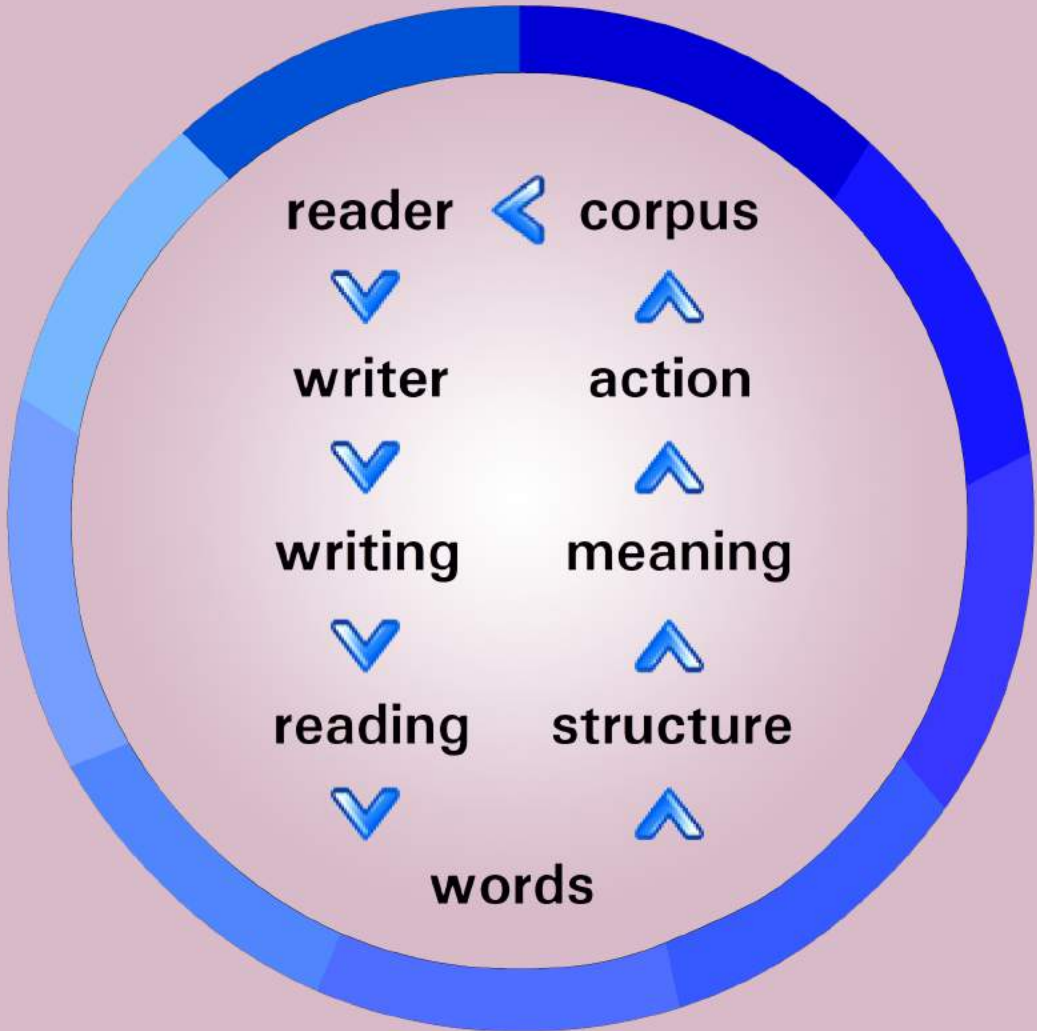




intermedia writing systems
revised 3rd edition



matthew t. butler
intermedia writing systems

matthew t butler @ gmail . com

Exercise 1: The Platypus and the Chimaera

Once upon a time there was a platypus who, boasting how he was more unique than anyone else, was forever teasing chimaera for it being supernatural. Then one day, the irate chimaera answered back: "Who do you think you are? There's no denying you're real, but even you are uncanny!" The platypus squealed with laughter.

