

Shruti Dube Y3337

Introduction

- Natural languages are characterized as a combination of rule-based generalization and lexical idiosyncracy
- The rules governing the syntax are actually quasi regular.
- Apparently allowed constructions are idiosycratically missing

Idiosyncracies

Baker's Paradox

The English past tense where the irregular form 'went' replaces '*goed*'

Dative Shift:

John gave/donated a book to the library John gave/*donated the library a book

• *'to be'* deletion rule

The baby appears/seems to be happy The baby appears/seems happy The baby seems to be sleeping *The baby appears/seems sleeping

Lexical constraints

strong/high/*stiff winds
strong/*high/*stiff currents
strong/*high/stiff breeze

Transitive/Intransitive

John broke the cup The cup broke John kissed Mary *Mary kissed

 We cannot interpret syntactic behaviour from semantics John waved Mary goodbye
 John waved goodbye to Mary
 *John said Mary hello
 John said hello to Mary

Question...

• Given these holes and quasi regular rules, what really is the learning mechanism ?

- Since only a finite set of sentences are heard, absence of a form does not imply that it is not allowed.
- Such holes are specific to languages so they cannot be attributed to postulating a Universal Grammar

Approach

- Learning quasi regular structures in a rudimentary language from positive evidence alone using the Simplicity principle.
- As a measure for simplicity, MDL (Minimum Description Length) was used:Kolmogorov complexity

- Simplicity :Cognitive Systems prefer simpler patterns over complex ones. The shortest program that regenerates the object is a natural measure of its complexity
- Instead of actual binary coding, codelengths are calculated for this purpose.
 - All grammars or hypotheses are expressed C = C(H)+C(D|H)

Simulation

2 approaches :

- There is a "super speaker" for all the listeners who knows the correct grammar entirely. All his utterances shall be completely correct and can have no exceptions
- Transmission over generations: The listener listens to sentences spoken by a speaker, who himself was the listener some time ago.

The first approach, though fair enough for simulation purposes, is unrealistic
The second is closer to reality.

It can be thought to be a model wherein a generation of parents instruct their successors about the exceptions in language. Children also posit exceptions on listening to their utterances Algorithmic details:

Rudimentary toy language:

S1 = AB S2 = BA $A=\{a1,a2,a3,a4\}$ $B=\{b1,b2,b3,b4\}$ * = { (a1),(a2,b2),(a2,b3),(a2,b4),(b1,a1), (b2,a1),(b3,a1),(b4,a1) }

- Grammars are represented by matrices of probabilities and an exception matrix.
- Words are ranked according to their frequencies. The speaker and the listener share the probabilities of these words.
- Using Zipf's law the probabilities are calculated: $p = f/\sum f$

 Learning proceeds by gambles. The listener gambles whether a sentence can be an exception depending on whether it satisfies :

Log(1/p(x)) < N p(x)

Scaling up of probabilities is done and if the gamble is correct, it results in shorter codelengths. If it is incorrect, the gamble is abandoned.

- Each time an exception is posited, a new hypothesis is generated
- A sentence which has been heard is never posited as an exception.
- The codelengths for these grammars are calculated.

Each hypothesis is exposed to 50 sentences of a grammar, which in the first approach are from the perfect grammar, and in the second from the previous grammar.

This attempts to mimic the situation of the poverty of stimulus, where one never hears all sentences and receive no negative feedback.

```
👎 ankua 🥲
```

Class Edit Touls Options

```
Convoile Undo G.1 Copy Pasts Find. Find Next Close
                                                                                                           Imp eme tat on
  import java. 10. 4 ;
  import jova.utal.Randoms
 public class accuals
  1
          static loolsan[]]] heards
          static granner [] array - new granner [:3]:
          static gramman actualS gramman = new gramman();
          final scatte int [1]] actuals except new int [11][3];
          conclustotac ant [1] sensetabergoses-new ant [3:][2];
          -nel stati-
 imt[][]=xvep =Luc.s={{2,3}, {4,3}, {5,3}, {7,3}, {1,1}, {1,2}, {1,3}, {2,1}, {4,1}, {5,1}, {7,1}};
         itatic int excep row-U:
         static int excep col-0;
         static blobesn []]] posited snew boolean[8][4];
         boolean [][] tried_for_rositing- new borlean[8][4];
      actual 3()
      ŧ
       for [1:: 1-0/148/1++]
         T.
              for (int. j-0;j-4;j-+)
              ×,
                  trisd_for_positing[1]["]-false;
              ł
         ł
      Presses as incurnent the gainer man and exceptions in prov graniar
      directions was of extendin convert grapper generated
```

