



- [about *Plus*](#)
- [support *Plus*](#)
- [subscribe to *Plus*](#)
- [terms of use](#)

search *plus* with [google](#)

- [home](#)
- [latest issue](#)
- [explore the archive](#)
- [careers library](#)
- [news](#)

© 1997–2004, *Millennium Mathematics Project, University of Cambridge.*

Permission is granted to print and copy this page on paper for non-commercial use. For other uses, including electronic redistribution, please contact us.

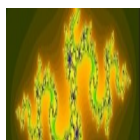
Latest news



John D Barrow wins Templeton Prize

(22/03/2006)

More honours for the director of the Millennium Mathematics Project



A fat chance of chaos?

(14/03/2006)

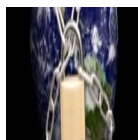
Some news on Julia sets



Science in school

(21/02/2006)

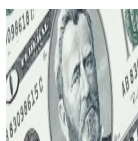
A new Europe-wide science magazine



The dangers of cracking hash

(16/02/2006)

A crisis in cryptography?



Beating bird flu with bills

(02/02/2006)

The travels of bank notes give important clues to epidemiologists



Innate geometry

(27/01/2006)

Is geometry hard-wired into our brain?

Plus... more news from the world of maths

Explore the news archive

Subscribe to our free newsletter

Get the *Plus* headlines [XML](#) (More info)

Highlight the maths in this page (More info)

Get your own copy of all the *Plus* posters!



12/08/2005

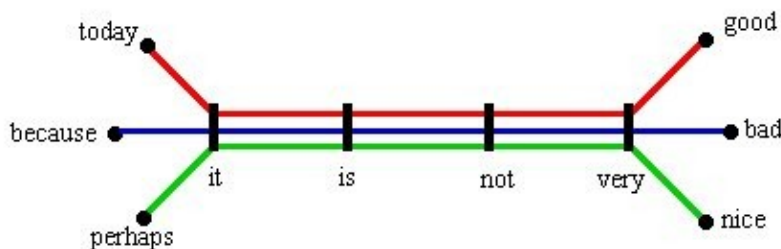
News

Machine prose

Imagine you get on the train one morning and rather than hearing the familiar "Stand clear of the doors" or "This train is for Cambridge" you're confronted with a piece of poetry, freshly composed by the on-board computer. Well, admittedly, such a thing is still some years away, and may never even happen (poetic trains might prove a bit of a nuisance), but scientists are well on the way to constructing a computer program that can teach itself languages and make up its own sentences.

Given a piece of text in any language, the program called ADIOS – automatic distillation of structure – searches for patterns and structures which it then generalises to produce new and meaningful sentences. The ADIOS algorithm is based on statistical and algebraic methods performed on one of the most basic and versatile objects of mathematics – the graph.

Given a text, the program loads it as a graph by representing each word by a *node*, or *vertex*, and each sentence by a sequence of nodes connected by lines. A string of words in the text is now represented by a *path* in the graph.



The words '...it is not very...' appear in three sentences.

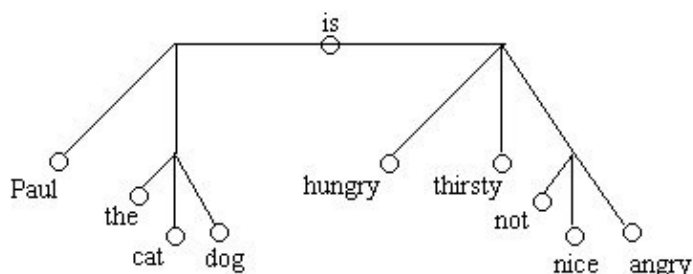
Once the text is in the form of a graph, the algorithm can get its teeth into it. It starts out by performing a statistical analysis to see which paths, or strings of words, occur unusually often. It then decides that those that appear most frequently – called "significant patterns" – can safely be regarded as a single unit, and replaces the set of vertices in each of these patterns by a single vertex, thus creating a new, generalised, graph.

Next, the program looks for paths in the graph which just differ by one vertex. These stand for parts of sentences that just differ by one word (or compound of words) like "the cat is hungry" and "the dog is hungry". Performing another statistical test on the frequency of these paths, the program identifies classes of vertices, or words, that can be regarded as interchangeable, or equivalent. The sentence involved is legitimate no matter which of the words in the class – in our example "cat" or "dog" – you put in.

Having noted down these classes, the program looks again for paths that occur very often, and, just as before, creates a third, even more generalised graph.

And so it continues, on and on, finding equivalence classes and significant patterns, and creating generalised graphs, until, finally, the resulting graph throws up no more significant patterns. Once this has happened, the program puts all the information it has gained into new graphs which display the equivalent words, or sentence fragments, and show how they are connected. Here is an example:

Machine prose



Although the original text may only have contained the sentences

"Paul is hungry" "Paul is not nice" "The cat is thirsty" and "The dog is not angry"

the program is now able to piece together new sentences, like "Paul is thirsty" and "The cat is not angry" by interchanging equivalent words or sentence fragments.

The scientists who are behind this latest advance, [Zach Solan](#), [David Horn](#), [Eytan Ruppin](#) and [Shimon Edelman](#), have tested their algorithm using several standard language proficiency tests, in languages as diverse as English and Chinese, and found that it scored well. They say that "this is the first time an unsupervised algorithm is shown capable of learning complex syntax [and] generating grammatical novel sentences". The algorithm is not restricted to languages either: it can find patterns and structure within any set of strings, whether it's amino acid sequence data (in bioinformatics) or musical notation.

All this doesn't mean, of course, that the program actually "understands" what it's saying. It simply knows how to have a good go at piecing together fragments of sentences it has identified, in the hope that they are grammatically correct. So if, like me, you're prone to swearing at your computer, you can safely continue to do so: it won't answer back for a long while yet.

Further Reading

You can find out more about ADIOS on the [ADIOS project homepage](#).

Marianne Freiberger

[Download PDF version](#) | [Printer friendly version](#)

[Return to article](#)

- [contact](#)
- [copyright info](#)
- [sponsors](#)
- [privacy info](#)



Machine prose

Plus is part of the family of activities in the Millennium Mathematics Project, which also includes the NRICH and MOTIVATE sites.