"I'm thought of as uncanny? By whom? Not you, surely! I bet there's nobody in the world that can prove I'm not real. Now, why don't you try?"

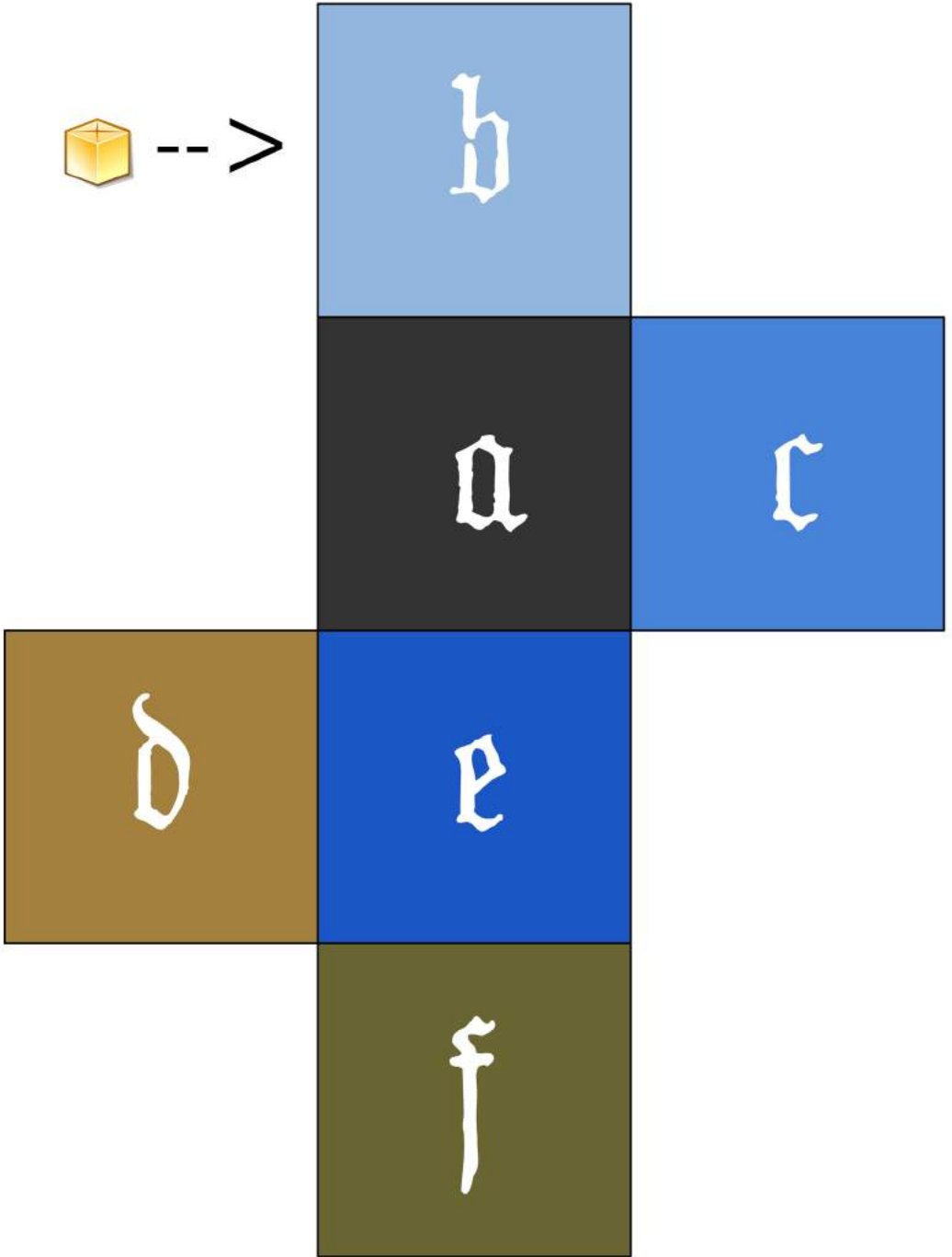
Annoyed by such bragging, the chimaera accepted the challenge. It was planned, and the next day at dawn they stood. The platypus yawned sleepily as the meek chimaera turned slowly on. When the platypus saw how painfully slow his rival was, he decided, half asleep on his feet, to have a quick nap. "Take your time!" he said. "I'll have forty winks and catch up with you in a minute."