👎 ankual 🤃

Class Edit Tulis Options

Contaile Undo C.1 Copy Fasts Find. Find Next Close

```
//iteration of inguineest thi grimes num and exceptions in prov granies
//iterations num of exception district granessing generated
public int generates granuars(int gran num,int exception)
//
```

ImportStreenReader isr- cen ImputStreenReader (Syster.cn): BufferelReader is= new EmfferedReader (Isr);

```
booleen excep encountrues
    int court-0;
    int y=0;
    int 1003
    int col;
    int actual loon;
    int act.al3.ul;
    int tearcol;
    int []] sents_typoicd_to= new int [s0,[3]:
    Anuitle grab of except
    double templai;
    double cemp2:
    double temps::
    double temp4-0;
    double temp5:
    double cemps:
    douric temp /-0:
   hoolean once done-felses
boolean [][] onle calculatei= naw blolean [5][4]:
Ranion remerator - new Randor();
 tor [int 1-0:1(8:11)]
   £
        Cor(int j=0; j<4; j-+)
        1
            oncc_colculated[1][3]=falses
```

Class compiled - no cyntax errors

Imp_eme_tation

7 anina ?!

Class Edit Tolls Options

```
Compile Undo C.1 Doby Fasts Find. Find Next Close
                                                                                                          Imp ementation
          boolean once doce-felses
      boolean [][] onle calculatei= naw blulean [5][4]:
     Ranion remerator - new Randor();
      tor [11: 1-0:1(8:11)]
          1
              Cor(int j=0; j<4; j-+)
                  omes_calculated[1][3]=false;
              ş,
      for (int i=laikeaill)
          1
              Ivr(int j=0;)<4;j++)</pre>
              1
                  tried for positing[1][1]-toloe:
              3
          1
          for ( int i-0;i<50;i+-)
         (
              do
L
                  actualScop = generatorinextInt( 8 );
                  tearcol - generator.nextInt( 10);
                  actualStol= array[gran_num11].probar_chrossing_actualErrow];tempco1]:
              while(rectual% grammarised word probjactualEcos]jactualEcol]==0);;
              array[gran num+1], indiheard[actual3rov][actual3col]-true;
              sents exposed to [i][0]-actual@rom;
              sents exposed to [i][1]=actual3uol;
```

👎 ankual 🤃

Class Edi Tulis Options

```
Contaile Lodo C.1 Doay Fasts Find. Find Next Close
int Locs
double revel:
double revel:
```

```
double totsum-0;
it (gray nux -- _)
ţ,
   Jun (_nt _ =0;1<50;1+-)</pre>
   T
       wor- sents_txpoted_to [1][U];
        loc = sents_exprsed_to [i][]];
       ssur = siray[0].set word prib[wid][lot]*siray[0].fit word prob[vor];
        zec = Nath.rov(ssum,-1);
        totoun - totoun | Noth.log(rec)/Math.log(2);
    E
   System.int.ci.ntlm"For the completely regular grammar codelength = "-t.rsun;;
   seturn (0);
$
  y/generates a postulate of a could be exception
  20
  Ł
   du
f
        if ( excep_exhibited;tried_for_positing);
    2
            System.out.prin.in("These are all the grammans that can be generated");
            Syntem.exit(0):
         撼
```

cou...+;

