Language is a Complex Thing

For Children: You will need to know the difference between Friday and a fried egg. It's quite a simple difference, but an important one. Friday comes at the end of the week, whereas a fried egg comes out of a chicken. Like most things, of course, it isn't quite that simple. The fried egg isn't properly a fried egg until it's been put in a frying pan and fried. This is something you wouldn't do to a Friday, of course, though you might do it *on* a Friday. You can also fry eggs on a Thursday, if you like, or on a cooker. **It's all rather complicated**, but it makes a kind of sense if you think about it for a while.

Douglas Adams (2002) *The Salmon of Doubt*

The mirage of morphological complexity

Fermín Moscoso del Prado Martín
fermosc@gmail.com

Laboratoire Dynamique du Langage (UMR – 5596)
Centre National de la Recherche Scientifique, Lyon &
Institut Rhône-Alpin des Sytèmes Complexes, Lyon

UCSD, San Diego, January 15, 2011

Overview

- Yet another stand in the "measures of linguistic complexity" bazaar

Overview

- Yet another stand in the "measures of linguistic complexity" bazaar
- Of course, I'll try to convince you that mine is THE ONE

Overview

- Yet another stand in the "measures of linguistic complexity" bazaar
- Of course, I'll try to convince you that mine is THE ONE
- Part I: What is linguistic complexity?

Overview

- Yet another stand in the "measures of linguistic complexity" bazaar
- Of course, I'll try to convince you that mine is THE ONE
- Part I: What is linguistic complexity?
- Part II: What is morphological complexity? Can we isolate morphological complexity?

What I'm trying to sell you (it's also cheap!)

- Entropy, by itself, is the wrong measure of complexity

What I'm trying to sell you (it's also cheap!)

- Entropy, by itself, is the wrong measure of complexity (do not worry, no evil mind has taken over me)

What I'm trying to sell you (it's also cheap!)

- Entropy, by itself, is the wrong measure of complexity
(do not worry, no evil mind has taken over me)
- The complexity of a language is the length of the shortest possible description of its structure

What I'm trying to sell you (it's also cheap!)

- Entropy, by itself, is the wrong measure of complexity (do not worry, no evil mind has taken over me)
- The complexity of a language is the length of the shortest possible description of its structure
- One can measure this without having a clue about the actual structure

What I'm trying to sell you (it's also cheap!)

- Entropy, by itself, is the wrong measure of complexity (do not worry, no evil mind has taken over me)
- The complexity of a language is the length of the shortest possible description of its structure
- One can measure this without having a clue about the actual structure
- It does not seem possible to achieve a complete and correct description of a language

What I'm trying to sell you (it's also cheap!)

- Entropy, by itself, is the wrong measure of complexity (do not worry, no evil mind has taken over me)
- The complexity of a language is the length of the shortest possible description of its structure
- One can measure this without having a clue about the actual structure
- It does not seem possible to achieve a complete and correct description of a language
- 'Morphology by itself': NO WAY, rather, morphology when it is useful!

What I'm trying to sell you (it's also cheap!)

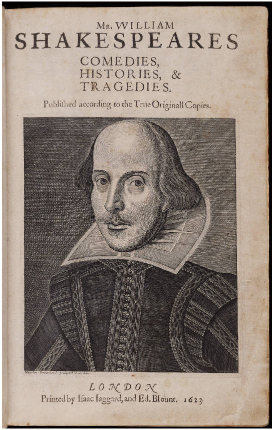
- Entropy, by itself, is the wrong measure of complexity (do not worry, no evil mind has taken over me)
- The complexity of a language is the length of the shortest possible description of its structure
- One can measure this without having a clue about the actual structure
- It does not seem possible to achieve a complete and correct description of a language
- 'Morphology by itself': NO WAY, rather, morphology when it is useful!
- There does not seem to be much variability in the morphological complexity of languages

What is more complex? Candidate 1

My nasty alarm clock

"... tik tak tik tak ..."

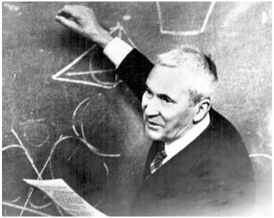
What is more complex? Candidate 2



William Shakespeare

"...Though this be madness, yet there is method in 't..."

Algorithmic Information Content (AIC)



Andrey N. Kolmogorov

The complexity of a string corresponds to the length of the shortest program that can reproduce the string. In other words, the complexity of a string is the minimal size to which it can be compressed.

(1965. Three approaches to the quantitative definition of information. *Problemy Peredachi Informatsii* **1**, 3–11)

Algorithmic Information Content (AIC)

- The alarm clock sequence can be compressed to a very short program: *repeat "tik tak" forever*

Algorithmic Information Content (AIC)

- The alarm clock sequence can be compressed to a very short program: *repeat "tik tak" forever*
- However, the compressibility of Shakespeare's plays is limited

Algorithmic Information Content (AIC)

- The alarm clock sequence can be compressed to a very short program: *repeat "tik tak" forever*
- However, the compressibility of Shakespeare's plays is limited
- AIC would therefore conclude that the complexity of Shakespeare's plays is higher than that of the sequence produced by an alarm clock. This seems right

What is more complex? Candidate 3

Bill Pearshaker

"...asljoewf ewliwejd 13je1dm 1kp..."

Algorithmic Information Content (AIC)

- The alarm clock sequence can be compressed to a very short program: *repeat "tik tak" forever*
- However, the compressibility of Shakespeare's plays is limited
- AIC would therefore conclude that the complexity of Shakespeare's plays is higher than that of the sequence produced by an alarm clock. This seems right

Algorithmic Information Content (AIC)

- The alarm clock sequence can be compressed to a very short program: *repeat "tik tak" forever*
- However, the compressibility of Shakespeare's plays is limited
- AIC would therefore conclude that the complexity of Shakespeare's plays is higher than that of the sequence produced by an alarm clock. This seems right
- The symbol sequence produced by the typing monkey is completely random, no symbol contains any predictive information about the others. It cannot be compressed at all.