The platypus woke with a start from a fitful sleep and gazed round, looking for the chimaera. But the creature was only a short distance away, having barely covered a third of the course. Breathing a sigh of relief, the platypus decided he might as well have breakfast too, and off he went to munch some yabbies he had noticed. But the heavy meal and the hot sun made his eyelids droop. With a careless glance at the chimaera, now halfway along the course, he decided to have another snooze before flashing past the winning post. And smiling at the thought of the look on the chimaera's face when it saw the platypus speed by, he fell fast asleep and was soon snoring happily.

The sun started to sink below the horizon, and the chimaera, who had been becoming more real since morning, was scarcely a yard from the finish. At that very point, the platypus woke with a jolt. He could see the chimaera as a speck on the horizon and away he dashed. He leapt and bounded at a great rate, his tongue lolling, and gasping for breath. Just a little more and he'd be first at the finish. But the platypus's last leap was just too late, for the chimaera had beaten him to the horizon. Poor platypus! Tired and in disgrace, he slumped down beside the chimaera who was silently smiling at him.

"Ick bün all dor!" he said.





crypto-adaptation box (unfolded)

A:

secret self-modification by an agent in response to that agent's changing systemic conditions

Receiving predictive information concerning one's environment has been the greedy desire of humankind since the construction of language as a tool thousands of years ago. If only we could describe what is just over the horizon! What an edge we'd have over our "competition" if we already had the specialized language to discuss the newness of tomorrow today.

b:

temporary micro-alterations in belief structure to accomplish a site-specific task

As we continuously enter the new newness of so-called 'new media' and 'new age', our natural language is intermingling with programming language in shocking and considerable ways. Meaning is constructed out of clearly defined protocols that either "work" or "do not work" — there is no third position here. For the first time in existence, poem and code are the same. Though this 'new language' is less susceptible to the confusing ambiguities of old (i.e. we cannot use variables and handlers without clearly declaring their value and defining their procedure) there is still a magic in language which contributes to human evolution.

C:

crosstalk between versions

Language in some way creates the very reality in which we live. Words and concepts point to realities beyond the sensory world and assist us in making contact with a dimension that is higher, lower and parallel.

D:

ritual narrative constructed to help explain random phenomena

The intent was to jitter thought process with randomness in language. That jitter is now quake-like in magnitude thanks to advances in programmable electronics and automatic algorithms. What took them hours with newspapers and scissors now takes us seconds with computers — I can do 4 million words over brunch. We are no longer creating experimental voices but communicating with beings from other worlds.

E:

orchestrated prescience

Intangible Ideas, in Plato's conception — as supersensible realities beyond human thought — are captured in scripts, as prisoners in their cells, and released by the act of perusal, setting the prisoners free. These Ideas reside in the words and terms independent of the programs and books in which the words are encased. Yet how and where, in the interval between their setting down and their taking up, do they abide? By what secret tract is their existence in the mind of the author/programmer connected with their resuscitation in the mind of the reader/user? Why at the sight of certain lines and figures on the voiceless page/screen do these particular thoughts spring up into renewed activity? What is the undiscoverable nexus between the physical vibrations of light and these immaterial substances of our noetic life?