row - ginerator.nextInt(8);

```
lempcol = generator.nextInt( 10);
```

Imp ementation

```
Thunun F
Class Edit Tolls Options
 Compile Undo 0.1 Copy Fasts Find. Find Next Close
                                                                                                                 Imp ementation
                the second s
                trate for prosting[row_lcoi]=true;
           1
           while(istray)gram num).indiheari[roy][cil]--trus( | (positei'roy][cil]--trus();
           pros_of_excep-(inray_grim_num).tir_vord_pros_rov]; (array[gram_num,.sec_word_prob[rov][co_]);
           posited['ow][col]-true;
           if (once done--faise)
           a.
           Jos ( in. i=0/i<50;i++)</pre>
           É.
           act:allrom-sents_exposef_to[i][];
           sollai3loi=sents exposed to[i][1];
           . 1
                arrey[gran mm+1].ind heard[actual@ron[[actual@rol]]-true:
                once done-true;
                actualSrow-femts_exposes_to [1.[U];
                actual@indiatente_exposed_to_[1][1];
  1
               if ((row--altualSrow) &2 (col--actualEcol))
               ł.
                    excep dinser-folge:
           break/
                tengl=(hrray gram num).tir word proc.row] ? (array gram num, see word prob[row][tot_) ;
                read2-N=th.gom(temp1,-1):
                leur S=N=lh.log(teup2)/N=lh.log(2);
                tent4-Nath.tot((1-tamp1),-1);
                if (L(1))*terp_/temp:///
                ų
                    excep chosen=trie;
```

4 anina 9

Class Edil Tulls Options



```
7 anima ?!
```

Class Edit Touls Options

```
Compile Undo G.1 Copy Pasts Find. Find Next Close
                                                                                                         Imp eme tat on
        WALLC (TRUI) ;
          lor(int 1=0;1<8;1-+)
          r
              cor (int j=_:)X4:;[]]
              l,
                  srray[qran_num+1].fiz_word_prob[i]-stray[qran_num].fir_word_prob[1];
                  arriy[gram_num[1].sid_wind_prob[1][]]=array[gran_mum].sid_wind_prob[1][]];
                  armsy[gran_num+1].exceptodelength_armsy[gran_num].exceptodelength;
              1
         Ł
         sizey[gram com+1].set word prob[row][co1]=0;
        array[grar_mun]_..occ_word_prob[row][co_..0;
       for (int 1-0;1<8;1+-)
        4
            from (int. j=0;j<4;j+-)
            1.
               if (istray qram_num-1].set_word_prob[1][*]( -1)
                 Ł
                     (arrey[gran_num+1].sec_wood_prob[i][j](+((array[gran_num+1].sec_wood_prob[c][j])*(Math.prm(temp4,0.5)));
                 3
            ĩ.
      Ĵ,
      for [11: 1-0/1<8/1++]
      55
            istray[gram_brow+1].fit_mod_prob[i];=((atrag[gram_num+1].fit_rotd_ptoh[:])*(Narb.gor(temp4.0.5));
```

7 anina ?!

Class Edi Tulis Options

```
Conveile Undo C.1 Copy Fasts Find. Find Next Close
                                                                                                          Imp ementation
        int counter-U:
        int 5-0:
        if (gram :....==0)
          ſ
              arrey[gran_num+1].list_rf_excep_with_each_gran[[][0]-ror:
              array[1].list of extep with each gram[0][1]=col;
              for (int i-0; :< 70; i++)
              £
                  actual3rov-sents_exposei_to[1][0];
                  octualScol=cents_exposes_to[1][1];
                  if (once_calculated[actualOnom][actualOno]]--ture;
                  Ø
                      continus:
                  松
                  e' #*
                  È
                      once calculated[actual3rov'[actual3col'-true;
                      tcap1=Nith.pov((array[gram_cum[1],fit_wind_prob[ectio_3rev_)*[erray[gram_num1],.see word_prib[ectualSter][ectualSt
                      lemi2=(Xath.log(lemi1))/(Hath.log(2));
                      array[gran_num[1].datagadelangta =array[gran_num]__.datagadalength[temps:
                  $
              4
          System.sut.groutlou"datecodelength for granner no [+(green num-1) - is "terreg[green num+1].feterodelength;;
          tengl=Math.cov((array[gran_num'.fur_word_prob[rcw])*(array[gran_num].set_word_prob[rcw][col](,-1))
```

```
7 anima ?!
```