Algorithmic Information Content (AIC)

- The alarm clock sequence can be compressed to a very short program: *repeat "tik tak" forever*
- However, the compressibility of Shakespeare's plays is limited
- AIC would therefore conclude that the complexity of Shakespeare's plays is higher than that of the sequence produced by an alarm clock. This seems right
- The symbol sequence produced by the typing monkey is completely random, no symbol contains any predictive information about the others. It cannot be compressed at all.
- For sequences of equal length, the monkey's output will require a longer description than an equivalent sample of Shakespeare's work.

Algorithmic Information Content (AIC)

- The alarm clock sequence can be compressed to a very short program: *repeat "tik tak" forever*
- However, the compressibility of Shakespeare's plays is limited
- AIC would therefore conclude that the complexity of Shakespeare's plays is higher than that of the sequence produced by an alarm clock. This seems right
- The symbol sequence produced by the typing monkey is completely random, no symbol contains any predictive information about the others. It cannot be compressed at all.
- For sequences of equal length, the monkey's output will require a longer description than an equivalent sample of Shakespeare's work.
- Is the monkey typing something really more complex than any of Shakespeare's plays?
  This does not seem right

Effective Complexity (EC)

Murray Gell-Mann

A measure that corresponds much better to what is usually meant by complexity in ordinary conversation, as well as in scientic discourse, refers not to the length of the most concise description of an entity (which is roughly what AIC is), but to the length of a concise description of a set of the entity's regularities.

(1995. What is complexity? *Complexity*, **1**, 16–19.)

Effective Complexity (EC)

- The alarm clock regularities can still be well compressed:
  "tak" follows "tik"

Effective Complexity (EC)

- The alarm clock regularities can still be well compressed: *"tak" follows "tik"*
- The monkey's output has no regularities whatsoever: Its complexity is zero

Effective Complexity (EC)

- The alarm clock regularities can still be well compressed: *"tak" follows "tik"*
- The monkey's output has no regularities whatsoever: Its complexity is zero
- Considering the fine regularities in Shakespeare's plays, which will take some place to detail (and even some full faculties), we can safely conclude that:

Effective Complexity (EC)

- The alarm clock regularities can still be well compressed:
  "tak" follows "tik"
- The monkey's output has no regularities whatsoever: Its complexity is zero
- Considering the fine regularities in Shakespeare's plays, which will take some place to detail (and even some full faculties), we can safely conclude that:
  Shakespeare's plays are more complex than either the noises produced by an alarm clock, or the texts a monkey would type.

Effective Complexity (EC)

Murray Gell-Mann

Thus something almost entirely random, with practically no regularities, would have effective complexity near zero. So would something completely regular, such as a bit string consisting entirely of zeroes. Effective complexity can be high only a region intermediate between total order and complete disorder.

(1995. What is complexity? *Complexity*, **1**, 16–19.)

EC of Language

- EC The complexity of an entity is the length of the most compact description of its regularities.

EC of Language

- EC The complexity of an entity is the length of the most compact description of its regularities.
- The 'regularities' present in language is what we usually term grammars

EC of Language

- EC The complexity of an entity is the length of the most compact description of its regularities.
- The 'regularities' present in language is what we usually term grammars
- Therefore, the complexity of a language is the length of the minimal grammar (in whichever grammatical paradigm) that is necessary to describe it.

EC of Language

- EC The complexity of an entity is the length of the most compact description of its regularities.
- The ‘regularities’ present in language is what we usually term grammars
- Therefore, the complexity of a language is the length of the minimal grammar (in whichever grammatical paradigm) that is necessary to describe it.
- Sure but, can one measure what is the minimal grammar length? We are having problems even in agreeing on whether one grammar is or is not adequate, let alone the best one?

EC of Language

- EC The complexity of an entity is the length of the most compact description of its regularities.
- The 'regularities' present in language is what we usually term grammars
- Therefore, the complexity of a language is the length of the minimal grammar (in whichever grammatical paradigm) that is necessary to describe it.
- Sure but, can one measure what is the minimal grammar length? We are having problems even in agreeing on whether one grammar is or is not adequate, let alone the best one?
  I will try to sell you the idea that it is possible to estimate the length of the minimal grammar, without actually knowing anything about it

Practicalities

- I have ignored another important question:

Practicalities

- I have ignored another important question:
  What is a language?

Practicalities

- I have ignored another important question:
  What is a language?
- Instead of an abstract definition, let's start from a tangible object, a corpus, in line with the pre-generative tradition of Zellig Harris (and modern followers such as Geoff Sampson)

Practicalities

- I have ignored another important question:
  What is a language?
- Instead of an abstract definition, let's start from a tangible object, a corpus, in line with the pre-generative tradition of Zellig Harris (and modern followers such as Geoff Sampson)
- I consider a reference corpus of L characters (for arbitrarily large L)

Practicalities

- I have ignored another important question:
  What is a language?
- Instead of an abstract definition, let's start from a tangible object, a corpus, in line with the pre-generative tradition of Zellig Harris (and modern followers such as Geoff Sampson)
- I consider a reference corpus of L characters (for arbitrarily large L)
- Let $G(L)$ be the size of the minimal grammar that can generate all sentences in the corpus, and only those. $G(L)$ is the EC of the corpus.

Practicalities

- I have ignored another important question:
  What is a language?
- Instead of an abstract definition, let's start from a tangible object, a corpus, in line with the pre-generative tradition of Zellig Harris (and modern followers such as Geoff Sampson)
- I consider a reference corpus of L characters (for arbitrarily large L)
- Let $G(L)$ be the size of the minimal grammar that can generate all sentences in the corpus, and only those. $G(L)$ is the EC of the corpus.
- Let $H(L)$ be the size of the most compressed possible version of the corpus. $H(L)$ is the AIC of the corpus.