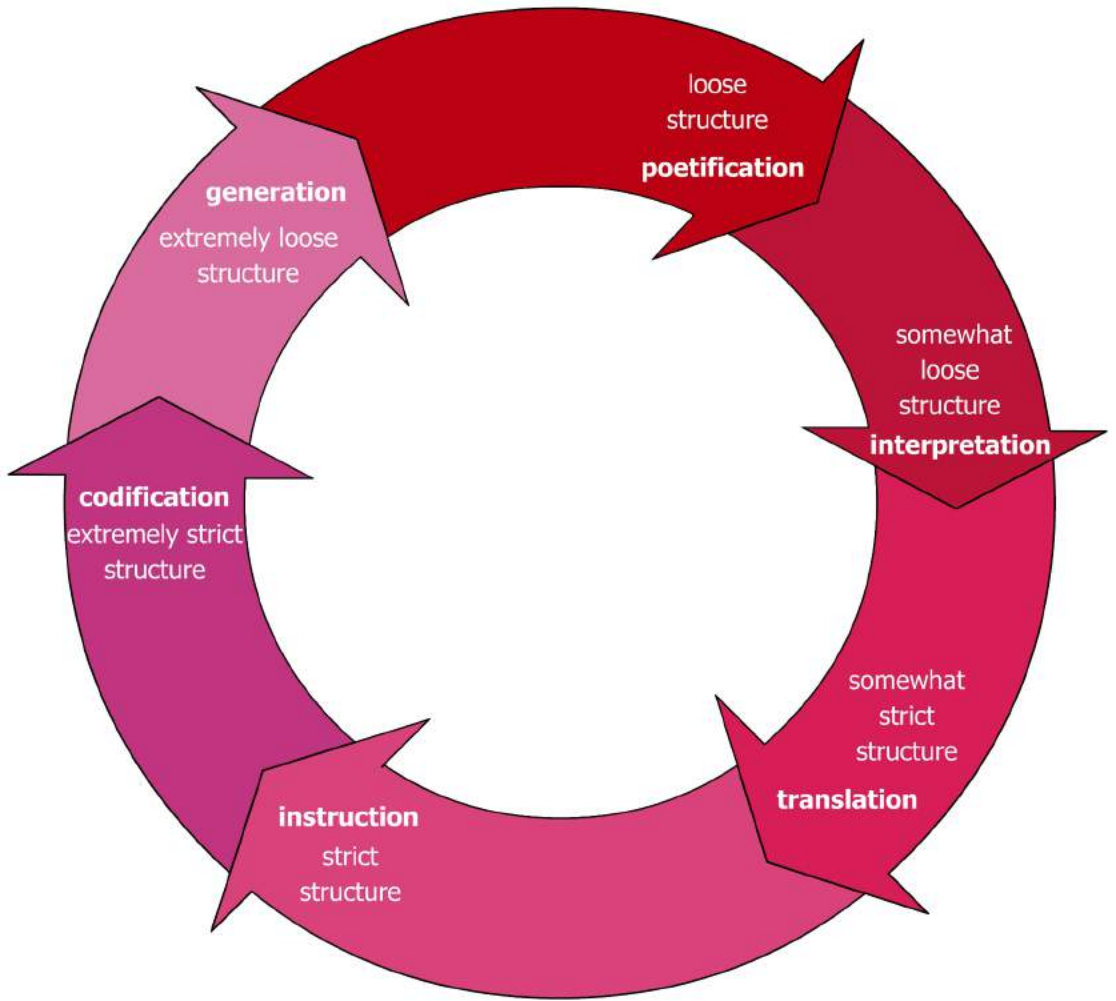
f:

a hermetic sales pitch from beyond

Homo Sapiens unlocked the mystery power of their brains through the invention of language thousands of years ago. Language allowed them to reflect in a self-aware manner, invent finely crafted tools, and create handmade ornaments, shelter, and paintings. This epoch process is culminating in programmable electronics. We are becoming aware, again since before language, that the fracture between “subject” and “object” is unreal. What is more necessary? the operating language of a word processor or the inputted creative language of its user?