Class Edit Touls Options



```
Thunun F
Class Edil Touls Options
 Compile Undo C.1 Copy Pasta Find. Find Next Close
                                                                                                           Imp ementation
                           array[gran_num+1],sec_wind_inob[actual3row][actual3iol]-array[0],set_word_prob[actual3riw][actual3cil];
                           tempo=Nath.gov((array[0], far. word prob[actualSraw])*(array[1], see wird prob[actualSraw_LoctualSta_).( 1));
                            ____6=rXath.log(lem;5))/(Eath.log(2));
                           array[gran_num+1], exceptodelength = array[gran_num+1], exceptodelength-tamp6;
                           arrey[gram_cum+1]. list of exception to each gram[commtex:[0]-stray[gram_max=1]. list of except with each gram[excl
                           array[gram num+1], list if excep with each gram[counter][i]=array[gram num+1], list of extep with each gram[exch
                           array[gran_num+1].list_tf_excep_with_eath_gran[sxchangetounter[0]--1/
                           array[gram_num(1], 11st_tz_excep_with_each_gram[twohingetounter_[1]= 1/
                           exclangecourter++;
                           //stale down pulbebilities
                           for (inc 1-1:1:E:1++)
                           ¥
                               for ('nt k-':k:4:k++)
                                £
                                    arriy[gram_cumi1].sic_wind_prob[j][1]=(inray_gram_num_1].sci_word_prob[j][6](*(Asth.pow((__(arriy[U],
                                82
                           for (int 1-0;1-8;1-+)
                            ł
                               arrey[gram num+1].fin word prob[j]-erray[grem num-1].fin mord prob[j]=N="h.parr((1-)=rray[0].fin more preb
                           ¥
                       ंड
                    counter+-;
                ē.,
                array[clam_num+1].list of excep with each glam[couller]"0"slowy
```

```
Thunun F
Class Edil Touls Options
 Compile Undo C.1 Copy Pasta Find. Find Next Close
                                                                                                           Imp ementation
                           array[gran_num+1],sec_wind_inob[actual3row][actual3iol]-array[0],set_word_prob[actual3riw][actual3cil];
                           tempo=Nath.gov((array[0], far. word prob[actualSraw])*(array[1], see wird prob[actualSraw_LoctualSta_).( 1));
                            ____6=rXath.log(lem;5))/(Eath.log(2));
                           array[gran_num+1], exceptodelength = array[gran_num+1], exceptodelength-tamp6;
                           arrey[gram_cum+1]. list of exception to each gram[commtex:[0]-stray[gram_max=1]. list of except with each gram[excl
                           array[gram num+1], list if excep with each gram[counter][i]=array[gram num+1], list of extep with each gram[exch
                           array[gran_num+1].list_tf_excep_with_eath_gran[sxchangetounter[0]--1/
                           array[gram_num(1], 11st_tz_excep_with_each_gram[twohingetounter_[1]= 1/
                           exclangecourter++;
                           //stale down pulbebilities
                           for (inc 1-1:1:E:1++)
                           ¥
                               for ('nt k-':k:4:k++)
                                £
                                    arriy[gram_cumi1].sic_wind_prob[j][1]=(inray_gram_num_1].sci_word_prob[j][6](*(Asth.pow((__(arriy[U],
                                82
                           for (int 1-0;1-8;1-+)
                            ł
                               arrey[gram num+1].fin word prob[j]-erray[grem num-1].fin mord prob[j]=N="h.parr((1-)=rray[0].fin more preb
                           ¥
                       ंड
                    counter+-;
                ē.,
                array[clam_num+1].list of excep with each glam[couller]"0"slowy
```

```
Thunun F
Class Edit Touls Options
 Constile Undo C.1 Docy Fasts Find. Find Next Close
                                                                                                           Imp ementation
               3
               System. rut. rrintin("data cole langth for grammar number "+(gram num+1)+" is "+array gram num-1], datacolelangth);
             templ=Math.pox((array[grai min],fir 0.1d prob[low](*(siray[grain nim])see word prob[row][cri]),+1);
             temp3- Hath.ltg[temp1)/[Nath.log(2)];
             array[gran_nunl__.exceptorelength = array[gran_nunl__.exceptorelength]tear3;
             Syster.out.printl:("exception code 'ength for granner number "-upper num+1;+" is "-engag[gran num+1].exceptorfelength):
             druble sur - Erray[qram_num-1],exceptidelength+array[rram_num+1].istaiodelength)
             System.out.println("Ine code length for gronnin number "[(gran numl_)]" if "[cum];
         1
         Syster.out.println("Inis growner contains " (counter v1_) ["exceptions");
         if gran num -- 5 t
         4
             τey
                 hr. readhire();
             datch (10Exception e) () /
        Syster.out.printl"();
        felum: (clumter-v+1);
         )
       public static wild main (String Ergs[]]
       $
           Tor (int 1=0;1<32:_++)
```