Formulation (I)

- $G(L)$ needs to generate every sentence in the corpus.

Formulation (I)

- $G(L)$ needs to generate every sentence in the corpus.
- $H(L)$ needs to reconstruct the full corpus (including the ordering and frequency of the sentences).

Formulation (I)

- $G(L)$ needs to generate every sentence in the corpus.
- $H(L)$ needs to reconstruct the full corpus (including the ordering and frequency of the sentences).
- Therefore:

$$H(L) = G(L) + \Delta(L) \;\rightarrow\; G(L) = H(L) - \Delta(L)$$

$$H(L), G(L), \Delta(L) \geq 0$$

Formulation (I)

- $G(L)$ needs to generate every sentence in the corpus.
- $H(L)$ needs to reconstruct the full corpus (including the ordering and frequency of the sentences).
- Therefore:

$$H(L) = G(L) + \Delta(L) \rightarrow G(L) = H(L) - \Delta(L)$$

$$H(L), G(L), \Delta(L) \geq 0$$

- $\Delta(L) \geq 0$ is the information contained by the particular ordering and frequencies of the individual sentences

Formulation (II)

We can also think in terms of rates, units of complexity per character in the corpus:

Formulation (II)

We can also think in terms of rates, units of complexity per character in the corpus:

- Grammatical density of the corpus

$$g(L) = \frac{G(L)}{L},$$

Formulation (II)

We can also think in terms of rates, units of complexity per character in the corpus:

- Grammatical density of the corpus

$$g(L) = \frac{G(L)}{L},$$

-

$$h(L) = \frac{H(L)}{L}$$

Formulation (II)

We can also think in terms of rates, units of complexity per character in the corpus:

- Grammatical density of the corpus
$$g(L) = \frac{G(L)}{L},$$
- $$h(L) = \frac{H(L)}{L}$$
- $$\delta(L) = \frac{\Delta(L)}{L}$$

Formulation (II)

We can also think in terms of rates, units of complexity per character in the corpus:

- Grammatical density of the corpus
$$g(L) = \frac{G(L)}{L},$$
- $$h(L) = \frac{H(L)}{L}$$
- $$\delta(L) = \frac{\Delta(L)}{L}$$
- And we can reconstruct the equality
$$G(L) = H(L) - \Delta(L) \ \rightarrow \ g(L) = h(L) - \delta(L)$$

Formulation (III)

We can now generalize to infinite corpus size, that is, to consider every possible sentence that could eventually occur in the language:

Formulation (III)

We can now generalize to infinite corpus size, that is, to consider every possible sentence that could eventually occur in the language:

- Grammatical complexity of the language

$$G = \lim_{L \to \infty} G(L) = \lim_{L \to \infty} [H(L) - \Delta(L)]$$

Formulation (III)

We can now generalize to infinite corpus size, that is, to consider every possible sentence that could eventually occur in the language:

- Grammatical complexity of the language

$$G = \lim_{L \to \infty} G(L) = \lim_{L \to \infty} [H(L) - \Delta(L)]$$

- Grammatical density of the language

$$g = \lim_{L \to \infty} g(L) = \lim_{L \to \infty} [h(L) - \delta(L)]$$

Formulation (IV)

- If the language has a finite grammar size, then for a sufficiently large corpus, the grammar should be complete:

$$G = \lim_{L \to \infty} G(L) < \infty$$

Formulation (IV)

- If the language has a finite grammar size, then for a sufficiently large corpus, the grammar should be complete:

$$G = \lim_{L \to \infty} G(L) < \infty$$

- But notice that this implies that

$$g = \lim_{L \to \infty} g(L) = 0$$

Formulation (IV)

- If the language has a finite grammar size, then for a sufficiently large corpus, the grammar should be complete:

$$G = \lim_{L \to \infty} G(L) < \infty$$

- But notice that this implies that

$$g = \lim_{L \to \infty} g(L) = 0$$

- A condition for a finite grammar to exist for a language is that its grammatical density is zero. If $g > 0$, no finite grammar can describe the language without over- or under-generating.

Formulation (V)

In the infinite corpus size limit, the measures g, h, and δ have clear interpretations

- Kolmogorov-Sinai entropy of the language

$$0 \leq h = \lim_{L \to \infty} h(L) < \infty$$

Amount of entropy production per character (uncertainty of the next character in a sequence, provided an infinitely long history is known).

Formulation (V)

In the infinite corpus size limit, the measures g, h, and δ have clear interpretations

- Kolmogorov-Sinai entropy of the language

$$0 \leq h = \lim_{L \to \infty} h(L) < \infty$$

  Amount of entropy production per character (uncertainty of the next character in a sequence, provided an infinitely long history is known).

- Kolmogorov-Sinai entropy of a modified language

$$0 \leq \delta = \lim_{L \to \infty} \delta(L) < \infty$$

  K-S entropy of a corpus in which each sentence has been replaced by an individual symbol, divided by the mean length of the sentences in characters.

Formulation (VI)

- Grammatical density of the language:

$$g = h - \delta$$

Formulation (VI)

