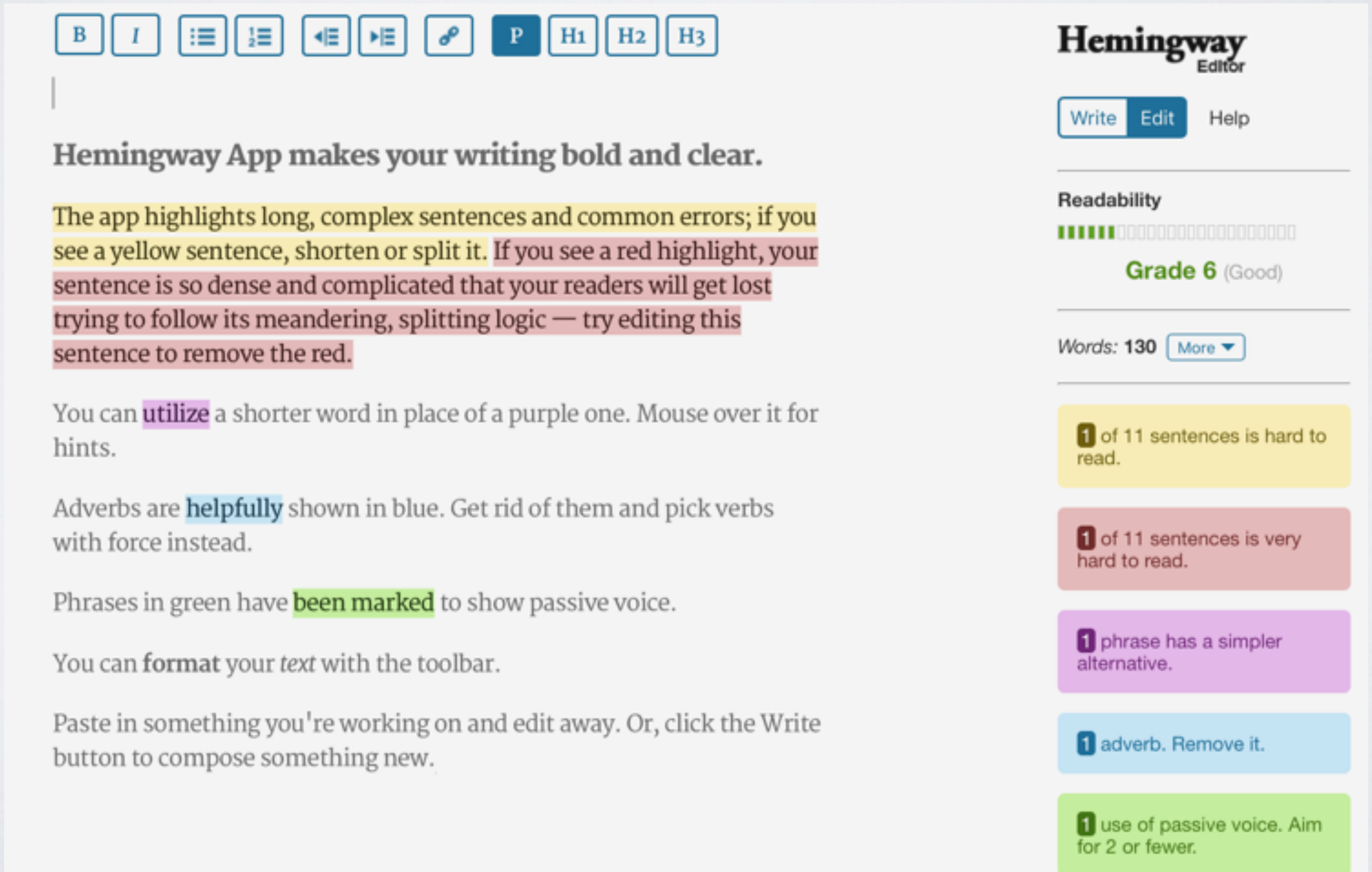
A STUDY ON LINGUISTIC COMPLEXITY

Robin Malhotra

INSPIRATION

- The Hemingway App (http://www.hemingwayapp.com)



Hemingway App makes your writing bold and clear.

