérico distinto, caracterizado por Boersma e Hayes (2001) como "ponto de seleção". Em avaliações (momentos de produção linguística) sucessivas, restrições que apresentam tais valores centrais próximos um do outro poderão variar em termos de ranqueamento. Assim, é possível que, em um dado momento de produção, uma restrição A assuma um ponto de seleção mais alto do que B, enquanto que, em outros momentos, B assuma um valor de ponto de seleção mais alto do que A[5], ainda que, por exemplo, o valor central de A seja superior (ainda que bastante próximo) ao de B.

Conseguimos, desse modo, expressar a ocorrência de *outputs* variáveis em uma língua: a variação ocorre porque, em alguns momentos de conversão dos valores numéricos em *rankings*, as restrições com valores centrais próximos podem apresentar uma relação hierárquica X >> Y, enquanto que, em outros momentos, a relação Y >> X pode ocorrer. Isso somente acontece quando ambas as restrições apresentam valores centrais bastante próximos, de modo que o valor de ruído propicie que, em alguns momentos de avaliação, X apresente valor mais alto do que Y e, em outros momentos, Y consiga expressar valores mais altos do que X.

---

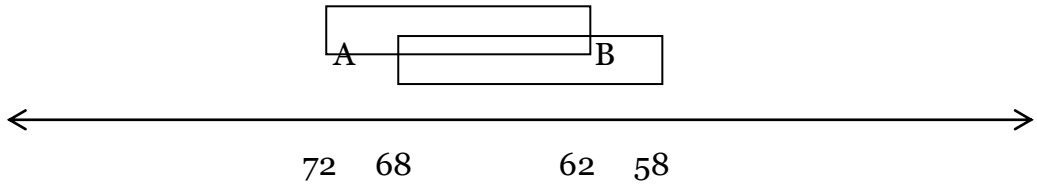
[3] O valor *x*, que corresponde à taxa de incremento/decremento do algoritmo, é definido, na simulação computacional, através do valor de *plasticidade*, valor esse que pode ser definido pelo pesquisador no *software Praat*.

[4] O valor de ruído *default* do algoritmo computacional, sugerido em Boersma e Hayes (2001), é 2.0.

[5] Boersma e Hayes (2001) chamam a atenção para o fato de que, ainda que o ponto de seleção possa compreender qualquer índice numérico dentro da faixa de valores, ele é mais provável de assumir pesos numéricos mais próximos do ponto central de tal gama de valores (ou, conforme chamam Boersma e Hayes, do *ranking value* - valor de ranqueamento, por nós chamado de "valor central"). Por exemplo, considerando-se uma restrição A, que apresenta valor central 67 e uma faixa que vai de 62 a 72, é mais provável que o ponto de seleção venha a assumir um índice numérico tal como 66, 67 ou 68, ao invés de 62 ou 72, ainda que esses últimos sejam também probabilisticamente possíveis.

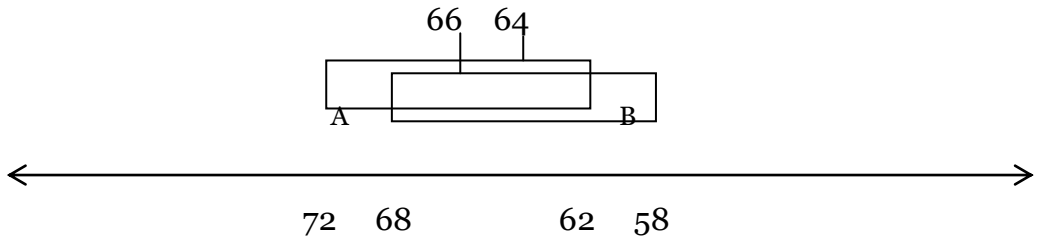
Vejamos o que foi acima afirmado de uma maneira mais aplicada. Consideremos que uma restrição A apresenta um valor central de 67, e uma restrição B, um valor central de 63. Uma vez que ambas as restrições apresentam valores centrais muito próximos, encontramos um cruzamento das faixas de valores possíveis de serem assumidos pelas restrições, o que podemos ver em (02):

(02)



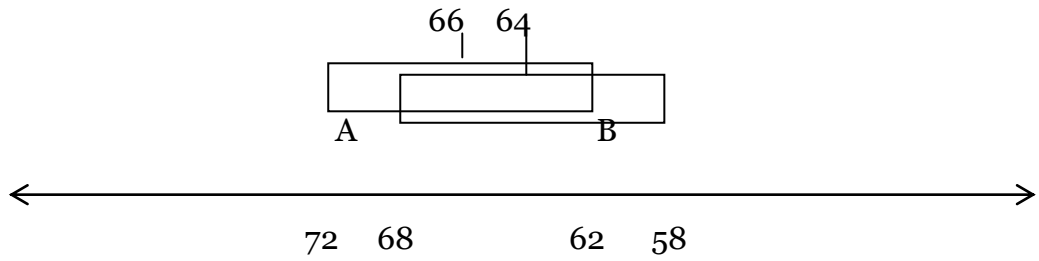
É este *overlap*, justamente, que deixa claro que as restrições se encontram suficientemente próximas para que, em determinados momentos de avaliação, a restrição B assuma um ponto de seleção mais alto do que o de A. Isso fica claro em (03), em que, no momento de produção linguística, a restrição A assume um ponto de seleção com valor de 64, enquanto que B assume um valor de ponto de seleção de 66.

(03)



Em outros momentos, entretanto, A assume um valor de ponto de seleção mais alto do que B. Em (04), é A que apresenta um ponto de seleção igual a 66, ao passo que B apresenta um valor de seleção igual a 64.

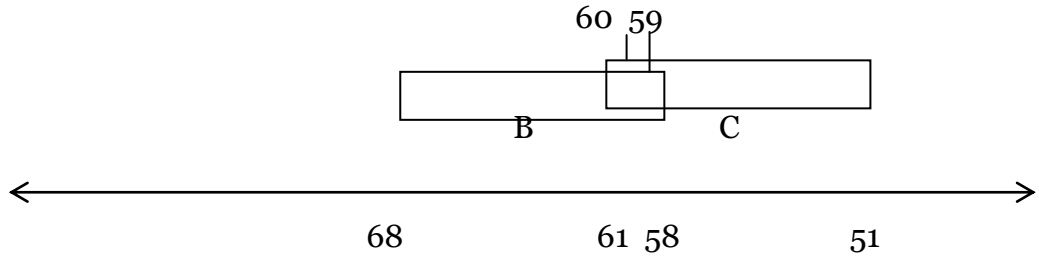
(04)



Uma vez que as relações hierárquicas entre as restrições são determinadas pelos valores de ponto de seleção que elas assumem, em (03) B>>A, enquanto que em (04) A>>B.

Vejamos agora um outro exemplo, em que apresentamos as restrições hipotéticas B e C, com valores centrais de 63 e 56, respectivamente. Conforme vemos em (05), há, também, um cruzamento entre as faixas de valores a serem assumidos por B e C. Em outras palavras, é possível que, em determinados momentos de produção, o ponto de seleção de C seja mais alto do que o de B, enquanto que a relação contrária seja possível em outros momentos de avaliação. É justificada, assim, a produção da forma $Output_x$, em alguns momentos de fala, e sob a forma $Output_y$ em outros, por um mesmo falante.

(05)

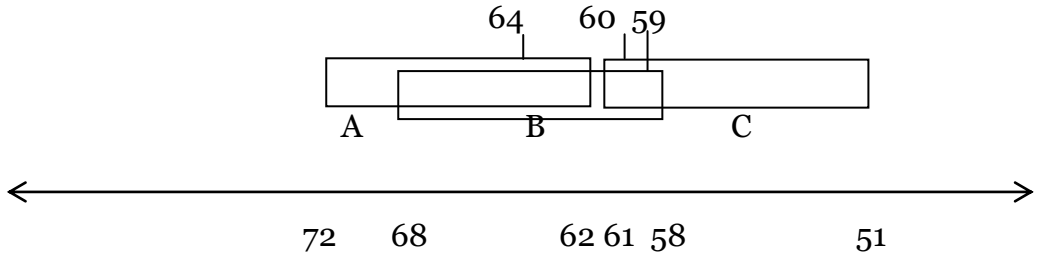


No momento de avaliação expresso em (05), C>>B. Precisamos atentar, entretanto, para o fato de que a faixa de valores em que há cruzamento entre B e C, em (05), é menor do que a faixa comum entre A e B, no exemplo apresentado em (04). Isso significa, em outras palavras, que as probabilidades de variação de *ranking* B>>C ~ C>>B, em função dos diferentes valores de ponto de seleção a serem assumidos por B e C, são menores do que a probabilidade de variação A>>B ~ B>>A.

Assim, ainda que possa haver *outputs* variáveis em função da supremacia ora de B, ora de C, o candidato ótimo advindo da relação B>>C constitui a forma variável predominante, pois grande é a chance de o ponto de seleção de uma das restrições assumir um valor fora da área de *overlap*, que é bastante curta.

Considerar que uma restrição domina categoricamente outra, de modo que a segunda nunca possa vir a apresentar um valor de ponto de seleção (e, consequentemente, um *status* hierárquico) mais alto do que a primeira, significa que, na escala contínua em questão, as duas restrições apresentam valores centrais bastante afastados, para que não haja um *overlap* em suas gamas de possíveis valores de ponto de seleção e, desse modo, não seja possível uma inversão hierárquica no momento da avaliação[6]. Tal fato, considerando-se as restrições já apresentadas, pode ser retratado através da relação hierárquica existente entre A e C, uma vez que seus valores centrais (67 e 56, respectivamente) estão afastados o suficiente para que as faixas de possíveis valores probabilísticos que podem ser assumidos pelos pontos de seleção dessas duas restrições não se cruzem.

(06)



Com base nas premissas básicas acima apresentadas, demonstraremos, nas seções que seguem, como realizar a implementação computacional do algoritmo, de modo que possamos observar e apontar os valores dos pontos centrais e de seleção de todas as restrições que compõem a gramática a ser aprendida.

[6] Em termos de simulação computacional, uma distância superior a 10 entre os valores centrais das restrições é suficiente para garantir que a variação não ocorra. Tessier (2007), entretanto, chama a atenção para o fato de que, em termos estatísticos, não podemos afirmar que uma possível reversão hierárquica dos valores de ponto de seleção das restrições que se encontram afastadas, em um dado momento de avaliação, seja impossível. Tal reversão é, na verdade, extremamente improvável.