- Grammatical density of the language:

$$g = h - \delta$$

- *Theorem:* a finite grammar for a language exists if and only if

$$h = \delta$$

h: Kolmogorov-Sinai entropy of the corpus

- (Lempel & Ziv, 1976): 'Parse' a string:

$$1001101110010100100$$

into

$$1 \cdot 0 \cdot 01 \cdot 101 \cdot 1100 \cdot 1010 \cdot 00100$$

h: Kolmogorov-Sinai entropy of the corpus

- (Lempel & Ziv, 1976): 'Parse' a string:

  $$10011011100101000100$$

  into

  $$1 \cdot 0 \cdot 01 \cdot 101 \cdot 1100 \cdot 1010 \cdot 00100$$

- N_w is the number of words in the parse, N is the length of the original string

h: Kolmogorov-Sinai entropy of the corpus

- (Lempel & Ziv, 1976): 'Parse' a string:

$$1001101110010100010 0$$

into

$$1 \cdot 0 \cdot 01 \cdot 101 \cdot 1100 \cdot 1010 \cdot 00100$$

- N_w is the number of words in the parse, N is the length of the original string
- Lempel-Ziv Complexity:

$$L_N = \frac{N_w}{N} \ln N$$

h: Kolmogorov-Sinai entropy of the corpus

- Ziv & Lempel (1978) proved that (if the sequence is stationary):

$$\lim_{N \to \infty} L_N = h$$

h: Kolmogorov-Sinai entropy of the corpus

- Ziv & Lempel (1978) proved that (if the sequence is stationary):

$$\lim_{N \to \infty} L_N = h$$

- The convergence of the algorithm is very fast (Lesne et al., 2009)

h: Kolmogorov-Sinai entropy of the corpus

- Ziv & Lempel (1978) proved that (if the sequence is stationary):

$$\lim_{N \to \infty} L_N = h$$

- The convergence of the algorithm is very fast (Lesne et al., 2009)
- Schurmann & Grassberger (1996) and Moscoso del Prado (2010) found that the convergence is well-modelled by

$$L_N \approx h + a\, N^{-b} \ln N, \quad b > 0$$

which can be fitted from corpora of different sizes to estimate h.

δ: K-S entropy of a modified corpus

- Consider the corpus as a sequence of sentences $C = s_1 s_2 \ldots s_S$, and record the average sentence length L_S.

δ: K-S entropy of a modified corpus

- Consider the corpus as a sequence of sentences $C = s_1 s_2 \ldots s_S$, and record the average sentence length L_S.
- The entropy rate of C divided by L_S is δ.

δ: K-S entropy of a modified corpus

- Consider the corpus as a sequence of sentences $C = s_1 s_2 \ldots s_S$, and record the average sentence length L_S.
- The entropy rate of C divided by L_S is δ.
- Chao & Shen (2004) developed an estimator for that entropy

$$H_s^{\text{C-S}}(S) = -\sum_{s \in C} \frac{\tilde{p}(s) \ln \tilde{p}(s)}{1 - [1 - \tilde{p}(s)]^S}$$

$\tilde{p}(s)$: Good-Turing adjusted probability of each sentence

$$\tilde{p}(s) = \left(1 - \frac{f_1}{S}\right) \frac{f(s)}{S}$$

δ: K-S entropy of a modified corpus

- Consider the corpus as a sequence of sentences $C = s_1 s_2 \dots s_S$, and record the average sentence length L_S.
- The entropy rate of C divided by L_S is δ.
- Chao & Shen (2004) developed an estimator for that entropy

$$H_s^{\text{C-S}}(S) = -\sum_{s \in C} \frac{\tilde{p}(s) \ln \tilde{p}(s)}{1 - [1 - \tilde{p}(s)]^S}$$

$\tilde{p}(s)$: Good-Turing adjusted probability of each sentence

$$\tilde{p}(s) = \left(1 - \frac{f_1}{S}\right) \frac{f(s)}{S}$$

- The convergence is well-modelled by

$$H_s^{\text{C-S}}(S) \approx L_S \delta + a\, S^{-b} \ln S, \quad b > 0$$

which can again be fitted from corpora of different sizes to estimate h.

Corpora & Processing

- Open American National corpus: ~ 12M words written, and ~ 3M words spoken

Corpora & Processing

- Open American National corpus: ˜ 12M words written, and ˜ 3M words spoken
- Synchronic corpus: Consisting of samples written or spoken by native US speakers in the 1990's.

Corpora & Processing

- Open American National corpus: ~ 12M words written, and ~ 3M words spoken
- Synchronic corpus: Consisting of samples written or spoken by native US speakers in the 1990's.
- Ordering of sentences randomized to ensure stationarity

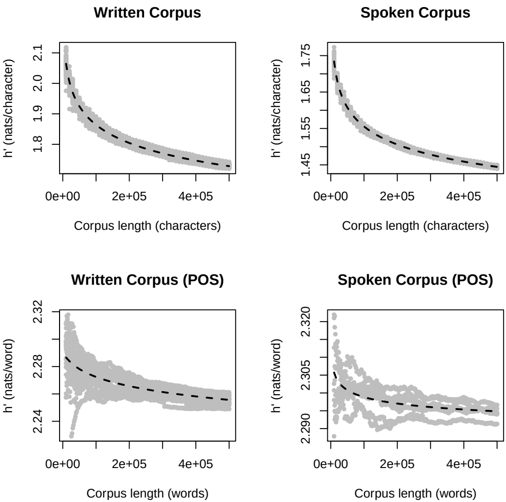
Corpora & Processing

- Open American National corpus: ˜ 12M words written, and ˜ 3M words spoken
- Synchronic corpus: Consisting of samples written or spoken by native US speakers in the 1990's.
- Ordering of sentences randomized to ensure stationarity
- Compute h(N) and $\delta(N)$ for subsets of the corpora of different sizes.

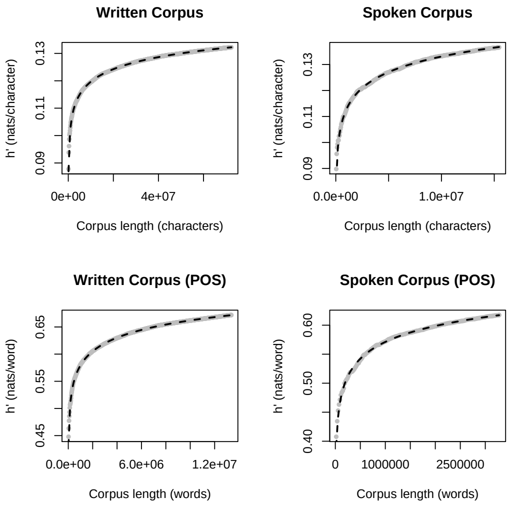
Corpora & Processing

- Open American National corpus: ~ 12M words written, and ~ 3M words spoken
- Synchronic corpus: Consisting of samples written or spoken by native US speakers in the 1990's.
- Ordering of sentences randomized to ensure stationarity
- Compute h(N) and $\delta(N)$ for subsets of the corpora of different sizes.
- Use non-linear regression to estimate h and δ for $N \rightarrow \infty$.