The app highlights long, complex sentences and common errors; if you see a yellow sentence, shorten or split it. If you see a red highlight, your sentence is so dense and complicated that your readers will get lost trying to follow its meandering, splitting logic — try editing this sentence to remove the red.

You can utilize a shorter word in place of a purple one. Mouse over it for hints.

Adverbs are helpfully shown in blue. Get rid of them and pick verbs with force instead.

Phrases in green have been marked to show passive voice.

You can **format** your *text* with the toolbar.

Paste in something you're working on and edit away. Or, click the Write button to compose something new.

Readability

Grade 6 (Good)

Words: 130 More

1 of 11 sentences is hard to read.

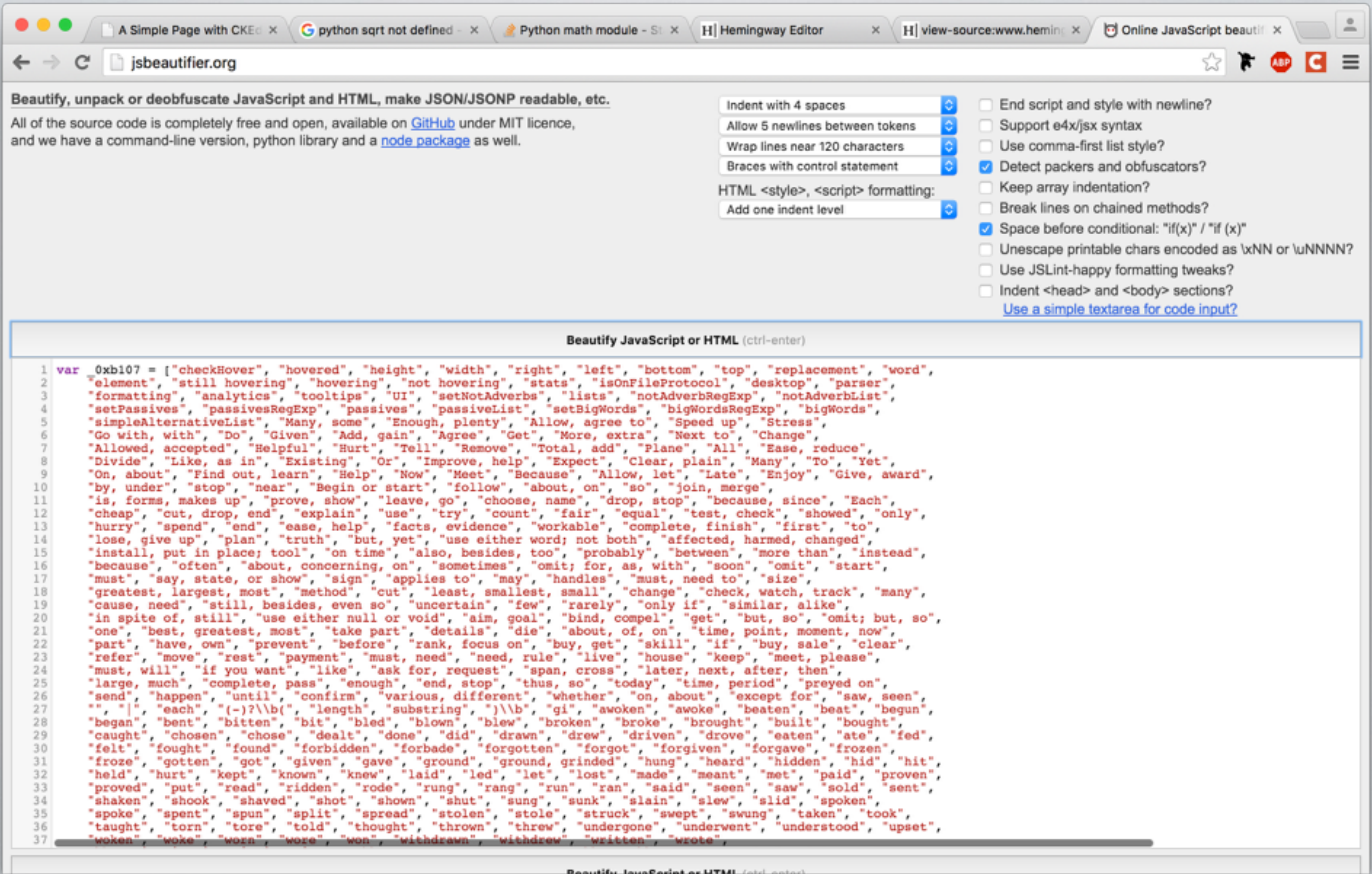
1 of 11 sentences is very hard to read.