3. O ALGORITMO DE APRENDIZAGEM GRADUAL: TUTORIAL PARA SIMULAÇÃO COMPUTACIONAL

No tutorial que apresentaremos a partir de agora, explicaremos o processo de simulação computacional de uma gramática categórica a partir de três etapas. Primeiramente, descreveremos o processo de preparação dos arquivos de .txt com os quais o *Software Praat* será alimentado. Em um segundo momento, explicaremos os comandos, a serem realizados no *Praat*, que garantirão a entrada de tais arquivos, bem como a simulação da aquisição da gramática-alvo. Na terceira etapa, descreveremos os métodos de observação do resultado final da gramática, fornecido pelo programa. Após a descrição desses três passos, explicaremos, na seção 3.4, os passos a serem tomados na simulação da aquisição de gramáticas a partir das quais são obtidos *outputs* variáveis.

3.1 PREPARAÇÃO DOS ARQUIVOS

Descritos os princípios de funcionamento do algoritmo, passemos a discutir, nesta seção, como realizar a simulação computacional do processo de aprendizagem com o GLA. Para isso, utilizaremos o *software* livre *Praat* (Boersma; Weenink). O *software* em questão, além de se mostrar como um mecanismo poderoso na simulação do processo de aquisição fonológica, pode ser usado não somente para a simulação do processo à luz da OT Estocástica, mas também nas simulações à luz do modelo da Gramática Harmônica (HG – Smolensky; Lengendre, 2016)[7], através de etapas procedimentais muito semelhantes às que vamos apresentar na presente seção[8].

Para um melhor entendimento do processo de simulação de aquisição de linguagem, pensemos no processo de aquisição de um sistema silábico de uma língua hipotética (mais especificamente, a posição de coda). Vamos pensar em um sistema

[7] Para maiores informações a respeito do modelo da Gramática Harmônica, vejam-se Alves (2010, 2012), Quintanilha-Azevedo (2011) e Quintanilha-Azevedo, Matzenauer & Alves (2013).

[8] O *Software Praat* tem, como *default,* a realização de simulações à luz da OT Estocástica. Portanto, para simulações com o GLA-OT, não é necessária nenhuma modificação adicional. Para simulações com a Gramática Harmônica, após alimentados os dois arquivos .txt (conforme as informações a serem fornecidas em breve), será necessário clicar no botão *Modify Behaviour – Set Decision Strategy*, e optar por uma das opções fornecidas pelo programa, de acordo com o modelo de gramática desejado pelo pesquisador (dentre as quais se encontra, inclusive, a opção *Optimality Theory* - o próprio GLA -, para os casos em que o pesquisador opte por realizar simulações à luz da OT Estocástica após ter "setado" o *Praat* para realizar simulações com a HG).

parecido com o português brasileiro, em que segmentos plosivos são proibidos em posição final (situação a partir da qual resulta a epêntese)[9].

Como em qualquer versão da OT, a escolha das restrições apropriadas é o segredo de uma análise bem-sucedida. O algoritmo, enquanto operação computacional, não faz o papel do analista. De fato, o algoritmo apenas desloca as restrições; portanto, se o pesquisador estiver equivocado no que diz respeito à proposição ou às marcas de violações das restrições, a implementação computacional não resolverá o problema. Assim, antes de mais nada, é necessário pensar no sistema de restrições com que operaremos. Tal escolha é um trabalho prévio à própria implementação computacional, e é um dos principais passos da análise. Para fins de exemplificação neste tutorial, utilizaremos, apenas, três restrições: a restrição de marcação *{stop}coda, que proíbe codas finais com plosivas, e as restrições de fidelidade Max e Dep, que proíbem *outputs* infiéis com apagamento e epêntese, respectivamente.

Pensemos, conforme já exposto, em um sistema em que um *input* tal como /top/ seja produzido categoricamente com epêntese ([topi]), em função da proibição a plosivas em coda. Para a simulação via *Praat*, é fundamental que o pesquisador prepare dois arquivos de texto (.txt), que podem ser elaborados no Bloco de Notas do *Microsoft Windows*: no primeiro arquivo, devemos informar o estágio inicial da gramática do aprendiz, ou seja, os valores numéricos iniciais de cada uma das restrições, além das marcas de violação, incorridas por cada um dos candidatos a *output,* sobre cada uma das restrições utilizadas. No segundo arquivo, devemos informar o algoritmo acerca da emergência das formas de saída no sistema-alvo, de modo a dizer qual(is) dos candidatos sagra(m)-se como ótimo(s). No caso de variação nos padrões de *output*, o algoritmo precisa ser informado, ainda, sobre a frequência de ocorrência de cada padrão, conforme veremos adiante.

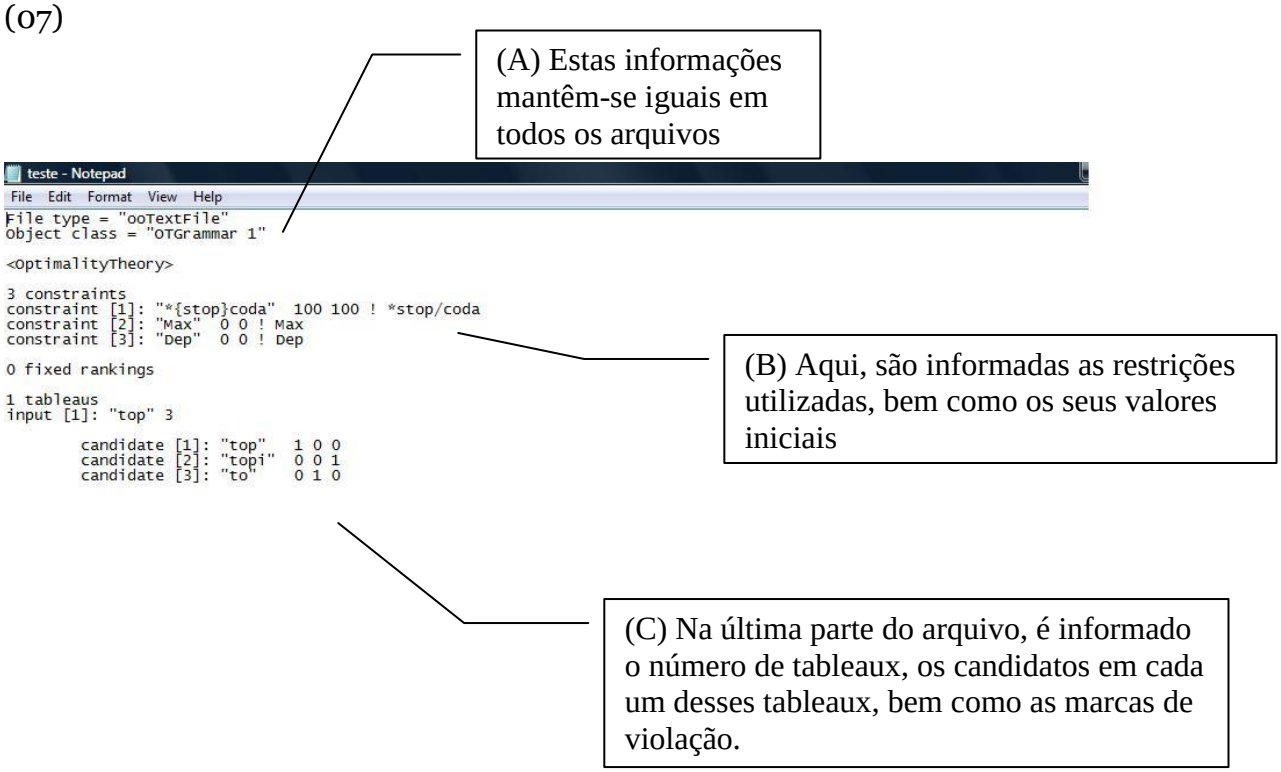
Os arquivos tipo txt seguem uma sintaxe pré-definida e rígida. Dessa forma, apresentamos, no que segue, modelos de arquivos para a simulação em questão. O leitor poderá utilizar-se destes modelos para realizar suas simulações, de modo a modificar, apenas, os detalhes específicos à sua análise.

[9] Esse é, também, o caso do português, no que diz respeito a codas finais. Entretanto, uma vez que não estamos tratando de outros segmentos de coda, e por reconhecermos que a aquisição de coda do português implica a presença de restrições adicionais (cf. Alves, 2008; Quintanilha-Azevedo, 2011), optamos por não associar explicitamente a gramática hipotética utilizada a um sistema linguístico específico.

O primeiro arquivo, conforme já mencionado, deve informar todas as restrições utilizadas na análise, bem como todos os *tableaux* a serem considerados, com todos os candidatos presentes em cada um deles, além das marcas de violação por eles incorridas. Neste arquivo, não é informado o candidato ótimo de cada *tableau*, já que isso será feito no segundo arquivo de texto.

Comecemos, então, pelo primeiro arquivo, que será por nós chamado de "teste.txt":

(07)



A figura em (07) apresenta uma imagem do arquivo que fornece, ao *Praat*, as informações acerca dos *tableaux* utilizados na análise, de modo a descrever as restrições utilizadas e as marcas de violação incorridas por cada um dos candidatos. Para facilitar o entendimento de tal arquivo, dividimos nossa explanação em três etapas (A, B, C), conforme pode ser visto na figura.

As informações em (A) devem ser consideradas como padrão para o *Praat*; tais informações dizem, ao programa em questão, que o arquivo de texto a ser recebido será utilizado para a aplicação de um algoritmo de aprendizagem. Desse modo, toda vez que desejemos rodar um algoritmo no *Praat*, nosso arquivo deverá ser inicializado pelas três primeiras linhas apresentadas em (08), com as seguintes informações:

(08)
“File type = “ooTextFile”
Object class = "OTGrammar 1"

<OptimalityTheory>

Ao contrário da etapa A, que se mostra comum a todos os arquivos de gramática a serem utilizados em simulações, as etapas B e C sofrerão mudanças em função das restrições e da análise proposta pelo analista. Na etapa B, são informadas quais as restrições utilizadas na análise, bem como os valores iniciais destas restrições, ou seja, os valores que elas apresentam antes mesmo da simulação do processo de aprendizagem via algoritmo.