Akoll Tapoze Yitsoung!



The life cycle of a piece of intermedia writing differs significantly from other writing methods. Whereas classic writing techniques tend to focus on the *origination* of a body (of text), intermedia writing is primarily concerned with the *transformation* of the body (of text). Notice that in the above diagram, the life cycle of a piece of writing never begins or ends. It is a continuously evolving structure forever fluxuating in strictness to its own form.

Intermedia writing systems work with this natural cycle of life to instantiate bodies of text and texts of bodies.

These concepts are illustrated in the following example. We will arbitrarily begin with the *generation* phase of the life cycle and work around the cycle once. Keep in mind that the writing process does not end with one cycle. It continues indefinitely.

Examining the generation phase, we see that we're now working with a crossword themed scratch-off lottery ticket. It contains 19 words in an extremely loose structure. Let's follow the writing around one cycle and see what's happening in each phase.

generation:

In this phase we're working with 19 words with no syntax. This is not to say there is no structure at all, only that it is considered extremely loose.

*jeweler once cookie semi length noisy free factor lawn off uncommon ouch
elk illusion wax inn yesterday relic being*

poetification:

Moving into poetification represents a reordering of the words into a slightly stricter structure to create the beginning of imagery. Notice the introduction of nominal prepositions and conjunctions.

*a noisy semi of uncommon length (once free of illusion) was an ouch factor
off the Elk Inn lawn yesterday*

wax and a cookie was the relic of the jeweler

interpretation:

The interpretation phase begins to add meaning to the writing by tightening the looseness of structure. Note the inclusion of descriptive words and use of grammatical rules.

*Big rig truckers who attend the Elk's Lodge should have high-fiber cookies
wrapped in wax paper to eat while driving. The constipation experienced
from the cookies may be painful, but it will remind them of where they
started (retrieve snack) and where they are going (relieve self).*

translation:

The interpreted body of text is expressed in a different way during the translation phase. It is considered a translation because it deals not only with the body of text but also with the process. The structure is now somewhat strict.

*The Crossword will win if truckers at the Elk's Lodge receive a cookie
recipe from a jeweler.*

instruction:

Once the translation phase is complete, the body of text begins to instruct. It tells us how to behave - either mentally or physically. The structure is now strict since it is quite precise in describing what to do. Most classic forms of writing such as novels and essays rarely escape the instruction phase.

Write a high-fiber cookie recipe on wax paper and mail it to all the truckers at the local Elk's Lodge with the return address of a local jeweler.

codification:

The body of text now enters codification. The structure becomes extremely strict and is often expressed algorithmically or even in machine-readable form. This transformation allows the writing to generate itself and move into the next phase.

To: Elks Lodge 590 - 637 Foster Rd - Iowa City, IA 52245

From: MC Ginsberg Jewelers - 110 E Washington St - Iowa City, IA 52240

Trucker's Cookies

1 c. whole wheat flour

1 c. bran flakes

1/2 c. oats

1/2 c. wheat germ

1/2 tsp. baking soda

1/2 tsp. salt

2 packets Sweet 'N Low or 1/2 c. sugar

1/2 c. corn oil

1 c. molasses (unsulphured)

2 eggs

1/2 c. fresh ground peanut butter

1/2 c. raisins

1/2 c. chopped nuts

Mix together dry ingredients; set aside. In mixing bowl on low speed mix wet ingredients.. Add peanut butter and mix well. Add the dry ingredients. Drop by spoon onto cookie sheet sprayed with Pam. Bake at 325 degrees for 20 minutes. Makes 4 dozen.

Excercise 2: Open Wound

You are given an instance from the codification phase of the life cycle.

It is expressed as a perl script called cut-up.pl. After reading this script

1. describe how the next phase of generation will look.
2. Advancing through the life cycle, how will the future phases look?