1 phrase has a simpler alternative.

1 adverb. Remove it.

1 use of passive voice. Aim for 2 or fewer.

CURRENT STATE



jsbeautifier.org

Beautify, unpack or deobfuscate JavaScript and HTML, make JSON/JSONP readable, etc.

All of the source code is completely free and open, available on GitHub under MIT licence, and we have a command-line version, python library and a node package as well.

- Indent with 4 spaces
- Allow 5 newlines between tokens
- Wrap lines near 120 characters
- Braces with control statement

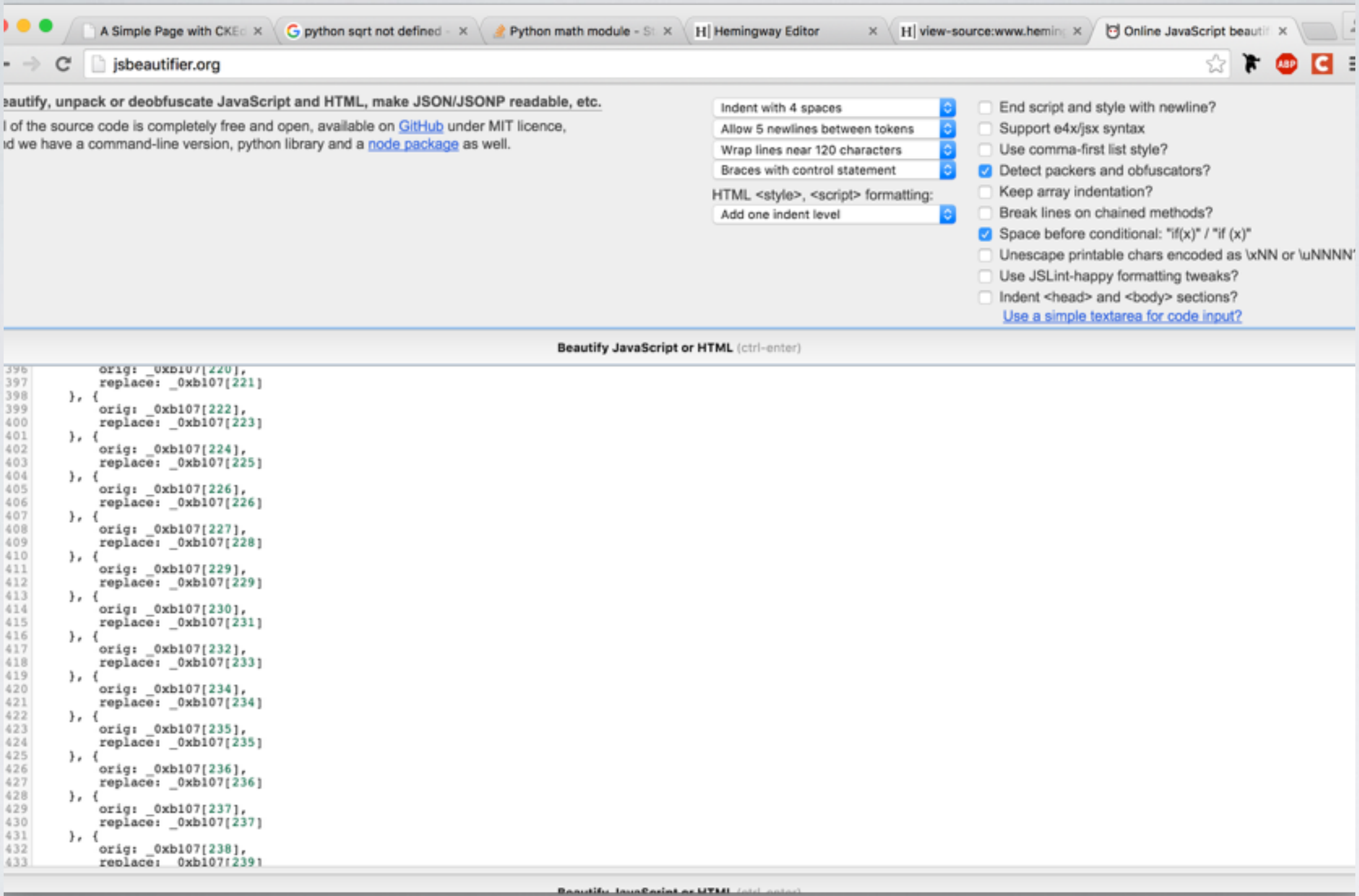
HTML <style>, <script> formatting:
- Add one indent level