A primeira coisa a fazer, nesta segunda etapa de elaboração do arquivo, é informar o número de restrições utilizadas. No caso da simulação que estamos aqui propondo, contamos com três restrições, o que fica claro através da inscrição “3 constraints”, no arquivo de texto. Definido o número de restrições, devemos descrever cada uma delas; o ordenamento de tal apresentação é indiferente. No caso do arquivo de texto apresentado, optamos por definir a restrição de marcação antes das de fidelidade. As três restrições foram da seguinte forma definidas:

(09)
constraint [1]: "*{stop}coda" 100 100 ! *{stop}coda
constraint [2]: "Max" 0 0 ! Max
constraint [3]: "Dep" 0 0 ! Dep

Cada uma das restrições recebe um número entre colchetes ([1], [2] e [3]). Em seguida, é informado, entre aspas, o nome da restrição. O nome informado não é nada mais do que um nome fantasia, aos olhos do programa. Dessa forma, se chamarmos nossas restrições de “A”, “B” e “C”, ou, ainda, de “João”, “Paulo” e “Luiz”, o programa, mesmo assim, ranqueará tais restrições de acordo com os preceitos do algoritmo.

Após a informação do nome da restrição, seguem dois valores numéricos. Tais valores correspondem aos índices numéricos iniciais a serem assumidos pelo

algoritmo, valores de restrições esses que caracterizarão a gramática inicial. Tomando-se por base o pressuposto de que, no estágio inicial da aquisição de L1, Marcação >> Fidelidade (Demuth 1995; Pater; Paradis, 1996; Smolensky, 1996; Gnanadesikan, 2004; Levelt; Van de Vijver, 2004; Davidson *et al.*, 2004), definimos como peso inicial o valor de 100 para a restrição de marcação, e de 0, para as restrições de fidelidade (ou seja, nesta etapa inicial da aquisição, a gramática resultaria nos *outputs* [topi] e [to], mas nunca [top]). Cabe mencionar que devem ser informados tanto o valor central inicial da restrição (o primeiro índice numérico informado) quanto o valor inicial do ponto de seleção da restrição em questão (o segundo índice numérico), o que explica a inscrição "100 100" ou "0 0". Por uma questão de padronização, optamos por definir os valores centrais e de ponto de seleção como iguais, no estágio inicial. Após informados esses dois valores numéricos, a sintaxe do algoritmo solicita que seja informado, após um ponto de exclamação, o nome ou rótulo da restrição que deverá ser apresentado nos *tableaux*, quando o resultado da gramática final for fornecido pelo programa, através do ambiente gráfico característico do *Praat*.

Finalmente, na terceira etapa de elaboração do primeiro arquivo, são informados os *tableaux* utilizados na análise. Uma vez que, na análise que estamos aqui apresentando, contamos com uma única forma de *input*, teremos, nesse caso, apenas um *tableau*. As informações devem ser da seguinte forma apresentadas:

(10)
1 tableaus
input [1]: "top" 3

candidate [1]: "top" 1 0 0
candidate [2]: "topi" 0 0 1
candidate [3]: "to" 0 1 0

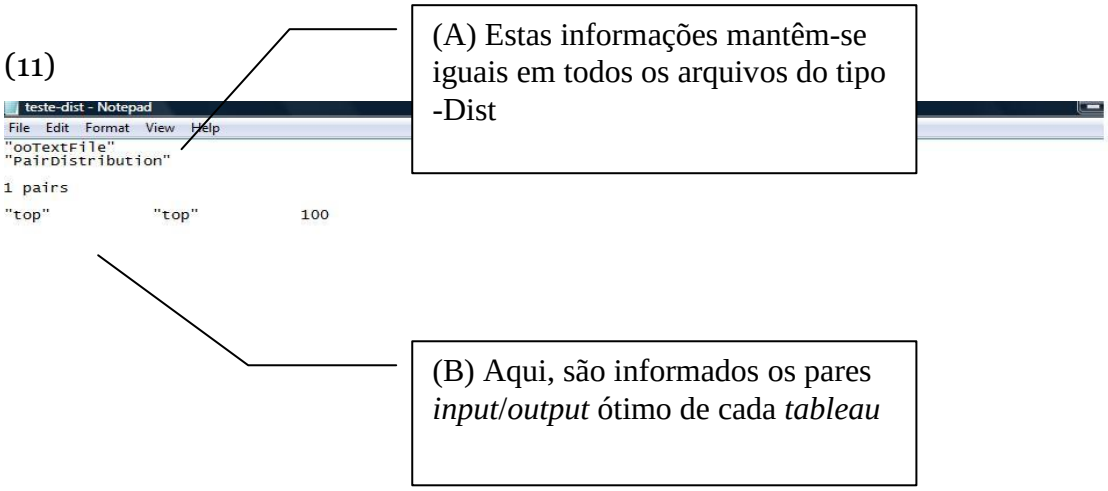
Após ser informado o número de *tableaux*, passa-se à descrição de cada um deles. É informado o input da aquisição (no caso, "top"), e, logo após, o número de candidatos a *output* (3, no caso em questão). É importante lembrar que a escolha do candidato ótimo só é informada no segundo arquivo de texto que alimentará o algoritmo.

Após essa caracterização do número de possíveis formas de saída, chegamos, então, à descrição de cada um dos candidatos a *output*, apresentados entre aspas duplas. No exemplo acima, os candidatos são "top" (fiel), "topi" (com epêntese) e "to" (sem epêntese, com apagamento da plosiva). Como em qualquer análise à luz da OT, o número de candidatos é definido pelo pesquisador em sua análise linguística. Ao lado da descrição de cada candidato, são apresentadas as marcas de violação de candidato em cada uma das restrições previamente caracterizadas. Por exemplo, o candidato "top" é descrito como 1 0 0, o que significa que ele viola uma vez (um asterico) a restrição 1 ({stop}coda), não viola a restrição 2 (Max) e não viola a restrição 3 (Dep). O segundo candidato, com a informação 0 0 1, viola uma vez a terceira restrição (Dep). Finalmente, o terceiro candidato viola Max, a segunda restrição. O ordenamento das marcas de violação obedece à ordem em que as restrições foram apresentadas na etapa B. Cabe ressaltar que, caso um dos candidatos violasse duas vezes uma mesma restrição, o número seria 2, uma vez que os algarismos informados nesta etapa correspondem ao número de asteriscos que cada um dos candidatos impõe sobre as restrições em questão.

Está pronto, assim, o primeiro arquivo que deverá ser fornecido ao *software Praat*. Precisamos salvá-lo com a extensão .txt (como pode ser visto, este arquivo modelo está intitulado como "teste.txt"). É importante que prestemos atenção ao nome do arquivo, pois o segundo arquivo de texto, cuja descrição iniciaremos a seguir, deverá ter exatamente o mesmo nome do primeiro, acrescentando-se, somente, a inscrição "–dist". Dessa forma, dado que o arquivo recém elaborado tem o nome "teste.txt", o segundo arquivo a ser informado ao programa, que informa os *outputs* ótimos de cada *tableau*, deverá ser denominado de "teste-dist.txt".

No que segue, apresentamos a gravura do modelo de arquivo "teste-dist.txt", que elaboramos para o exemplo de simulação aqui fornecido:

(11)



Os arquivos –dist são arquivos do tipo "*Pair Distribution*"; é através deles que informamos o(s) *output*(s) ótimo(s) de cada um dos *tableaux*. Com tais informações, estamos informando, em outras palavras, os padrões de saída a serem alcançados no estágio final da gramática. A etapa (A) de elaboração do referido arquivo é comum a todos os arquivos do tipo –dist. Ou seja, para que o *Praat* interprete tal arquivo de texto como um informativo do mapeamento *input*/*output*, deve-se sempre iniciar com a seguinte inscrição:

(12)
"ooTextFile"
"PairDistribution"

A etapa B, por sua vez, variará em função da análise proposta. Neste momento, informamos o número de pares *input/output* ótimos a serem considerados no processo de aquisição, ou seja, quantos padrões de *output* ótimo a gramática final deve fornecer. No caso de nossa análise, contamos com apenas um *tableau*, e, conforme já afirmamos, a forma-alvo a ser atingida é o padrão fiel ([top]), que deverá ser adquirido como um *output* categórico. Dessa forma, devemos informar ao arquivo apenas um par. Caso a gramática a ser adquirida resultasse em *outputs* variáveis, seria necessário informar a existência de dois ou mais pares, conforme será visto mais adiante.

Tendo sido informado o número de pares a serem considerados pelo programa (através da informação “1 pairs”), descreve-se o par em questão. No arquivo-exemplo que mostramos, apresentamos a seguinte informação:

(13)
"top" "top" 100

O primeiro termo entre aspas corresponde ao *input* do tableau apresentado no arquivo anterior. O segundo termo corresponde ao candidato ótimo do *tableau* em questão. O índice numérico corresponde à percentagem de emergência de tal candidato como ótimo, sob o *input* em questão. Como estamos tratando da aquisição de um padrão categórico de *output*, temos 100% de emergência de tal *output* ótimo, o que nos leva a informar o algarismo 100. Dessa forma, podemos ler a linha acima como se fosse a seguinte instrução: “no tableau com *input* /top/, a gramática deve resultar no *output* [top] em 100% dos casos”.

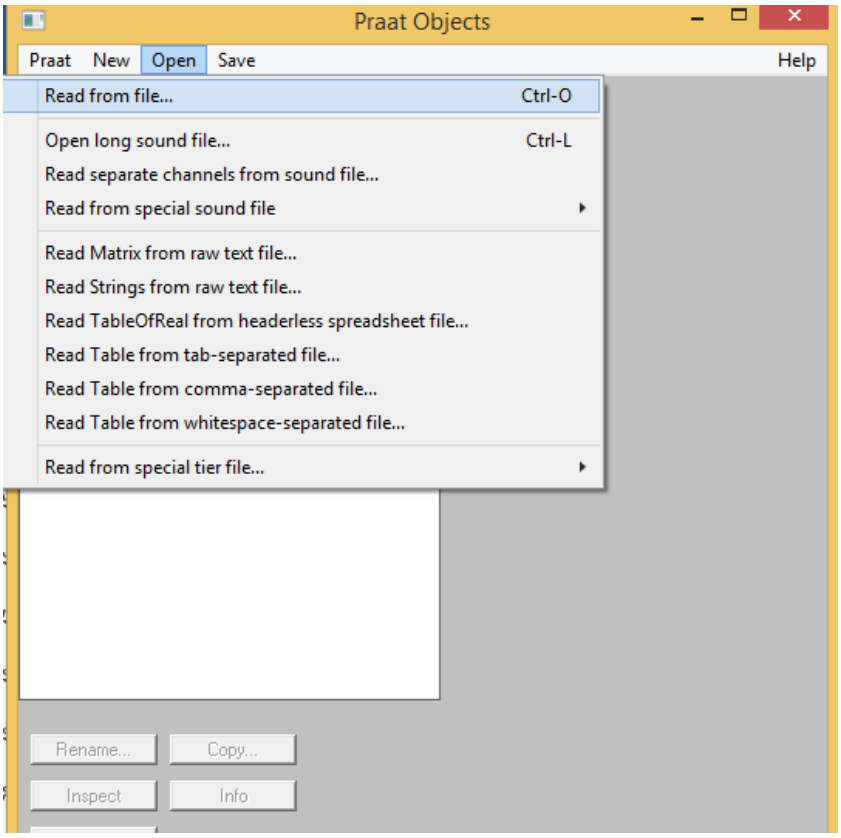
Em suma, finalizada esta etapa, temos, finalmente, os dois arquivos que norteiam o funcionamento do algoritmo. De posse desses dois arquivos, podemos alimentar o algoritmo do *Praat*. A seção que segue descreverá os passos que devem ser tomados.