Class Edil Touls Options

7

```
Contaile Undo C.1 Coax Fasts Find. Find Next Close
                                                                                                          Imp ementation
          actualS granmarised word prib[7][3]=0;
          actual3_qranmar.sec_wori_prob[3][3]-0;
          octuol3_granmar.sec_wor1_prib[1].2.-U:
          actual@_granmar.sec_worf_prob[1][0]-0;
          actual3 graimar.sec word prib[6][2]=0;
          actual3_grammar.sec_wor2_prib(4)[2]-0;
          actual3_granmar.sec_word_prob[5].2.=0;
         ecture10 granzar.sec_word_prov[7][1]-1:
          actual3 obj= nex actual3();
          int count--1:
          int exceps-.:
           tht h-T:
         ALL: (Crue)
          E
              b = obj.generates granning(count.exceps);
             exceps-'ra
          vou:..+;
          1
         //doing the granier and the indeptions hiw
      public hoolsam except exhausted hoolsam[][] htt
          cor(int 1-0:1-8:1 1)
          1
              Jus(int j=0;)<4;j→)</pre>
              ſ
                  it ( b[i][:]--tcloc)
                      Let.in falses
              ų
          set.in time;
      Т
```

Cptions

For the completely regular graniar codelength = 256.034F669F302F85 datacodelength for graniar no 1 is F5.7E83178081608F except orde length = 5.1863F581E833F04 This granuar contains largegraphs

data cole length for grannar number 2 18 93.73568554141206 exception code length for grannin number 2 18 11.451475536268673 The code length for grannar number 2 is 105.17716207770774 This grannar contains 2-sceptions

data cone langth for grammar number 3 10 9.,7057.57794206 exception code length for grammer number 3 is 10.214051015596725 The code length for grammar number 5 is 117.92057205501788 This grammar contains Sexceptions

date code length for grammar number 4 is 95.07407912098654 exception code length for grammar number 4 is 24.641359161518028 The code length for grammar number 4 is 120.01548901551457 That grammar contains 45xceptions

date code length for grammar number 5 is 75.63553003391694 exception code length for grammar number 5 is 30.636762938224553 The code length for grammar number 5 to 106.27229297214149 This grammar contains Secondars

date cole length for grammar number 6 18 87.20427259502941 exception code length for grammar number 6 10 28.52304/727566155 The code length for grammar number 1 to 115.72712077269555 This grammar contains 5exceptions

📽 BlueD: Tempinal Mudou - Evolution

op.ion=

data rode length for granner number 7 18 86.36243831764393 exception code length for granner number 7 15 35.83/1324/50857 The orde length for granner number 7 is 122.15960769252964 This granner contains devieptions

dita code length for granner number = 10 8..4132/2606879.9 exception code "ength for granner number 0 i= 00.440114224907.566 The olde length for granner number 8 is 119.85138582751555 This granner contains desceptions

dena node length for grannat tumber: 3 to 76.1520(575)00579. exception code length for grannar number 9 is 45.41153117057637 The code length for grannar number 9 is 121.76399692746235 This grannar contains Veroeptions

dela Jode Length für grannar number 10 is 92.57351864183541 exception code Length för grannar number 10 is 51.36554154245792 The olde length för grannar number 10 is 143.13735048429758 This grannar contains Desceptions

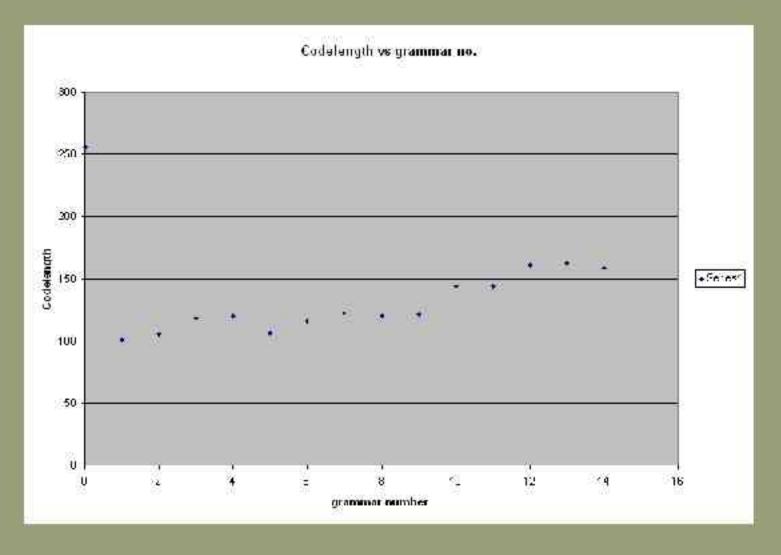
data code length for grannar number 11 18 85.93858942294727 exception code length for grannar number 11 is 140.71507796551673 The orde length for granner number 11 is 140.71507796551673 This grannar contains Sexueptions