- [ ] End script and style with newline?
- [ ] Support e4x/jsx syntax
- [ ] Use comma-first list style?
- [x] Detect packers and obfuscators?
- [ ] Keep array indentation?
- [ ] Break lines on chained methods?
- [x] Space before conditional: "if(x)" / "if (x)"
- [ ] Unescape printable chars encoded as \xNN or \uNNNN?
- [ ] Use JSLint-happy formatting tweaks?
- [ ] Indent <head> and <body> sections?

Use a simple textarea for code input?

Beautify JavaScript or HTML (ctrl-enter)

```
var _0xb107 = ["checkHover", "hovered", "height", "width", "right", "left", "bottom", "top", "replacement", "word",
    "element", "still hovering", "hovering", "not hovering", "stats", "isOnFileProtocol", "desktop", "parser",
    "formatting", "analytics", "tooltips", "UI", "setNotAdverbs", "lists", "notAdverbRegExp", "notAdverbList",
    "setPassives", "passivesRegExp", "passives", "passiveList", "setBigWords", "bigWordsRegExp", "bigWords",
    "simpleAlternativeList", "Many, some", "Enough, plenty", "Allow, agree to", "Speed up", "Stress",
    "Go with, with", "Do", "Given", "Add, gain", "Agree", "Get", "More, extra", "Next to", "Change",
    "Allowed, accepted", "Helpful", "Hurt", "Tell", "Remove", "Total, add", "Plane", "All", "Ease, reduce",
    "Divide", "Like, as in", "Existing", "Or", "Improve, help", "Expect", "Clear, plain", "Many", "To", "Yet",
    "On, about", "Find out, learn", "Help", "Now", "Meet", "Because", "Allow, let", "Late", "Enjoy", "Give, award",
    "by, under", "stop", "near", "Begin or start", "follow", "about, on", "so", "join, merge",
    "is, forms, makes up", "prove, show", "leave, go", "choose, name", "drop, stop", "because, since", "Each",
    "cheap", "cut, drop, end", "explain", "use", "try", "count", "fair", "equal", "test, check", "showed", "only",
    "hurry", "spend", "end", "ease, help", "facts, evidence", "workable", "complete, finish", "first", "to",
    "lose, give up", "plan", "truth", "but, yet", "use either word; not both", "affected, harmed, changed",
    "install, put in place; tool", "on time", "also, besides, too", "probably", "between", "more than", "instead",
    "because", "often", "about, concerning, on", "sometimes", "omit; for, as, with", "soon", "omit", "start",
    "must", "say, state, or show", "sign", "applies to", "may", "handles", "must, need to", "size",
    "greatest, largest, most", "method", "cut", "least, smallest, small", "change", "check, watch, track", "many",
    "cause, need", "still, besides, even so", "uncertain", "few", "rarely", "only if", "similar, alike",
    "in spite of, still", "use either null or void", "aim, goal", "bind, compel", "get", "but, so", "omit; but, so",
    "one", "best, greatest, most", "take part", "details", "die", "about, of, on", "time, point, moment, now",
    "part", "have, own", "prevent", "before", "rank, focus on", "buy, get", "skill", "if", "buy, sale", "clear",
    "refer", "move", "rest", "payment", "must, need", "need, rule", "live", "house", "keep", "meet, please",
    "must, will", "if you want", "like", "ask for, request", "span, cross", "later, next, after, then",
    "large, much", "complete, pass", "enough", "end, stop", "thus, so", "today", "time, period", "preyed on",
    "send", "happen", "until", "confirm", "various, different", "whether", "on, about", "except for", "saw, seen",
    "", "|", "each", "(-)?\\b(", "length", "substring", ")\\b", "gi", "awoken", "awoke", "beaten", "beat", "begun",
    "began", "bent", "bitten", "bit", "bled", "blown", "blew", "broken", "broke", "brought", "built", "bought",
    "caught", "chosen", "chose", "dealt", "done", "did", "drawn", "drew", "driven", "drove", "eaten", "ate", "fed",
    "felt", "fought", "found", "forbidden", "forbade", "forgotten", "forgot", "forgiven", "forgave", "frozen",
    "froze", "gotten", "got", "given", "gave", "ground", "ground, grinded", "hung", "heard", "hidden", "hid", "hit",
    "held", "hurt", "kept", "known", "knew", "laid", "led", "let", "lost", "made", "meant", "met", "paid", "proven",
    "proved", "put", "read", "ridden", "rode", "rung", "rang", "run", "ran", "said", "seen", "saw", "sold", "sent",
    "shaken", "shook", "shaved", "shot", "shown", "shut", "sung", "sunk", "slain", "slew", "slid", "spoken",
    "spoke", "spent", "spun", "split", "spread", "stolen", "stole", "struck", "swept", "swung", "taken", "took",
    "taught", "torn", "tore", "told", "thought", "thrown", "threw", "undergone", "underwent", "understood", "upset",
    "woken", "woke", "worn", "wore", "won", "withdrawn", "withdrew", "written", "wrote",
```