3.2 ALIMENTAÇÃO DO ALGORITMO

Consideremos que o *software Praat* já esteja instalado no computador. Caso não esteja, é preciso baixá-lo em www.praat.org. Para abrir o *software Praat*, clicamos duas vezes sobre o ícone que está na área de trabalho. Neste momento, abrirão duas janelas: *Praat Objects* e *Praat Window*. A janela *Praat Window* pode ser fechada, pois, para a simulação computacional com o algoritmo, utilizaremos sempre a *Praat Objects*.

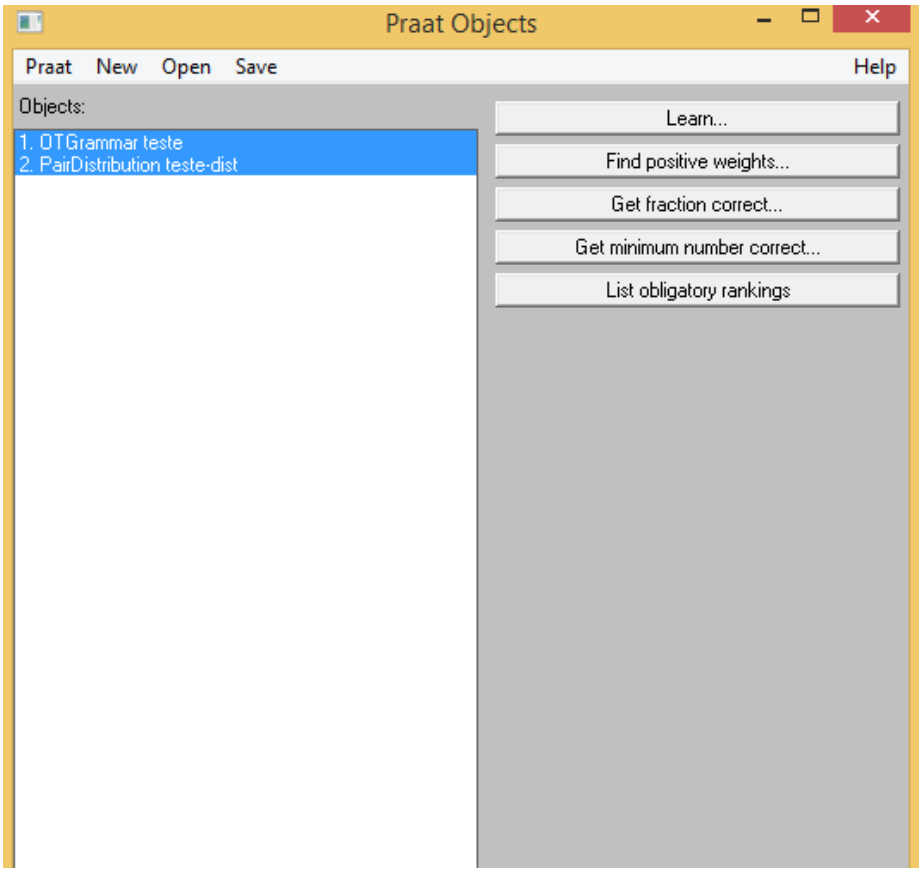
Com a janela *Praat Objects* aberta, alimentamos o programa com os dois arquivos previamente preparados. Para isso, devemos clicar no comando “*Open*”, e, logo após, “*Read from file*”.

(14)



Selecionamos, um de cada vez, os dois arquivos de texto preparados, de modo que, ao final da operação, ambos apareçam na área de trabalho do *Praat Objects*. O arquivo "teste.txt" deverá aparecer automaticamente sob a inscrição "*OT Grammar teste*", e o arquivo –dist aparecerá como "*PairDistribution teste-dist*", conforme pode ser visto no que segue:

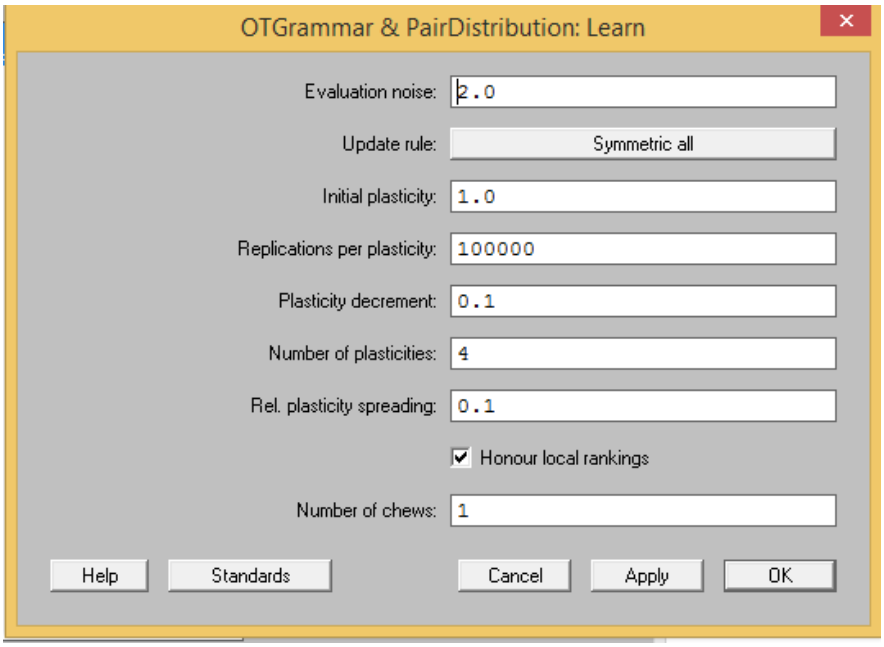
(15)



Isso quer dizer que as sintaxes dos dois arquivos de texto estão bem elaboradas, e que o programa pôde interpretar a função de cada um dos arquivos de texto previamente preparados. Tendo chegado a esta fase, basta solicitar que o algoritmo leia os dois arquivos, em conjunto, e aprenda a gramática prevista. Para isso, selecionamos os dois arquivos em questão, conforme a figura em (15).

No momento em que ambos os arquivos estiverem selecionados, na barra de ferramentas ao lado da lista de arquivos, surgirá o botão de comando "*Learn...*" (aprenda). Ao clicarmos neste botão, damos início ao processo de aquisição da gramática-alvo. Após o botão ser clicado, o programa fornecerá uma nova tela, requerendo informações acerca dos parâmetros numéricos a partir dos quais a simulação será feita:

(16)



Não nos deteremos por muito tempo na descrição de cada um dos parâmetros apresentados em (16). Aconselhamos, em um primeiro momento, a utilização dos valores *default* fornecidos pelo programa, valores esses que podem ser vistos na figura. Cabe mencionar, entretanto, que é neste momento que o pesquisador tem a oportunidade de determinar, a seu gosto ou em virtude de suas necessidades, diferentes índices de ruído (que, conforme vimos na seção 2, determina os índices de variação), plasticidade inicial, replicações por plasticidade, e decremento de plasticidade à medida em que a gramática for se aproximando do alvo. Nesta etapa é definido, também, o número de exemplares de *input* com o qual o programa vai ser alimentado, sendo 100.000 o valor *default* de repetições sugerido pelo programa.

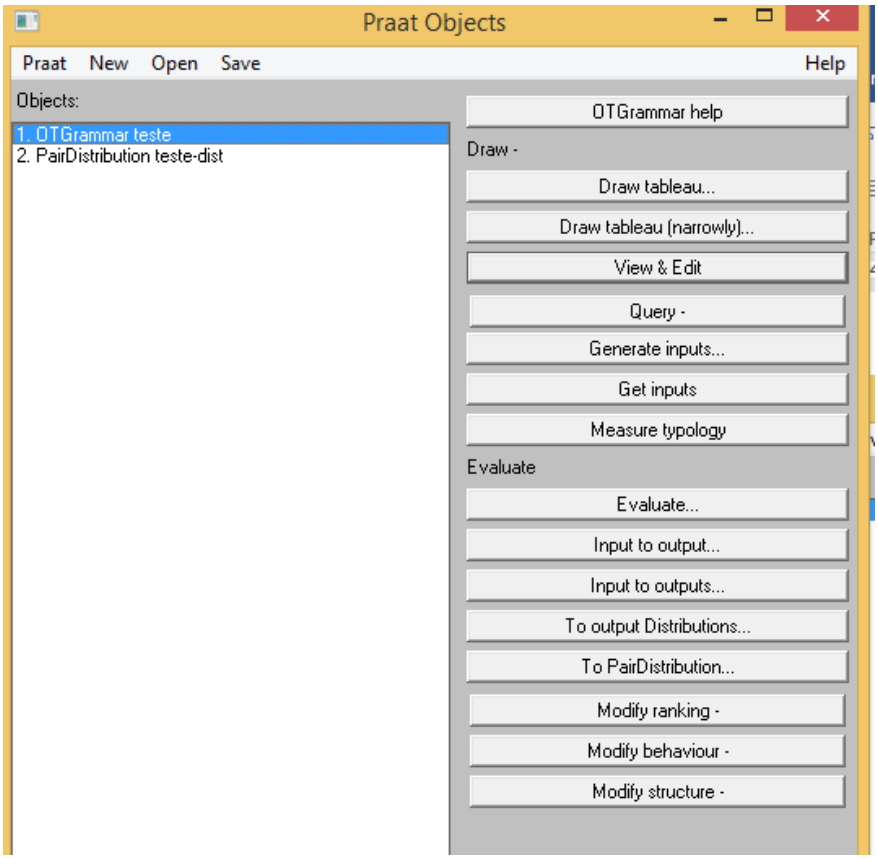
Ao utilizarmos, neste trabalho, os valores *default* fornecidos pelo programa, mantemos os valores apresentados na tela e clicamos em "OK". Através deste comando, o algoritmo entra em ação, e a simulação é efetivada. O algoritmo atingirá, dessa forma, a gramática alvo.