data code length for granner number 12 is 97.37573635060514 exception code length for granner number 12 is 67.0744.5700575927 The cide length for granner number 12 is 160.55015195114138 This granner contains loexceptions

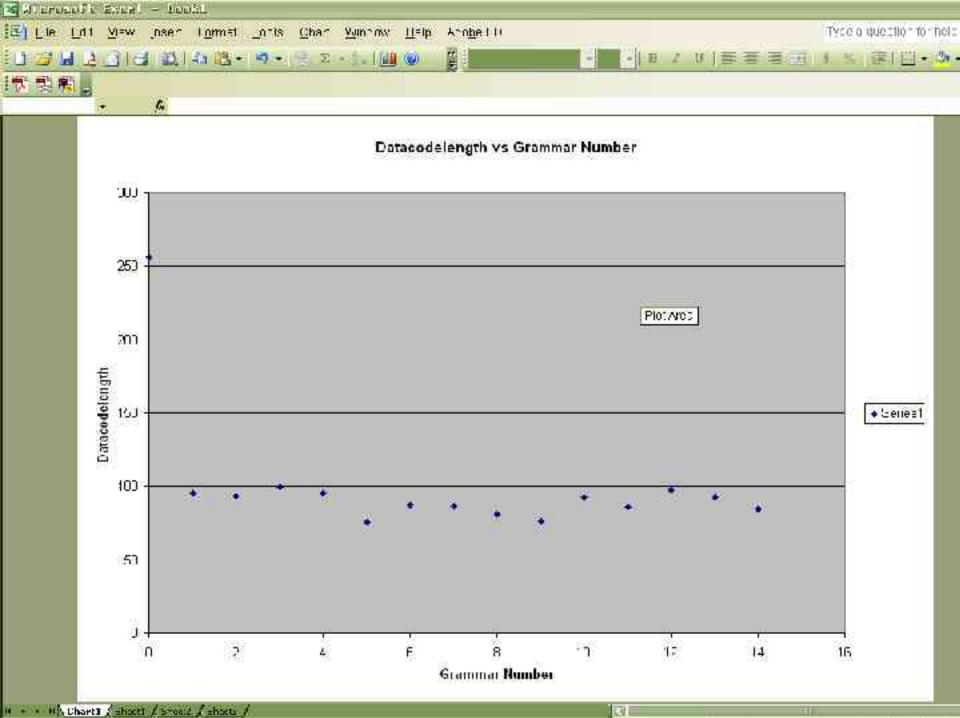
deta tode length for granner turber 10 is 92,9095545140651 exception code length for granner number 13 is 65,66969419495495 The dide length for granner number 13 is 162,50874875583016 Juis granner contains lexceptions

dena Lode Length for grannar number 14 is 83.39950175048208 exception code length for grannar number 14 is 74.94548453214615 The code length for granning number 14 is 188.54595572552853 This grannar contains ligesceptions





131



op.ion=

Fin the completely regular grammar codelength = 257.7184350643798 detactdelength for grammar no 1 is 93.60852932947283 except codd length = 5.671288519054461 This meaner contains lexieptions

data code length for grannar number 2 is 86.95130988534552 exception code length for grannar number 2 is 9510570679067902 The orde length for granner number 2 is 9510570679067902 This grannar contains 2exceptions

data code length for granner number = 10-102.661.7855336505 exception code "ength for granner number 0 i= "7.702560050"0140 The cide length for granner number 3 is 120.44493545649654 This granner contains Sexceptions

deta tode length for granner number: 4 to 95,40701050430771 exception code length for granner number: 4 is 24,600344321025714 The dide length for granner number 4 is 120,01816290541603 Juis granner contains 4exceptions

deta tode length for granner number 5 is 82.41085617967458 exception code length for granner number 5 18 29.597523851965336 The code length for granner number 5 15 112.01837.98184091 This granner contains Jerceptions