eautify, unpack or deobfuscate JavaScript and HTML, make JSON/JSONP readable, etc.

l of the source code is completely free and open, available on GitHub under MIT licence,
d we have a command-line version, python library and a node package as well.

Indent with 4 spaces
Allow 5 newlines between tokens
Wrap lines near 120 characters
Braces with control statement
HTML <style>, <script> formatting:
Add one indent level

- [ ] End script and style with newline?
- [ ] Support e4x/jsx syntax
- [ ] Use comma-first list style?
- [x] Detect packers and obfuscators?
- [ ] Keep array indentation?
- [ ] Break lines on chained methods?
- [x] Space before conditional: "if(x)" / "if (x)"
- [ ] Unescape printable chars encoded as \xNN or \uNNNN
- [ ] Use JSLint-happy formatting tweaks?
- [ ] Indent <head> and <body> sections?

Use a simple textarea for code input?

Beautify JavaScript or HTML (ctrl-enter)

```
            orig: _0xb107[220],
            replace: _0xb107[221]
        }, {
            orig: _0xb107[222],
            replace: _0xb107[223]
        }, {
            orig: _0xb107[224],
            replace: _0xb107[225]
        }, {
            orig: _0xb107[226],
            replace: _0xb107[226]
        }, {
            orig: _0xb107[227],
            replace: _0xb107[228]
        }, {
            orig: _0xb107[229],
            replace: _0xb107[229]
        }, {
            orig: _0xb107[230],
            replace: _0xb107[231]
        }, {
            orig: _0xb107[232],
            replace: _0xb107[233]
        }, {
            orig: _0xb107[234],
            replace: _0xb107[234]
        }, {
            orig: _0xb107[235],
            replace: _0xb107[235]
        }, {
            orig: _0xb107[236],
            replace: _0xb107[236]
        }, {
            orig: _0xb107[237],
            replace: _0xb107[237]
        }, {
            orig: _0xb107[238],
            replace: _0xb107[239]
```

Beautify JavaScript or HTML (ctrl-enter)

THE NOTION OF READABILITY

- What makes text readable?
- Is there a metric to capture readability?