3.3 VERIFICAÇÃO DOS RESULTADOS

Para ver os novos valores numéricos das restrições, que caracterizam a gramática adquirida, o seguinte procedimento deve ser adotado: na janela *Praat*

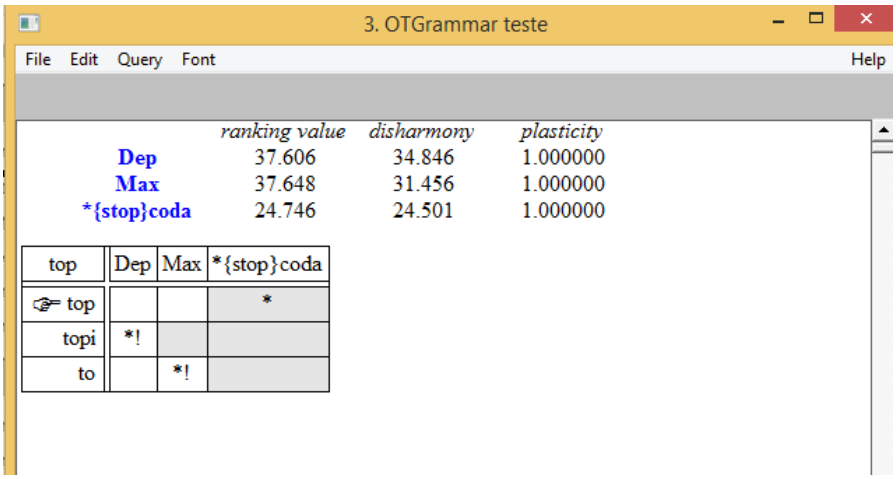
Objects, após a etapa de aprendizagem, selecionamos o arquivo *OT-Grammar Teste*, conforme pode ser visto a seguir:

(17)



Através da seleção em questão, surgem uma série de novos comandos. É necessário, então, clicar em "*View & Edit*". O comando em questão fornecerá o resultado final da gramática, de modo a informar tanto os valores centrais das restrições, os valores dos pontos de seleção e, ainda, os *tableaux* que expressam a escolha do candidato ótimo, ao serem considerados os valores dos pontos de seleção no momento da avaliação em questão. Vejamos o exemplo em (18).

(18)



Retomamos que, antes da simulação do processo de aprendizagem, Dep e Max apresentavam valor central 0, e*{stop}coda, por ser uma restrição de marcação, havia sido definida com valor central inicial de 100. Após o processo de aprendizagem, vemos que, na simulação que nos serve de exemplo, Dep apresenta um valor central de 37,606, Max apresenta um valor central de 37,648 e a restrição de marcação *{stop}coda, por sua vez, possui o valor central de 24,746. Em outras palavras, a restrição de marcação foi demovida, enquanto que as de fidelidade foram, por sua vez, promovidas. É importante mencionar que o resultado final da gramática resulta em um *output* categórico, pois a restrição de marcação está bastante afastada e com valor central bastante inferior aos apresentados pelas restrições de fidelidade, exibindo uma diferença de mais de 10 pontos. Dessa forma, considerando-se os preceitos de funcionamento do GLA, ainda que possa haver variação nos valores de ponto de seleção das três restrições, a relação de dominância de fidelidade sobre marcação se mostra categórica, neste caso específico.

Conforme podemos ver na figura, o comando *Edit* fornece, também, os valores de ponto de seleção de um determinado momento de avaliação, bem como o *tableau* correspondente a tal momento. Os valores de ponto de seleção são apresentados sob o rótulo de "*disharmony*". No momento de avaliação apresentado pelo programa, expresso em (18), as restrições apresentam os valores de ponto de seleção de 34,864 (Dep), 31,456 (Max) e 24,501 (*{stop}coda). O *tableau* apresentado em 18 explicita a relação de dominância determinada pelos valores de ponto de seleção ali retratados: como Dep está apresentando, no momento de produção em questão, um valor de ponto de seleção superior ao de Max (ainda que o valor central de Max seja superior

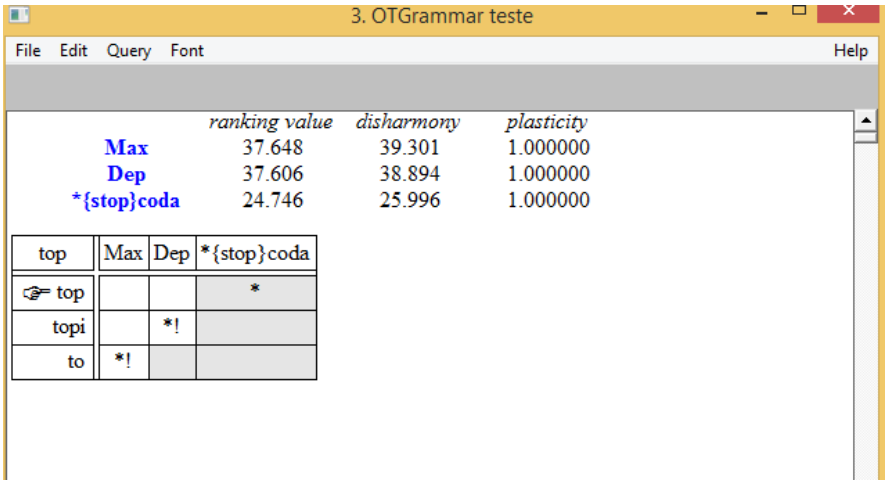
ao de Dep), o *tableau* é organizado de tal forma que Dep>>Max. Ressaltemos que, independentemente dos valores de ponto de seleção das duas restrições de fidelidade, o *output* ótimo é sempre o mesmo, uma vez que ambas se apresentam com valores centrais bastante superiores (e, portanto, bastante afastadas) ao da restrição de marcação.

Conforme discutimos na apresentação dos pressupostos do GLA, na seção 2, os valores de ponto de seleção são alteráveis a cada momento de avaliação (produção linguística), valores esses que se encontram sempre dentro de uma faixa numérica pré-definida. A simulação computacional, via *Praat*, mostra-se capaz de demonstrar, também, a possibilidade de variação dos pontos de seleção a cada momento de ato de fala. Para isso, podemos solicitar inúmeras "avaliações" (i.e., momentos de produção) da gramática em questão, a partir dos valores centrais atingidos pelo algoritmo. Em cada avaliação, os valores de ponto de seleção serão distintos. Uma vez que os valores centrais das restrições de fidelidade se mostram muito próximos (37,648 para Max e 37,606 para Dep), é possível que, em algumas avaliações, o valor de ponto de seleção de Dep seja maior do que o de Max, e vice-versa. Entretanto, conforme já afirmamos, como o algoritmo foi instruído a adquirir uma gramática que resultasse em *outputs* categóricos, os pontos de seleção dessas duas restrições serão sempre superiores a qualquer valor de ponto de seleção de *{stop}coda, uma vez que o valor central dessa restrição de marcação é bastante inferior, o que, à luz do GLA, caracteriza dominância categórica.

Para solicitarmos diferentes avaliações, ainda na janela que exibe os resultados da nova gramática, basta que cliquemos em *Edit* e solicitemos o comando *Evaluate* (*noise 2*)[10]. Cada solicitação do comando em questão representa um momento de produção linguística. Dessa forma, cada solicitação resultará em valores de pontos de seleção diferentes daqueles previamente encontrados. Como exemplo, podemos mostrar um momento de avaliação em que, diferentemente do visto em (18), Max tem ponto de seleção superior ao de Dep, de modo que, na avaliação em questão, Max >> Dep.

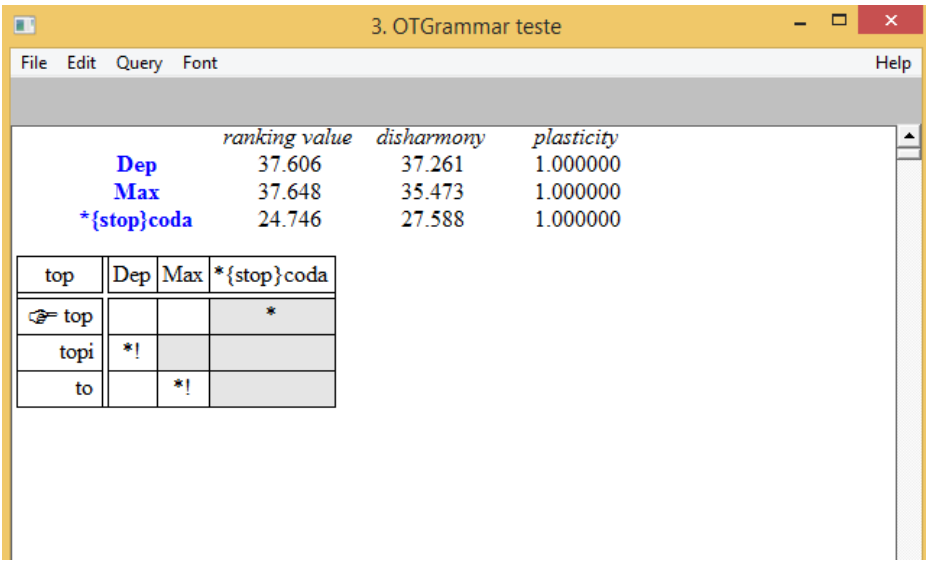
[10] Há várias opções para o comando *Evaluate*, além de *Evaluate (noise 2.0)*. Dentre essas opções, vale ressaltar que o comando "*Evaluate – zero noise*" faz a avaliação dos candidatos sem ruído (e, portanto, não há variação nos índices de ponto de seleção, de modo que não haja alteração nesses valores). Por sua vez, o comando "*Evaluate*" permite que o analista defina o valor de ruído com que queira operar, possibilitando-o ir além do valor *default* 2.

(19)



Já em outro momento, assim como na avaliação previamente mostrada em (18), o valor de ponto de seleção de Dep é superior ao de Max:

(20)



Ressaltemos, novamente, que tais relações não implicam distinção no *output* ótimo, em função do valor central da restrição de marcação. Também é importante mencionar que os valores obtidos a partir de tal comando serão diferentes a cada avaliação, ainda que os valores centrais sejam sempre os mesmos. Dessa forma, os valores de ponto de seleção aqui apresentados mostram-se apenas como exemplos, de modo que não serão igualmente atingidos pelo leitor em suas simulações.