data node length for grammar number 5 is 86.93879680495813

op.ion=

dela lode length for grannar number 7 is 92.23942065582151 exception code length for grannar number 7 is 42.90342523053584 The orde length for grannar number 7 is 1:0.143845887:4844 This grannar contains 7exceptions

data code length for grannar number 6 is 78.16945049503652 exception code length for grannar number 8 it 49.14/194252485696 The orde length for granner number 0 is 177.01667476171527 This grannar contains Sexueptions

dita code length for grannar number 5 to 85.52545501558758 exception code length for grannar number 9 is 55.706106440001006 The cide length for granner number 9 is 141.0122714585937 This grannar contains Sexceptions

deta tode length for granner turber 10 is 90.06917021071806 exception code length for granner number 10 is 61.674970881688904 The code length for granner number 10 is 155.54414903837756 This granner contains lUexception:

deta tode length for grannar number 11 is 102.81559228447402 exception code length for grannar number 11 is 65.95357516026125 The code length for granning number 11 is 171.76916724473829 This grannar contains l'exceptions

These are all the granmars that can be generated

The Second Approach:-

For 20 sentences:number of exceptions are very large

📽 BlueD: Tempinal Mindon - Evolution

= nui.qC

This granger contains Sexceptions

data onde length for granner number 17 ta 62.72007700767640 exteption tode length for grannar number 10 is 49.61848301549387 The coie length for grannar number 10 is 111.8385310961698 This grannin contains lUckseptions

data olde length fin gramman number 11 is 52.54200867588234 exception code length for grammar number 11 is 54.43410603594757 the code length for grammar number 11 is 115.975.0.7045303s This grammar contains liexceptions

data code length for granner number 12 is 56.91531278000918 exception code length for granner number 12 is 59.78258127644549 The code length for granner number 12 is 116.6570940554355 This granner contains 12exceptions

data dade length for grannin number 1: 10 51.7354:6835403:8 exception code length for grannar number 10 is 65.90255907570107 The code length for grannar number 13 is 117.63798572050475 This grannar contains lexceptions

data dade length for granner number 14 to 49.5002502550537 exception code length for grannar number 14 is 72.19428972893563 The code length for grannar number 14 is 121.7745779797495 This granner contains 14exceptions

data cide length fil gramman number 15 is 44.175459795066565 exteption code Length for gramman number 15 is 75.46748587920563 The code lingth for gramman number 15 is 119.5429556722726 This gramman contains Liewceptions

data code length for granner number 15 is 49.42905379477352

For 30 sentences: number of exceptions have decreased

🐔 Digel: Temptual Monday - Evolution

op.ion=

date cide length for granmin number 4 15 05.71389:37155035 extention code length for granmar number 4 to 20.79221777275196 The code length for granmar number 4 is 119.00583054512575 This granmar contains Mexceptions

data orde length for granner number 5 is 77.07900700252094 exteption tode length for grannar number 5 is 29.249359214837683 The code length for grannar number 5 is 115.33537629775161 This grannin contains texcoptions

data olde length fil grannel number 6 is 52.25458117438075 exception code length for granger number 5 is 34.18302986578952 This core lingth for granner number 5 is 115.447511.4417067 This granner contains fexceptions

data code length for grannin number 7 is 55.67229.19756807 extention code length for grannar number 7 cs 41.40021074740016 The code length for grannar number 7 is 125.15506254007623 This grannar contains 7exceptions

data dade length for grannen number 0 is 75.01074715629477 exception code length for grannar number 5 is 48.58267811030951 The code length for grannar number 8 is 133.94541626660433 This grannar contains eccoptions

data ulde length fli grammat number 9 is 51.75925340278805 exteption tode length for grammat number 5 is 54.05355976453455 The code lingth for grammar number 9 it 135.8228431673229 This grammat contains Sext-ptions

data code length for granner number 15 is 88.76174946141528 exception code length for granner number 10 is 59.3812159217547 The code length for granner number 10 is 140.04296739316994 This granner is plains 10es and ions

