Word Parsing

2014-05-18 BY GENE

It was in my Grandfather's breakfast area, in my teens, that I realized that I even though I knew about overlapping word-parts, I didn't know how to handle "word part under-lapping" or "unknowns." I became determined to study computer programming.

I've been working on the problem of how to best break a word into parts for a while. Naturally, there have been a number of milestones. I remember one Chicago diner at 2:30AM, when I figured out the current mystery. It was probably something combinatorial and mechanical…

I remember when my friend, Luc pointed out that the "brute forcing" that I was doing over every possible word part combination, was just iteration. Brilliant! Hallelujah! Things would no longer bog down after 20 letters and take longer than the Universe to evolve.

I live for brief moments of happiness like that… But that's another post, or therapy session.

What I have found along the way is that parsing a word, of known parts, like a science term, is a neat mechanical process of just a couple steps…

So:

Given a word like, biology, my mind organically breaks it into parts, each with micro-meanings. “Bio-logy” or is it “bi-o-log-y?” And what about that “y” on the end? Does it mean “like?”

These words are squirrely things!

After a while, I realized that the lexicon of parts needed to have **all** of bi, bio, o, log and logy.

This file: https://github.com/ology/Lex/blob/master/abioticaly.txt lays out what I wanted to see, for a made up test word. (And it’s amazing how many quadrille student notebooks of mine have it scribbled inside!)

Then, I went off to University.

…Time passes…

I think about how to keep track of over-and-under-laps (i.e. multiple knowns and unknown parts existing in the same position).

I think of how having a finite lexicon of parts makes it “domain specific” and also measurable. What is the “score” of a particular combination versus another? Equally valid combinations should have the same score.

I gave a short presentation about these ideas at a software conference (“YAPC 19100“). The luminaries of the computer language itself (i.e. Perl) – Larry, Damian, Nathaniel, Randal were in the front row! I barely made it!

But that was then. More time passes… Jobs come and go. Glaciers form and erode.

I wrote https://metacpan.org/release/Lingua-TokenParse as a first attempt. But it is not sufficient or efficient.

Along the way I bought every single science-word-formation dictionary I could find. On reading them I realized that an “agnostic” lexicon of regular expressions could encode whether the word part was a suffix or prefix. Enlightenment!

Cut to today (well a couple days ago), I finally unlocked the puzzle by realizing how to increment comparison sets. These sets are nothing less than the power-set of all the combinations of the known bitstrings! Enlightenment again!

If you’re curious, check out https://github.com/ology/Lingua-Word-Parser for the latest developments.

Read the next installment at Word Parsing, Part 2.

FILED UNDER: SOFTWARE
TAGGED WITH: MORPHEME, PERL, TERMINOLOGY, WORD PART

Epistemologist-at-large

^ Top