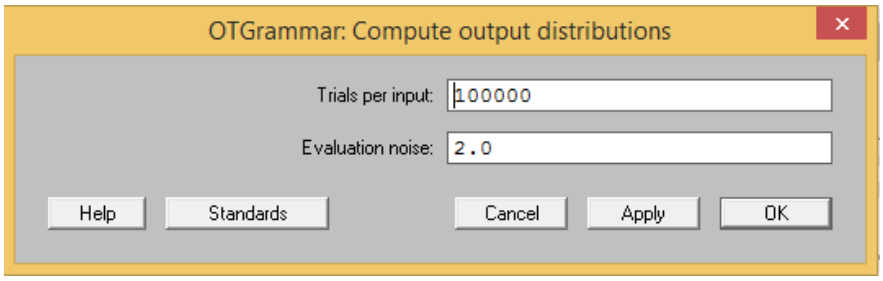
O caso que acabamos de demonstrar é bastante simples: pelo fato de sabermos que a restrição de marcação apresenta um valor central inferior em

aproximadamente 13 pontos ao das duas restrições de fidelidade, não há dúvidas de que não há cruzamento entre as faixas de valores das restrições de fidelidade e marcação e, portanto, o *output* é categórico. Entretanto, há casos, sobretudo quando solicitamos a aquisição de gramáticas que resultam em *outputs* variáveis, em que se mostra fundamental verificar as percentagens de emergência de cada um dos candidatos a *output* fornecidos pela gramática.

Frente a essa necessidade, o *software Praat* possibilita um comando adicional, chamado de "*Output Distributions*" (Distribuições de *Output*), que informa as percentagens de emergência de cada candidato a ótimo, a partir dos valores numéricos da gramática resultante do processo de aquisição. No caso da gramática categórica em questão, o resultado é simples: o candidato [top] deve apresentar 100% de ocorrência, ao passo que [to] e [topi], 0%. Se esse resultado (semelhante ao estipulado no arquivo -dist) for atingido, temos uma evidência de que o algoritmo convergiu em uma gramática correta.

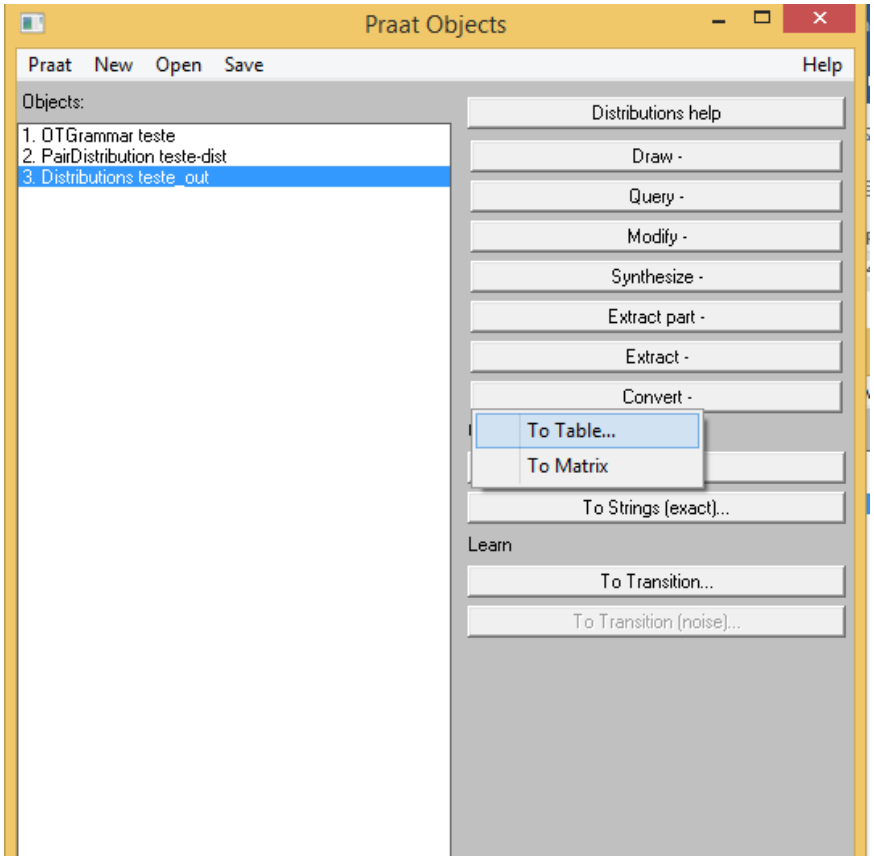
Para procedermos à verificação em questão, selecionamos, novamente na janela *Praat Objects*, o arquivo "*OT Grammar teste*", e, após, isso, selecionamos o comando "*To Output Distribution*". Ao clicarmos no comando em questão, uma nova janela aparecerá. Através de tal janela, definiremos os valores de tentativa por *input* (*trial per input*) e ruído (*evaluation noise*). Aconselhamos que, novamente, o usuário se mantenha fiel aos valores *default*, sugeridos pelo programa. O valor de ruído 2.0 é exatamente o mesmo aquele já utilizado anteriormente, durante a simulação do processo de aprendizado. O valor de *trials per input* é, também, um valor muito importante, pois é a partir de tal índice numérico que o programa calculará as percentagens de ocorrência de cada um dos candidatos a *output*, conforme veremos a seguir.

(21)



Selecionados os valores *default*, cliquemos em OK. Surgirá, neste momento, um novo item da lista de objetos: "*Distributions teste_out*". Tal item equivale ao resultado da verificação das percentagens de distribuição de cada *output*, a partir da gramática recém-aprendida. Para observar tais resultados, é necessário selecionar o item em questão e clicar na opção "*Convert to table*", conforme mostra a figura (22).

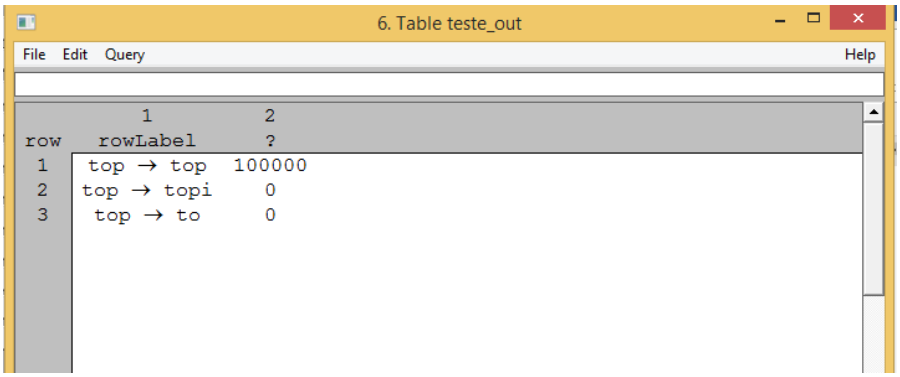
(22)



Na tela seguinte, clicamos em OK e, por fim, o resultado será uma Tabela também listada no *Object Window*. Para a verificação do resultado final de tal

simulação, basta selecionar tal objeto e clicar em *View & Edit*. O resultado pode ser visto em (23).

(23)



A tabela apresenta os índices de ocorrência de *output* para cada par *input*-candidato a *output*. Como podemos ver em (23), nossa gramática resulta em um *output* categórico. Conforme havíamos determinado previamente, o valor de "*trials per input*" é de 100.000. Isso significa dizer que, na verificação realizada pelo programa a partir do comando "*Output Distributions*", foram realizadas 100.000 avaliações. Considerando-se tais 100.000 exposições ao *input* /top/, no caso da gramática aqui aprendida, somos informados, em (23), da emergência do *output* fiel em 100.000 do número total de 100.000 exposições, o que equivale a 100% de ocorrências de tal padrão de *output*. Temos, portanto, uma emergência categórica do mesmo padrão, que se mostra de acordo com a solicitação previamente realizada por meio do arquivo -dist. Isso fica confirmado através dos mapeamentos /top/→[topi] e /top/→[to], que apresentam distribuições iguais a 0, considerando-se o índice de 100.000 exposições ao *input* /top/. Em outras palavras, a partir da gramática à qual convergiu o algoritmo, o *input* /top/ não resulta em *outputs* infiéis. Uma vez que isso era o que havíamos solicitado previamente, temos a certeza de que o algoritmo convergiu corretamente.

3.4 AQUISIÇÃO DE HIERARQUIAS COM *OUTPUTS* VARIÁVEIS

Conforme afirmamos na segunda seção, a variação é um aspecto inerente do GLA. À luz do GLA, *rankings* variáveis podem ser o resultado de duas diferentes situações: (a) um sistema que ainda não atingiu o estado final de desenvolvimento

(gramática final essa que pode, inclusive, ser categórica); (b) um sistema cujo próprio estado final de desenvolvimento é, efetivamente, um *ranking* que resulta em mais de um *output* ótimo (ou seja, o sistema-alvo resulta em *outputs* variáveis). Na simulação que acabamos de realizar na seção anterior, conforme já dissemos, a gramática final foi efetivamente atingida, e o *output* atingido pela gramática foi sempre categórico. A questão que fica é: como demonstrar os estágios intermediários de aquisição dessa gramática, característicos do processo de aquisição de linguagem? No que diz respeito à possibilidade levantada em (a), uma solução é diminuir a quantidade de exposições ao *input* nas simulações computacionais, para que possamos evidenciar o crescimento em direção à gramática final à medida em que o aprendiz for recebendo uma maior quantidade de exposição à língua-alvo. Para isso, basta que o pesquisador altere, após selecionar o comando "*Learn...*", o valor do índice de "*replications per plasticity*". O *default* de 100.000 replicações constitui um valor inegavelmente alto, que leva à convergência, caracterizada pela aquisição plena e a chegada à gramática final. Entretanto, caso venhamos a diminuir tal índice numérico, estaremos diminuindo o número de exposições à evidência positiva e, consequentemente, as possibilidades de chegada ao estágio final da gramática. Poderemos ir fazendo, manualmente, diversas simulações com um número bastante baixo de exposições ao *input*, para que possamos verificar "estágios intermediários" a uma gramática categórica, estágios esses a partir dos quais emergirão *outputs* variáveis.