For 40 sentences : very few grammars are generated

🚰 BlueD: Tempinal Mindon - Evolution

Op.ion=

This granner cintains lisexceptions

data onde length for granner number 14 to 49.5002502550537 exteption tode length for grannar number 14 is 72.19428972393563 The code length for grannar number 14 is 121.7745779797495 This grannin contains 14exceptions

data cide length fit grammin number 15 is 44.175459793066565 exception code length for grammar number 15 is 75.46748587930563 the code length for grimmar number 15 is 119.5439556726726 This grammer contains Liexceptions

data code length for granner number 16 ts 49.42905379477352 exception code length for granier number 16 is 80.99819875469.63 The code length for granner number 16 is 170.42725254946417 This granner contains 16exceptions

https inc ii. the grannond that can be generated datacoleleryth for grannon no 1 is 119.7001044109617 extep tode length = 5.671288317054461 This granner contains lexceptions

data orde length for granner number 2 is 116.00765051759214 exception code length for grannar number 2 is 10.144059735828591 The code length for grannar number 2 is 125.81175324442073 This grannar contains eccoptions

data ulde length fli grannel number 3 is 119.5038581252565 exception code Length for grannar number 5 is 15.55034911792554 The core length for grannar number 3 is 135.45423724004985 This granner contains feaceptions

data code length for grannar number 4 is 108.16905940043422 exception code length for grannar number 4 is 19.507174665061592 The code length for grannar number 4 is 22.67126456045561 This granner is plains deviantions

With a larger corpus of data: approach 1

🚭 Bluels Terminal Mundos - Evolution

= nui.qC

cone length for greannarl num 6. 13 530.364/613.64644 This granner " contains 15nd of excepts

code lingth for grimmari num 61 is 592.0.01581507878 This gramment contains 1505 of except

cone langth for grammarl run 6d no 547.1505156505143 This grammard contains 14nd of except

cone lingth for grimmarl num 6: 13 561.3367596187173 "Dis granne"' conteins 14ac of except

Code lingth for grimmarl num 64 13 590.5195070145175 This grammer' contains 1505 of except

core lingth for grammarl num 65 10 575.7822337793415 "Dis grammer" contains 1501 of except

cone lingth for grimmarl num 65 is 595.161:7231126.1 This greeners' contains 15nd of except

cone length for grammari num 67 10 555.360.868:310.1 This grammer' contains 1407 of except

Code langth for grannari num 65 13 572.771781.97618 "Dis granne"' conteins 14oc of except

Larger corpora : approach 2

🚰 BlueD: Tempinal Mindon - Evolution

=nui.qC

Cone length for grammari num 3: 15 761.0205518136458 "Dis grammer" contains 2007 of excepts

code lingth for grimmarl num 3, is 767.06746414779.3 This gramment contains 39nd of except

core length for grammarl mum 4, 10 787.365.906595456 This grammard contains 400c of except

cone lingth for grimmarl num 41 is 795.0.2.974:187.9 "Dis granne"' contains 4'nd of exceps

Code lingth for grimmarl num 4: 18 731.5.55367:73554 "Did grammer" contains 42nd of except

core length for grammarl num 4: 10 795.797.339.0.9.2 "Dis grammer" contains 43mc of except

cone lingth for grimmari num 44 is 813.60118481960:3 This greener: contains 44er of except

cone length for grimmari num 45 10 839.340:808:60143 This grammer' contains 45mm of except

cone length for greanart num 45 to 839,4455332153857 This granner contains 46nt of except

Conclusion:

- When a small corpus of data was chosen, the simplicity principle yielded the exact number of 11 exceptions in most cases.
- The exceptions obtained were the same as those found in the actual language.
- So, for a small corpus of data, given the "super speaker approach", the simplicity principle works quite efficiently.

- However, if we look at the second approach. When we were using 50 sentences approximately 6 exceptions were found.
 - If the number of sentences uttered are decreased, the exceptions possible increase.
 - It is true that the datacodelegth decreases as we increase the number of exceptions. Since coding these exceptions requires some bits as well, whether that investment can be reobtained remains questionable.

- When the number of sentences heard were approximately equal to the number of sample sentences, nearly perfect results were obtained.
- With the increased sample size however, this fails badly.
- The codelengths show a monotonic increase

- Even grammars with 11 exceptions never had the same exceptions as were present in the grammar.
- This happens primarily because data is chosen probabilistically. Since the exceptions themselves are randomly decided, investment in coding them may be quite large itself.

- Since this language has no semantics and no communicative function, it doesn't model the relation between meaning-signal-referents.
- Though it is true for small corpora of sentences, it cannot be established as a general learning mechanism.

Thank you!