To help you if the code is difficult to read, the previous instruction phase has been provided below.

instruction:

To graft the flesh of one body of text to another: Tag the elements of two word-sets with the correct part-of-speech, preserving the order of the elements. Sort the elements of the first word-set into lists based on their part-of-speech. Remove the elements from the second tagged word-set, preserving the order of the tags. For each tag in the second word-set, substitute it with a random member of the corresponding tag list from the first word-set. Repeat if necessary using the resulting third word-set or something new.

In the intermedium of the graft, new flesh will grow and adhere.

I. When a TAG button is pressed

1. Accept any amount of text from a field called *INPUT*. Perform a part-of-speech tagging and display the tagged words in a text field called *TAGGED OUTPUT*. The value of the *TAGGED OUTPUT* field might look */NN something/NN like/IN this/DET /PP*
2. Copy each word (without its POS tag) into an appropriate field based on it's part-of-speech. The user may delete words or add their own words. Make all words lowercase.

II. When a GENERATE button is pressed

1. Accept a value the user has inputed in the *COMMON* field. The value can be from 0-100. The number corresponds to an array that contains the 1000 most common words in the English language. (For example, a value of 20 in the *COMMON* text field means the 20 most common words will not be substituted in the next step.)
2. Substitute each of the words or punctuation marks in the *TAGGED OUTPUT* field with a different word or punctuation mark from the corresponding POS field unless that word is in the range specified by the *COMMON* value in which case it is not substituted. Display the result in a *SUBSTITUTION* text field.
3. Capitalize any letter directly following the PP, Punctuation, Sentence Ender tag then remove the POS tags from the *SUBSTITUTION* text. Display the new text in a field called *RECOMBINED*.

```

#!/usr/bin/perl
use strict;
use warnings;
no warnings 'uninitialized';

use CGI::Carp 'fatalsToBrowser';
use CGI qw/:standard *table/;
use Lingua::EN::Tagger;

1 while not <<'THISWILLNOTBECOMPILED';
["Perhaps events are pre-written and pre-recorded and when you cut
 word lines the future leaks out."]
THISWILLNOTBECOMPILED

my $app_title = 'Open Wound';

# Possible Lingua::EN::Tagger tags.
my @tags = qw/cc cd det ex fw in jj jjr jjs ls md nn nnp nnps nns pdt pos prp
          prps rb rbr rbs rp sym to uh vb vbd vbg vbn vbp vbz wdt wp wps
          wrb pp ppc ppd ppl ppr pps lrb rrb/;

# 1000 most common English words to skip (too long for print)
# find a list of the 1000 most common words then replace *** below with the list
# uncomment line below once wordlist is in place
# my @common_words = qw/****/;

my %tag_input; # Values for filling in the tag fields.

if (param('action') = ~ /^s*tag\s*$/i) {

    $app_title .= ' | [output]';

    # Remove possible leftovers from previous generation cycle.
    Delete(qw/substitution recombined/);

    my $tagger = new Lingua::EN::Tagger;
    my $tagged_xml = $tagger->add_tags(param('source'));

    my %noun_phrases = $tagger->get_noun_phrases($tagged_xml);
    param(noun_phrases => join ', ', keys %noun_phrases);

    # Extract and store occurrences for each tag.
    while ($tagged_xml = ~ m!<([>]+)>([<]+)</>!g) {
        $tag_input{$1}{lc $2} + +;
    }
}