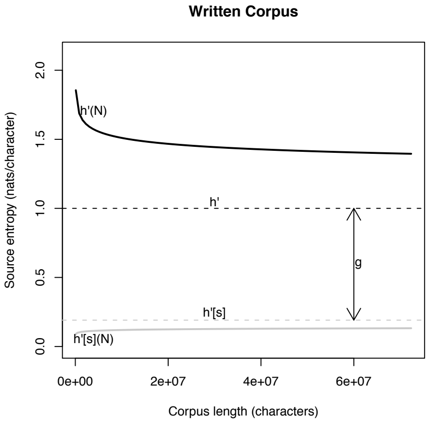
Convergence: *h*

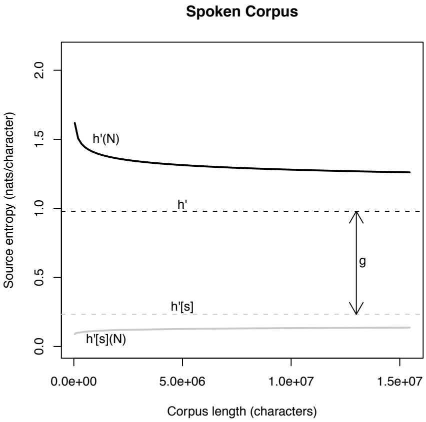


Convergence: δ



Results: Original corpora





Results: Original corpora

- In the convergence limit $h \gg \delta$, that is $g \gg 0$.

Results: Original corpora

- In the convergence limit $h \gg \delta$, that is $g \gg 0$.
- $g > 0$ implies that no finite grammar can account for English.

Results: Original corpora

- In the convergence limit $h \gg \delta$, that is $g \gg 0$.
- $g > 0$ implies that no finite grammar can account for English.
- WARNING! The 'grammar' I am referring conflates syntax and the lexicon

Results: Original corpora

- In the convergence limit $h \gg \delta$, that is $g \gg 0$.
- $g > 0$ implies that no finite grammar can account for English.
- WARNING! The 'grammar' I am referring conflates syntax and the lexicon
- The lexicon is known to be unstable, new words, and new ways to use words are constantly appearing (e.g., Baayen & Renouf, 1996).

Results: Original corpora

- In the convergence limit $h \gg \delta$, that is $g \gg 0$.
- $g > 0$ implies that no finite grammar can account for English.
- WARNING! The 'grammar' I am referring conflates syntax and the lexicon
- The lexicon is known to be unstable, new words, and new ways to use words are constantly appearing (e.g., Baayen & Renouf, 1996).
- Some theories argue that lexicon and grammar should be kept separate (e.g., Chomsky, 1956). The (syntactic) grammar is relatively stable

Discounting the lexicon

- New version of the corpus without lexical information.

Discounting the lexicon

- New version of the corpus without lexical information.
- “That ’s pretty much it .”

Discounting the lexicon

- New version of the corpus without lexical information.
- "That 's pretty much it ."
- "[DT] [VBZ] [RB] [JJ] [PRP] [.]"

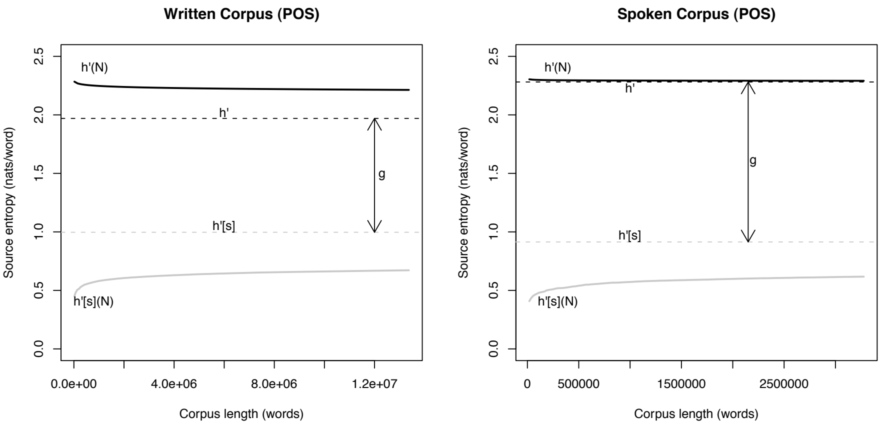
Discounting the lexicon

- New version of the corpus without lexical information.
- “That ’s pretty much it .”
- “[DT] [VBZ] [RB] [JJ] [PRP] [.]”
- These new corpora preserve all “syntactic” information, but discards any lexical information.

Discounting the lexicon

- New version of the corpus without lexical information.
- “That ’s pretty much it .”
- “[DT] [VBZ] [RB] [JJ] [PRP] [.]”
- These new corpora preserve all “syntactic” information, but discards any lexical information.
- I computed h, δ, and g for the new corpora.

Results: POS corpora



Results: POS corpora

- In the convergence limit $h \gg \delta$, that is $g \gg 0$. Between 20% and 40% of the nonzero grammatical density cannot be accounted for by lexical factors.

Results: POS corpora

- In the convergence limit $h \gg \delta$, that is $g \gg 0$. Between 20% and 40% of the nonzero grammatical density cannot be accounted for by lexical factors.
- $g > 0$ implies that no finite grammar can account for English.