Por fins de delimitação deste trabalho, não apresentaremos os *outputs* de uma sequência de simulações que sigam a possibilidade analítica mencionada em (a). A partir de agora, voltaremos nossa atenção para a possibilidade (b), ou seja, veremos como o GLA é capaz de convergir a *outputs* variáveis oriundos de uma gramática-alvo caracterizada pela variação. Esses *outputs* variáveis são previamente estipulados pelo pesquisador para representar (i) sistemas-alvo caracterizados por gramáticas variáveis do adulto; (ii) estágios desenvolvimentais intermediários, verificados nos casos de aquisição de linguagem (L1 ou L2), definidos, a efeitos de simulação computacional, como estágios finais.

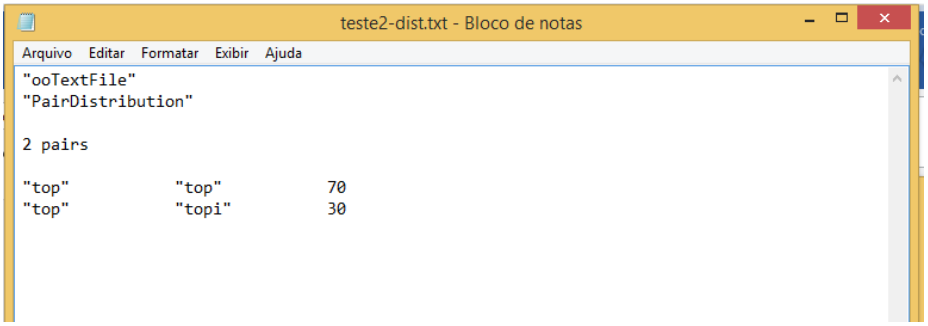
Para exemplificarmos a convergência a gramáticas que resultam em mais de uma forma de saída, mantenhamos o exemplo apresentado na seção anterior, referente à aquisição dos padrões de coda final, com uma pequena alteração: suponhamos um sistema alvo em que plosivas finais ([top]) são produzidas em variação com formas que exibem epêntese ([topi]), sem nunca haver a emergência de

apagamentos ([to]), de modo que desta gramática hipotética venham a emergir 70% de plosivas finais e 30% de vogais epentéticas.

O leitor vai verificar que os comandos que solicitam a rodagem do algoritmo são exatamente os mesmos já apresentados anteriormente. A diferença, como era de se esperar, encontra-se unicamente no arquivo de texto –dist. Para fins de explanação, pensemos em arquivos chamados de "teste2.txt" e "teste2-dist.txt". O arquivo "teste2.txt" – que contém as caracterizações dos *tableaux*, dos candidatos e das marcas de violação – vai ser exatamente o mesmo apresentado em "teste.txt". A diferença estará no arquivo de distribuições "teste2-dist.txt", uma vez que teremos que definir a percentagem de emergência de cada um dos candidatos a *output*, conforme pode ser visto no que segue.

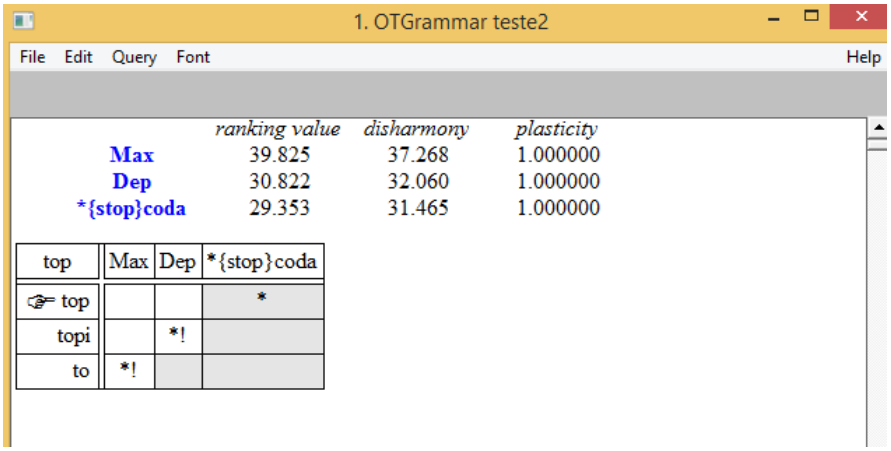
Conforme apontado em (24), notamos, na sintaxe do arquivo em questão, a presença de dois pares *input-output*. É importante chamar a atenção, na sintaxe, para a inscrição "2 pairs" no topo do arquivo, o que chama a atenção para o fato de que haverá dois pares de *input-output* ótimos. O primeiro par diz respeito à emergência da plosiva em coda, que deverá ocorrer em 70% dos casos. Os 30% restantes serão caracterizados pela emergência da epêntese, que corresponde ao segundo par *input-output*. Independentemente do número de candidatos ótimos que emergem em um contexto de variação linguística, é importante que a soma das percentagens de ocorrência dos *outputs* seja, sempre, 100 (ou seja, 100% dos casos de avaliação).

(24)



Tendo sido finalizado o processo de elaboração dos arquivos, o procedimento para alimentação do algoritmo é o mesmo: selecionamos os dois arquivos ("teste2.txt" e "teste2-dist.txt") no *Praat Objects* e solicitamos o comando "*Learn...*" Apresentamos, em (25), o resultado da gramática obtida após tal solicitação, considerando-se os valores *default* do programa:

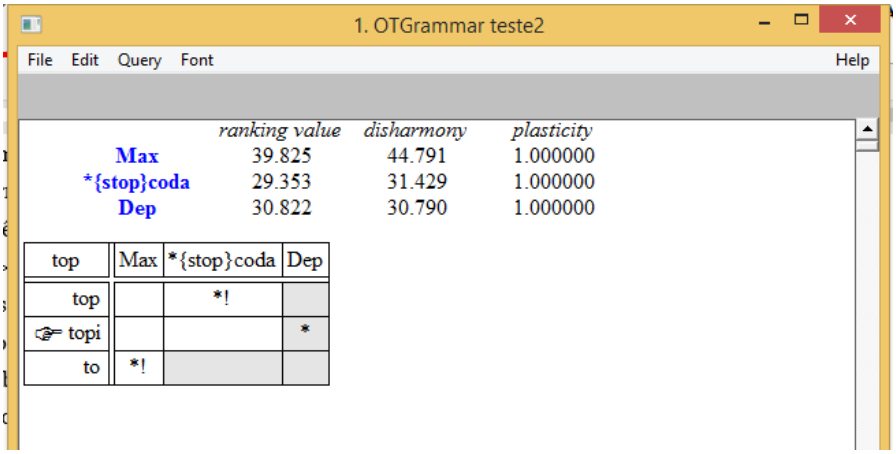
(25)



Os resultados acima apresentados demonstram numericamente os fundamentos do GLA, apresentados previamente na seção 2. A restrição Max apresenta o valor central de 39,825, bastante afastado e superior ao de Dep (30,822) e ao de *{stop}coda (29,353). Os 10 pontos de diferença entre Max e a restrição de marcação mostram que o apagamento nunca vem a ser uma alternativa à emergência da coda. Já os valores centrais de Dep e *{stop}coda nos permitem afirmar que há ocorrência variável de *outputs,* com a plosiva em coda ou com epêntese. O valor central levemente superior de Dep caracteriza uma evidência de que a relação Dep >> *{stop}coda seja a mais frequente, de modo que haja uma maior frequência de

outputs com a plosiva em coda. De fato, no momento de avaliação apresentado (25), isso é o que ocorre. Entretanto, se continuarmos solicitando avaliações ("*Evaluate – Noise 2*"), poderemos, também, encontrar o candidato ótimo com epêntese:

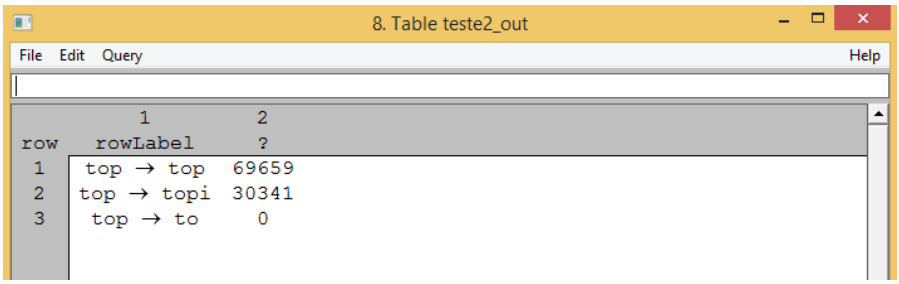
(26)



Uma vez que estamos lidando com gramáticas cujas restrições apresentam cruzamento em suas faixas de valores, uma etapa fundamental do processo de simulação é comprovar, efetivamente, as distribuições de frequência de *output* que emergem a partir de tal gramática. De fato, em gramáticas com *outputs* variáveis, a importância do comando "*Output Distributions*" se mostra ainda mais acentuada, pois, ao contrário de gramáticas categóricas, que apresentam valores centrais bastante afastados e, portanto, caráter categórico visualmente visualizável, nas gramáticas que exibem restrições com cruzamentos de faixas não temos como afirmar, apenas através da observação, se os valores percentuais referentes às frequências de emergência de cada padrão de *output,* decorrentes da gramática aprendida, condizem, efetivamente, com os previamente estabelecidos e solicitados por meio do arquivo -dist.

Frente a esta necessidade, devemos solicitar o comando "*Output Distributions*", seguindo-se os comandos já apresentados anteriormente. Para fins de verificação, os resultados da aplicação deste comando, no caso da simulação da gramática apresentada em (25) e (26), são retratados em (27).

(27)



A função "*Output Distributions*" permite verificar que, na possibilidade de 100.000 avaliações, 69.659 dessas avaliações (69,66%) resultarão no *output* com a plosiva em coda, e 30.341 (30,34%), na emergência da epêntese. Os resultados vão ao encontro dos valores de 70% e 30% estipulados através do arquivo -dist, e a pequena diferença entre a frequência verificada e a previamente estipulada se deve ao ruído estatístico, que pode alterar, moderadamente, a frequência dos padrões de saída. Em outras palavras, com esse recurso, verificamos que a o algoritmo conseguiu convergir em uma gramática que resultasse na frequência de distribuição de *outputs* estipulada pelo pesquisador.