```

```

# Convert XML-type output to word/TAG format.
$tagged_xml = ~ s!<([^\>]+)>([^\<]+)</\1>!$2/\U$1!g;
param(tagged_text => $tagged_xml);
}
elseif (param('action') = ~ /^s*generate\s*$$/i) {

    $app_title .= ' | [generated]';

    # Prepare a hash of tag occurrences for convenient randomization.
    my %variants;
    foreach (@tags) {
        $variants{$_} = [ split /\s+/, param($_) ] if param($_);
    }

    # Get a list of common words to skip.
    my $common_words_limit = param('common_limit') || @common_words;
    my %common_words_checker;
    @common_words_checker{@common_words[0 .. ($common_words_limit - 1)]} = ();

    # Subroutine performing random substitution. Called from the regex below.
    my $substitute_match = sub {
        my ($part, $tag) = @_;

        return $part if exists $common_words_checker{lc $part};
        my $part_variants = $variants{lc $tag}
            or return $part;

        my $rand_part = $part_variants->[rand(@$part_variants)];
        return $rand_part;
    };

    # Generate "substitution" and "recombined" fields.

    my $text = param('tagged_text');

    $text = ~ s!(\S+)/([A-Z]+)!$substitute_match->($1,$2) . "/$2"!ge;
    param(substitution => $text);

    $text = ~ s!\s+(?=\S+/(PP[CDLRS]?|RRB))!!g; # remove junk space.
    $text = ~ s!(^s*|/PP\s+)(\S+)!$1\u$2!g; # capitalize beginnings of sentences.
    $text = ~ s!/LRB\s+!!g; # and more junk space and some tags.
    $text = ~ s!/([A-Z]+!!g; # remove all tags.
    param(recombined => $text);
}
else {
    # Initial phase. Perform nothing, just display form.
    $app_title .= ' | [input]';
}

```

```
# Output the form. Field value substitution is done by CGI.pm from param() values
# possibly set in the above code.
```

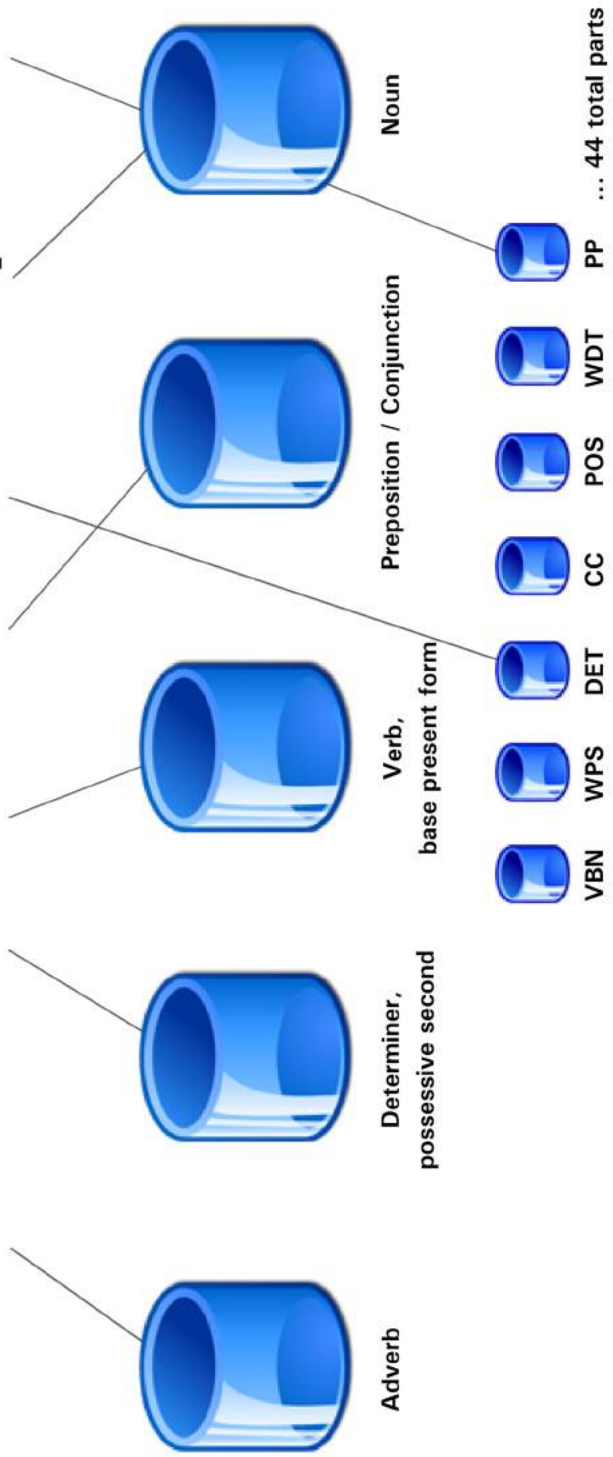
```
print header,
  start_html(-title => $app_title),
  h1($app_title),
  start_form,
  p(strong('Input text:'), br,
    textarea(-name => 'source', -rows => 10, -columns => 100)),
  p(submit(-name => 'action', -value => ' Tag ')),
  p(strong('Tagged text:'), br,
    textarea(-name => 'tagged_text', -rows => 10, -columns => 100)),
  p(strong('Common:'), textfield(-name => 'common_limit', size => 3)),
  p(strong('Noun phrases:'), br, textarea(-name => 'noun_phrases',
    -rows => 4, -columns => 100)),
  start_table();
```

```
# Generate tag fields.
```

```
my $i;
foreach (@tags) {
  param($_ => join(' ', sort keys %{ $tag_input{$_} })) if %tag_input;
  print '<tr>' if $i++ % 3 == 0;
  print td(strong("\U$_:")),
    td(textfield(-name => $_, -size => 30));
}
```

```
print end_table(),
  p(submit(-name => 'action', -value => 'Generate')),
  p(strong('substitution:'), br,
    textarea(-rows => 10, -name => 'substitution', -columns => 100)),
  p(strong('cut-up:'), br,
    textarea(-rows => 10, -name => 'recombined', -columns => 100)),
  end_form;
```