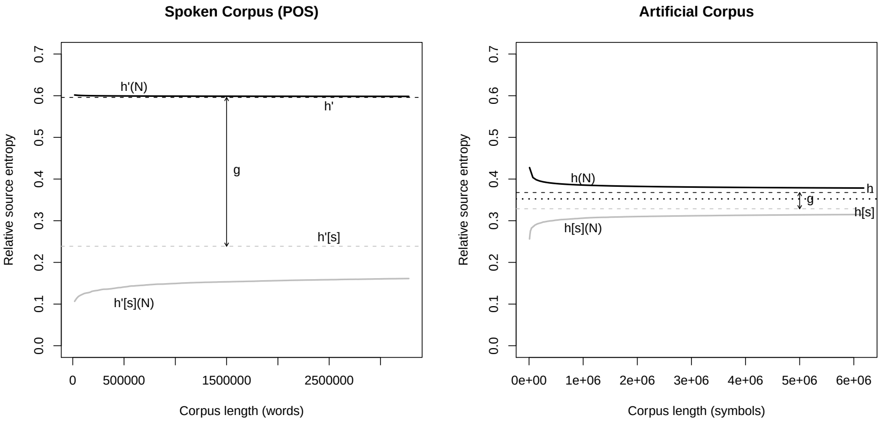
Results: POS corpora

- In the convergence limit $h \gg \delta$, that is $g \gg 0$. Between 20% and 40% of the nonzero grammatical density cannot be accounted for by lexical factors.
- $g > 0$ implies that no finite grammar can account for English.
- WARNING! The estimations are based on extrapolations to infinity. It could be that the the difference between g and δ is due just to inaccuracies.

Artificial corpus baseline

$$
\begin{aligned}
\langle \mathrm{S} \rangle &\rightarrow \langle \mathrm{NP} \rangle \, \langle \mathrm{VP} \rangle \, [.] \\
\langle \mathrm{NP} \rangle &\rightarrow [\textit{pronoun}] \Big| \, \langle \mathrm{NP}_2 \rangle \, [\textit{rel}] \, \langle \mathrm{VP} \rangle \, \Big| \, \langle \mathrm{NP}_2 \rangle \\
\langle \mathrm{NP}_2 \rangle &\rightarrow [\textit{det}][\textit{noun}] \Big| [\textit{det}] \, \langle \mathrm{ADJ} \rangle \, [\textit{noun}] \\
\langle \mathrm{VP} \rangle &\rightarrow [\textit{verb}] \, \langle \mathrm{COMP} \rangle \, \Big| [\textit{verb}] \\
\langle \mathrm{ADJ} \rangle &\rightarrow [\textit{adj}] \Big| [\textit{adj}] \, \langle \mathrm{ADJ} \ \rangle \\
\langle \mathrm{COMP} \rangle &\rightarrow [\textit{adv}] \, \langle \mathrm{COMP} \rangle \, \Big| \, \langle \mathrm{COMP} \rangle \, \langle \mathrm{PP} \rangle \, \Big| \, \langle \mathrm{PP} \rangle \, \Big| [\textit{adv}] \\
\langle \mathrm{PP} \rangle &\rightarrow [\textit{prep}] \, \langle \mathrm{NP} \rangle
\end{aligned}
$$

Results: Artificial baseline



Conclusion (I)

- No finite grammar can account for English.

Conclusion (I)

- No finite grammar can account for English.
- Every grammar will either over-generate or undergenerate. As we increase corpus size, the grammar will remain incomplete ($G \to \infty$)

Conclusion (I)

- No finite grammar can account for English.
- Every grammar will either over-generate or undergenerate. As we increase corpus size, the grammar will remain incomplete ($G \rightarrow \infty$)
- This cannot be due to non-grammatical (i.e., incomplete, etc.) senctences. Those are counted by δ.

Conclusion (I)

- No finite grammar can account for English.
- Every grammar will either over-generate or undergenerate. As we increase corpus size, the grammar will remain incomplete ($G \to \infty$)
- This cannot be due to non-grammatical (i.e., incomplete, etc.) senctences. Those are counted by δ.
- This goes against the generative hypothesis (Chomsky, 1956), that language is an infinite object generated by a finite grammar. No such finite grammar can account for human languages.

Conclusion I

- "Were a language ever completely "grammatical", it would be a perfect engine of conceptual expression. Unfortunately, or luckily, no language is tyrannically consistent. All grammars leak." (Sapir, 1921).

Conclusion I

- "Were a language ever completely "grammatical", it would be a perfect engine of conceptual expression. Unfortunately, or luckily, no language is tyrannically consistent. All grammars leak." (Sapir, 1921).
- Speakers ultimately require a finite representation of the language they acquire.

Conclusion I

- "Were a language ever completely "grammatical", it would be a perfect engine of conceptual expression. Unfortunately, or luckily, no language is tyrannically consistent. All grammars leak." (Sapir, 1921).
- Speakers ultimately require a finite representation of the language they acquire.
- As this grammar is bound to be imperfect, speakers must make use of statistics to minimize de 'leakage'.

Conclusion I

- "Were a language ever completely "grammatical", it would be a perfect engine of conceptual expression. Unfortunately, or luckily, no language is tyrannically consistent. All grammars leak." (Sapir, 1921).
- Speakers ultimately require a finite representation of the language they acquire.
- As this grammar is bound to be imperfect, speakers must make use of statistics to minimize de 'leakage'.
- This will result in cumulative language change.

Inflectional Complexity

- So, grammatical density (g) provides an objective measure of language complexity

Inflectional Complexity

- So, grammatical density (g) provides an objective measure of language complexity
- In order to ignore orthographical factors, it is best to use a per sentence measure of g:

  $$g^{(s)} = S \cdot g,$$

  where S is the mean sentence length in characters.

Inflectional Complexity

- So, grammatical density (g) provides an objective measure of language complexity
- In order to ignore orthographical factors, it is best to use a per sentence measure of g:

$$g^{(s)} = S \cdot g,$$

where S is the mean sentence length in characters.
- Can we use this to measure the complexity of a morphological system?