Em suma, demonstramos, nesta subseção, a capacidade do algoritmo de aprender gramáticas que resultam em mais de um *output* a partir de um mesmo *input*. Antes de encerrarmos esta seção, cabe, aqui, discutirmos a pertinência de tal capacidade do GLA. Através de tal algoritmo, é possível formalizar sistemas adultos caracterizados pela variação linguística. Assim, o estágio final da aquisição é, portanto, uma gramática que resulta em variação. Essa capacidade, embora ainda pouco utilizada, pode vir a ser de extrema importância para pesquisadores voltados para a Teoria Variacionista, que podem formalizar a variação linguística a partir dos mesmos princípios e aparatos teóricos utilizados para a formalização de gramáticas categóricas. Não são necessárias regras especiais, ou condições especiais para sistematizar a variação. A variação para o GLA é, efetivamente, parte inerente da gramática, afirmação essa que fica evidente através da atuação do algoritmo.

Antes de finalizar, cabe, ainda, mencionar a viabilidade de o algoritmo simular dados, também, de aquisição de L2. Ainda que, por razões de delimitação, não venhamos a demonstrar o passo-a-passo de tais simulações, julgamos importante mencionar o cuidado de adotarmos pelo menos dois passos analíticos. Conforme apontado por Alves (2013), é preciso (i) simular a aquisição da gramática da L1 (a partir de um estágio inicial em que M >> F); (ii) dado o resultado dessa primeira

simulação, alimentar os valores da gramática de L1 como estágio inicial para a aquisição da gramática que caracterizará a língua do aprendiz. Apesar de a aquisição de L2 implicar, portanto, dois estágios, os fundamentos são os mesmos usados na simulação da linguagem variável do adquirente de L1, ou da formalização de uma gramática final que resulte em *outputs* variáveis de uma comunidade adulta. Esse expediente formal comum será o tópico central de discussões da próxima seção, na qual finalizaremos o presente tutorial.

CONCLUSÃO

Neste trabalho, após apresentarmos os fundamentos da OT Estocástica e de seu Algoritmo de Aprendizagem Gradual, demonstramos como tal algoritmo pode ser aplicado em uma simulação computacional, através do *Software Praat* (Boersma; Weenink). Verificamos a capacidade do algoritmo de convergir em gramáticas que resultam tanto em *outputs* categóricos quanto em variáveis, a partir dos mesmos mecanismos formais.

Tais procedimentos permitem uma reflexão importante a respeito da relação entre teoria de gramática e variação linguística: no GLA, os mecanismos formais que dão conta de gramáticas categóricas são, efetivamente, os mesmos que respondem por sistemas que resultam em mais de um *output* a partir de um mesmo *input*. Acreditamos ter demonstrado, através das simulações com o algoritmo, o elo entre variação, aquisição e gramática, que, ao longo de todo o texto, defendemos ser a base do GLA. Além disso, uma variação resultante de etapas desenvolvimentais, apresentada pelo algoritmo em função de uma exposição ainda insuficiente à evidência positiva (em termos de simulação computacional, um número baixo de "*replications per plasticity*"), segue os mesmos princípios formais da variação linguística que ocorre na fala adulta. Não são necessários expedientes formais adicionais, uma vez que a gramática a ser adquirida pelo GLA-OT pode resultar tanto em *outputs* categóricos quanto em variáveis. Dessa forma, concluímos que, em termos formais, o GLA associado à OT Estocástica consegue expressar, sob o mesmo mecanismo, aquisição e variação linguística.

Esperamos que as instruções fornecidas ao longo deste tutorial incentivem o leitor a realizar as suas próprias simulações computacionais do processo de aquisição, em L1 ou L2, tanto de gramáticas variáveis quanto categóricas.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, Fernando Cabral; LUCENA, Rubens Marques de. Aquisição da lateral silábica do inglês: uma análise via Teoria da Otimidade Estocástica. *Letrônica*, v.7, n.2, p. 795-820, 2014.

ALVES, Ubiratã Kickhöfel. *A aquisição das sequências finais de obstruintes de inglês (L2) por falantes do Sul do Brasil: análise via Teoria da Otimidade*. Tese (Doutorado em Letras). Porto Alegre: Pontifícia Universidade Católica do Rio Grande do Sul, 2008.

ALVES, Ubiratã Kickhöfel. Teoria da Otimidade, Gramática Harmônica e Restrições Conjuntas. *Alfa: Revista de Linguística*, v. 54, p. 237-263, 2010.

ALVES, Ubiratã Kickhöfel. A epêntese vocálica na aquisição das plosivas finais do inglês (L2): tratamento pela OT Estocástica e pela Gramática Harmônica. In: LEE, Seung Hwa (org.). *Vogais além de Belo Horizonte*. Belo Horizonte: Editora da UFMG, p. 263-308, 2012.

ALVES, Ubiratã Kickhöfel. Aquisição fonológica de L2: formalização de fenômenos variáveis na língua-fonte, na língua-alvo e em seus sistemas intermediários. In: BISOL, Leda; COLLISCHONN, Gisela (orgs.). *Fonologia*: *teoria e perspectivas*. Porto Alegre: EDIPUCRS, 2013, p. 133-147.

BOERSMA, Paul; HAYES, Bruce. Empirical Tests of the Gradual Learning Algorithm. *Linguistic Inquiry*, v. 32, p. 45-86, 2001.

BOERSMA, Paul; WEENINK, David. *Praat: Doing phonetics by computer*. www.praat.org.

COETZEE, Andries W.; PATER, Joe. The place of variation in phonological theory. In: GOLDSMITH, John; RIGGLE, Jason; YU, Alan C. L. *The Handbook of Phonological Theory*. 2a edição. Blackwell, 2009, p. 401-434.

DAVIDSON, Lisa; JUSCZYK, Peter; SMOLENSKY, Paul. The initial and final states: theoretical implications and experimental explorations of Richness of the Base. *In:* KAGER, René; PATER, Joe; ZONNEVELD, Wim. *Constraints in Phonological Acquisition*. Cambridge University Press, 2004, p. 321-368.

DEMUTH, Katherine. Markedness and the development of prosodic structure. *NELS* 25, 1995, P. 13-25.

FERREIRA-GONÇALVES, Giovana. Aquisição. In: BISOL, Leda; SCHWINDT, Luiz Carlos (orgs.). *Teoria da Otimidade*: *Fonologia*. Campinas: Pontes Editores, 2010, p. 231-270.

GARCIA, Guilherme Duarte. *A aquisição de acento primário em inglês por falantes de português brasileiro*. Dissertação (Mestrado em Letras). Porto Alegre: Universidade Federal do Rio Grande do Sul, 2012.

GNANADESIKAN, Amalia. Markedness and faithfullness constraints in child phonology. *In:* KAGER, René; PATER, Joe; ZONNEVELD, Wim (eds.). *Constraints in Phonological Acquisition*. Cambridge University Press, 2004, p. 73-108.

GUTIERRES, Athany. *Variação na aquisição fonológica: a produção da nasal velar em inglês (L2)*. Tese (Doutorado em Letras). Porto Alegre: Universidade Federal do Rio Grande do Sul, 2016.

HAYES, Bruce; TESAR, Bruce; ZURAW, Kie. *OT SOFT*. Disponível em http://www.linguistics.ucla.edu/people/hayes/otsoft/.

LEVELT, Clara C.; Van de VIJVER, Ruben. Syllable types in cross-linguistic and developmental grammars. *In:* KAGER, René; PATER, Joe; ZONNEVELD, Wim (eds.). *Constraints in Phonological Acquisition*. Cambridge University Press, 2004, p. 204-218.

MATZENAUER, Carmen Lúcia Barreto; NEUSCHRANK, Aline; CARNIATO, Miriam C.; QUINTANILHA-AZEVEDO, Roberta. Vogais em posição pós-tônica final: percepção e produção (no Sul do Brasil). *Revista da ABRALIN*, v. 14, n.1, p. 19-45, 2015.

MATZENAUER, Carmen Lúcia Barreto; QUINTANILHA-AZEVEDO, Roberta. Teoria Fonológica e Aquisição de Língua Estrangeira. In: ALVES, Ubiratã Kickhöfel (org.). *Aquisição fonético-fonológica de Língua Estrangeira*: *investigações Rio-Grandenses e Argentinas em discussão*. Campinas: Pontes Editores, 2016, p. 25-47.

NEUSCHRANK, Aline; MATZENAUER, Carmen Lúcia B.; LUZARDO, Javier E. S.; CARNIATO, Miriam C.; VAZ, Raquel Menezes; QUINTANILHA-AZEVEDO, Roberta. A formalização da assimetria da lateral em onset e em coda de sílaba no português dos campos neutrais pela OT Estocástica. *Alfa – Revista de Linguística*, v. 59, n.1, p. 181-203, 2015.

PATER, Joe; PARADIS, Johanne. Truncation without templates in child phonology. *In:* STRINGFELLOW, Andy; CAHANA-AMITAY, Dalia; HUGHES, Elizabeth; ZUKOWSKY, Andrea. (eds.). *Proceedings of the 20th Annual Boston Universal Conference on Language Development*. Somerville, Mass: Cascadilla Press, 1996, p. 540-551.

QUINTANILHA-AZEVEDO, Roberta. *A epêntese no Português Brasileiro (L2), em segmentos plosivos em codas mediais, por falantes nativos do Espanhol Colombiano (L1): uma análise via Teoria da Otimidade Estocástica e Gramática Harmônica*. Dissertação (Mestrado em Letras). Pelotas: Universidade Católica de Pelotas, 2011.

QUINTANILHA-AZEVEDO, Roberta. *Formalização fonético-fonológica da interação de restrições na produção e percepção da epêntese no Português Brasileiro e no Português Europeu*. Tese (Doutorado em Letras). Pelotas: Universidade Católica de Pelotas, 2016.

QUINTANILHA-AZEVEDO, Roberta; MATZENAUER, Carmen Lúcia Barreto; ALVES, Ubiratã Kickhöfel. A produção da vogal epentética no Português Brasileiro por colombianos: uma análise via Gramática Harmônica. *Organon*, v. 28, p. 217-239, 2013.

SCHMITT, Bruna Koch; ALVES, Ubiratã Kickhöfel. The acquisition of /p/ and /k/ word-mid codas of English (L2) by learners from Southern Brazil (L1): a gestural analysis in Stochastic Optimality Theory. *Letrônica*, v. 7, n. 2, p. 765-794, 2014.

SMOLENSKY, Paul. On the internal structure of the constraint component CON of UG. *Rutgers Optimality Archive* 86, 1995.

SMOLENSKY, Paul; LEGENDRE, Géralndine. *The Harmonic Mind – vols. 1 and 2*. MIT Press, 2006.

TESSIER, Anne-Michelle. *Biases and stages in phonological acquisition*. Tese (Doutorado em Linguística). Amherst: University of Massachusetts, 2007.