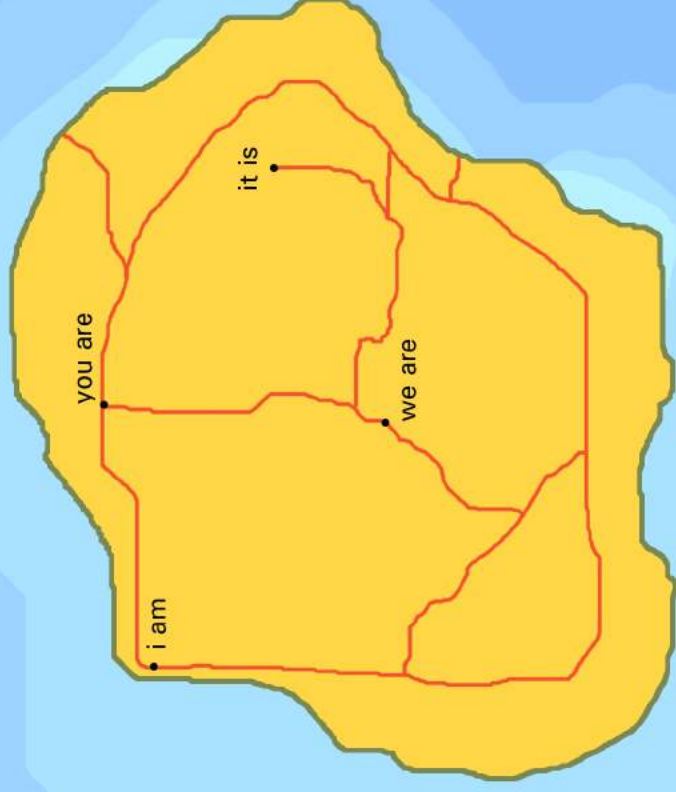
Sometimes I think like a computer.



using the intermedia framework
as a process for writing text

example implementations included

wounds, horizons, and life cycles



ecosystem: platypus, chimaera, quick brown fox, lazy dog