Inflectional Complexity

- So, grammatical density (g) provides an objective measure of language complexity
- In order to ignore orthographical factors, it is best to use a per sentence measure of g:

$$g^{(s)} = S \cdot g,$$

where S is the mean sentence length in characters.
- Can we use this to measure the complexity of a morphological system?
- A way to do this is to compare the g for samples of language with and without morphological information

Inflectional Complexity

- Inflectional information can be removed by lemmatizing a corpus, that is removing all inflectional markers

Inflectional Complexity

- Inflectional information can be removed by lemmatizing a corpus, that is removing all inflectional markers
- The difference between the original and lemmatized versions indicates the additional (per sentence) information that is carried by the inflectional system

$$g^{(s)}_{\text{inflectional}} = g^{(s)}_{\text{original}} - g^{(s)}_{\text{lemmatized}}$$

Paradigmatic *vs.* Syntagmatic?

- Many approaches to morphological complexity (embarrassingly including my own Moscoso del Prado 2004) treat inflectional paradigms as plain sets of forms.

Paradigmatic *vs.* Syntagmatic?

- Many approaches to morphological complexity (embarrassingly including my own Moscoso del Prado 2004) treat inflectional paradigms as plain sets of forms.
- The infectional paradigm
  to fail : fail, fails, failing, failed

Paradigmatic *vs.* Syntagmatic?

- Many approaches to morphological complexity (embarrassingly including my own Moscoso del Prado 2004) treat inflectional paradigms as plain sets of forms.
- The infectional paradigm
  to fail : fail, fails, failing, failed
- is not at all different from:
  to fail : fails, failed, failing, fail

Paradigmatic *vs.* Syntagmatic?

- Many approaches to morphological complexity (embarrassingly including my own Moscoso del Prado 2004) treat inflectional paradigms as plain sets of forms.
- The infectional paradigm
  to fail : fail, fails, failing, failed
- is not at all different from:
  to fail : fails, failed, failing, fail
- This results in 'form counts' measures of complexity (perhaps refined to consider the relative frequencies of the forms)

Paradigmatic *vs.* Syntagmatic?

- As opposed to the counting monkeys that some of us have been, those of you who ever took a Linguistics 101 class (in my excuse, I never did), find it self-evident that plain form counting is a capital sin.

Paradigmatic *vs.* Syntagmatic?

- As opposed to the counting monkeys that some of us have been, those of you who ever took a Linguistics 101 class (in my excuse, I never did), find it self-evident that plain form counting is a capital sin.
- One needs to explicitly consider the functions served by each of the inflected forms: Paradigm Cell Filling Problem, inferences across paradigm cells (see, *e.g.*, Ackerman & Malouf)

Paradigmatic *vs.* Syntagmatic?

- As opposed to the counting monkeys that some of us have been, those of you who ever took a Linguistics 101 class (in my excuse, I never did), find it self-evident that plain form counting is a capital sin.
- One needs to explicitly consider the functions served by each of the inflected forms: Paradigm Cell Filling Problem, inferences across paradigm cells (see, *e.g.*, Ackerman & Malouf)
- In fact, as noticed by some (*e.g.*, Kostić et al., 2003) these functions play an important role in the recognition of inflected forms.

Paradigmatic *vs.* Syntagmatic?

- But, what is a paradigm cell?

Paradigmatic *vs.* Syntagmatic?

- But, what is a paradigm cell?
- A particular grammatical function

Paradigmatic *vs.* Syntagmatic?

- But, what is a paradigm cell?
- A particular grammatical function
- How does one learn which forms go in which cells?

Paradigmatic *vs.* Syntagmatic?

- But, what is a paradigm cell?
- A particular grammatical function
- How does one learn which forms go in which cells?
- By their use in speech and text (syntagmatics!).

Paradigmatic *vs.* Syntagmatic?

- But, what is a paradigm cell?
- A particular grammatical function
- How does one learn which forms go in which cells?
- By their use in speech and text (syntagmatics!).
- Is there a difference between paradigmatic and syntagmatic approaches?

Paradigmatic *vs.* Syntagmatic?

- But, what is a paradigm cell?
- A particular grammatical function
- How does one learn which forms go in which cells?
- By their use in speech and text (syntagmatics!).
- Is there a difference between paradigmatic and syntagmatic approaches? I don't think so; the structure (i.e., the 'shape') of paradigms is syntagmatic from the start

Paradigmatic *vs.* Syntagmatic?

- But, what is a paradigm cell?
- A particular grammatical function
- How does one learn which forms go in which cells?
- By their use in speech and text (syntagmatics!).
- Is there a difference between paradigmatic and syntagmatic approaches? I don't think so; the structure (i.e., the 'shape') of paradigms is syntagmatic from the start
- To illustrate this, I will show how morphological complexity measures change dramatically depending on whether one considers the syntagmatic factors

Corpora & Processing

- Europarl Corpus: ~ 10M words (transcribed) in 6 European languages

Corpora & Processing

- Europarl Corpus: ˜ 10M words (transcribed) in 6 European languages
- 11 yearly files (proceedings of the Parlament in a given year)

Corpora & Processing

- Europarl Corpus: ~ 10M words (transcribed) in 6 European languages
- 11 yearly files (proceedings of the Parlament in a given year)
- Ordering of sentences randomized to ensure stationarity

Corpora & Processing

- Europarl Corpus: ˜ 10M words (transcribed) in 6 European languages
- 11 yearly files (proceedings of the Parlament in a given year)
- Ordering of sentences randomized to ensure stationarity
- Compute h(N) and $\delta(N)$ for subsets of the corpora of different sizes.

Corpora & Processing

- Europarl Corpus: ˜ 10M words (transcribed) in 6 European languages
- 11 yearly files (proceedings of the Parlament in a given year)
- Ordering of sentences randomized to ensure stationarity
- Compute h(N) and $\delta(N)$ for subsets of the corpora of different sizes.
- Use non-linear regression to estimate h and δ for $N \to \infty$.

Corpora & Processing

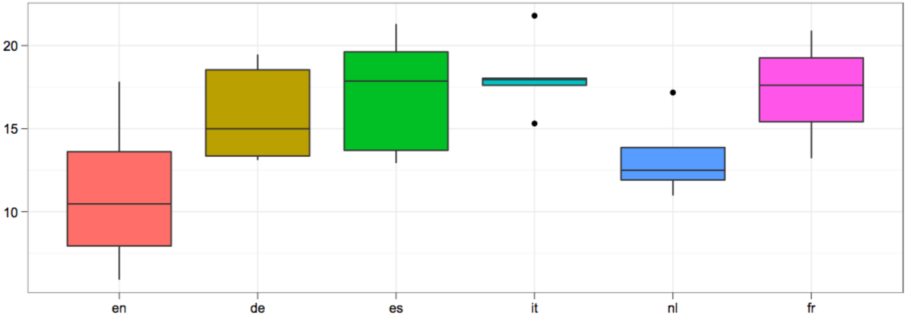
- Four versions of the corpus
  1. Original word order
     - Original
     - Lemmatized

Corpora & Processing

- Four versions of the corpus
  1. Original word order
     - Original
     - Lemmatized
  2. Word order randomized
     - Original
     - Lemmatized
- Compute morphological complexity using:

$$g^{(s)}_{\mathrm{inflectional}} = g^{(s)}_{\mathrm{original}} - g^{(s)}_{\mathrm{lemmatized}}$$

Morphological Complexity (randomized text)



Results (disconsidering function)

- One obtains the gradation of morphological complexity that "form counters" like (considering also the relative frequencies of forms and their regularity):

Results (disconsidering function)

- One obtains the gradation of morphological complexity that "form counters" like (considering also the relative frequencies of forms and their regularity):

$$\text{English} < \text{Dutch} < \text{German} < \text{Romance}$$

Results (disconsidering function)

- One obtains the gradation of morphological complexity that "form counters" like (considering also the relative frequencies of forms and their regularity):

$$\text{English} < \text{Dutch} < \text{German} < \text{Romance}$$

- Morphology appears to be 'costly', it takes more space to describe the regularities of language with morphology than that of a language without it.

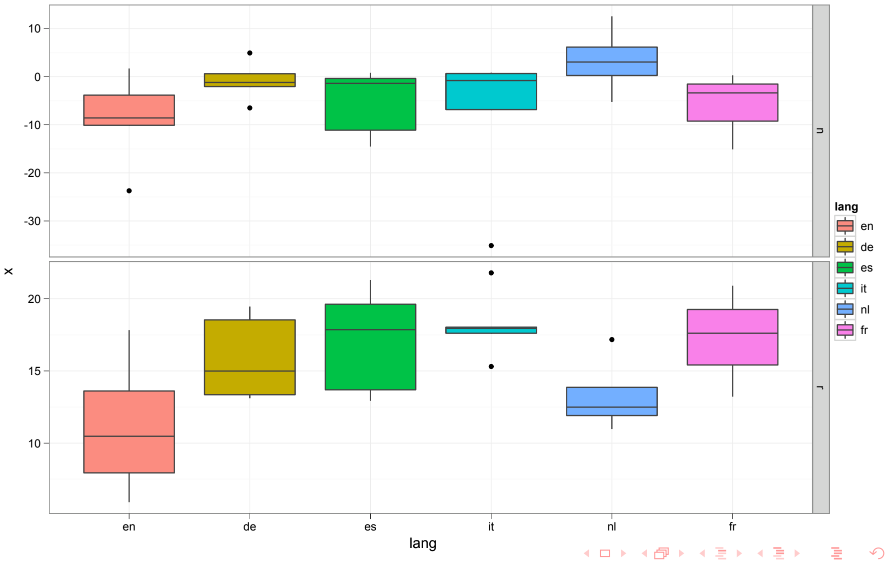
Results (disconsidering function)

- One obtains the gradation of morphological complexity that "form counters" like (considering also the relative frequencies of forms and their regularity):

$$\text{English} < \text{Dutch} < \text{German} < \text{Romance}$$

- Morphology appears to be 'costly', it takes more space to describe the regularities of language with morphology than that of a language without it.
- But this ignores the structure and function of the paradigms

Morphological Complexity



Results (considering function)

- The gradation dissappears – differences in morphological complexity across languages are not significant

Results (considering function)

- The gradation dissappears – differences in morphological complexity across languages are not significant
- Morphology is not very 'costly', it takes sightly less space to describe the regularities of language with morphology than that of a language without it.

Results (considering function)

- The gradation dissappears – differences in morphological complexity across languages are not significant
- Morphology is not very ‘costly’, it takes sightly less space to describe the regularities of language with morphology than that of a language without it.
- Morphology is there for a good reason! It is not capricious

Conclusion (II)

- No ‘morphology by itself’

Conclusion (II)

- No ‘morphology by itself’
- We need to regard the actual structure of the paradigms (cell filling problem, implications, etc.)

Conclusion (II)

- No 'morphology by itself'
- We need to regard the actual structure of the paradigms (cell filling problem, implications, etc.)
- But we also need to explicitly consider the actual functions that the cells serve

Conclusion (II)

- No ‘morphology by itself’
- We need to regard the actual structure of the paradigms (cell filling problem, implications, etc.)
- But we also need to explicitly consider the actual functions that the cells serve
- If one amputates syntax from morphology, one ends up with a disabled morphology and a mirage of complexity.

Conclusion (II)

- No 'morphology by itself'
- We need to regard the actual structure of the paradigms (cell filling problem, implications, etc.)
- But we also need to explicitly consider the actual functions that the cells serve
- If one amputates syntax from morphology, one ends up with a disabled morphology and a mirage of complexity.
- These measures can be computed in a rather theory-free way from corpora: One can reason about the morphology without actually describing the morphological system.