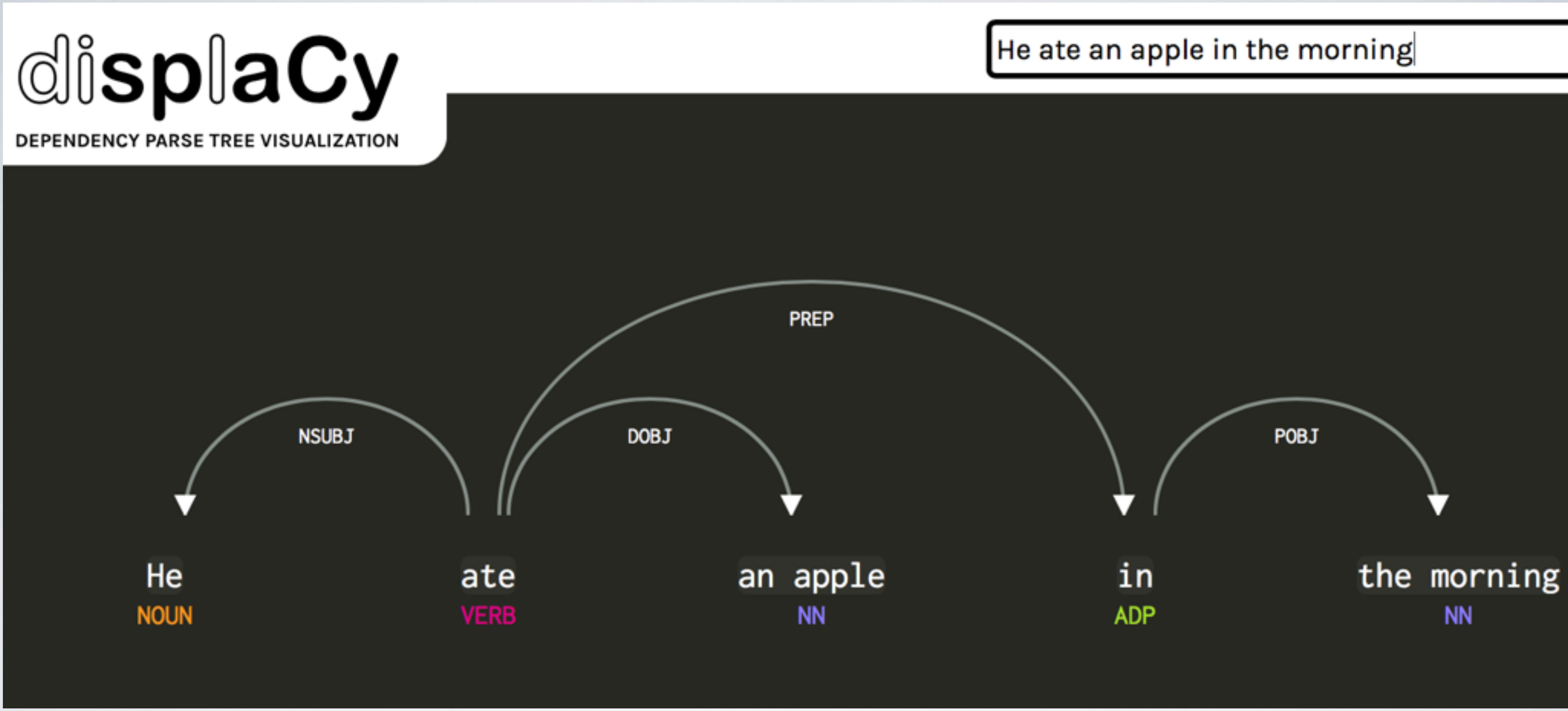
APPROACHES

- Word Frequency: unigram probability for a word to appear.
- More frequent words => difficult to read

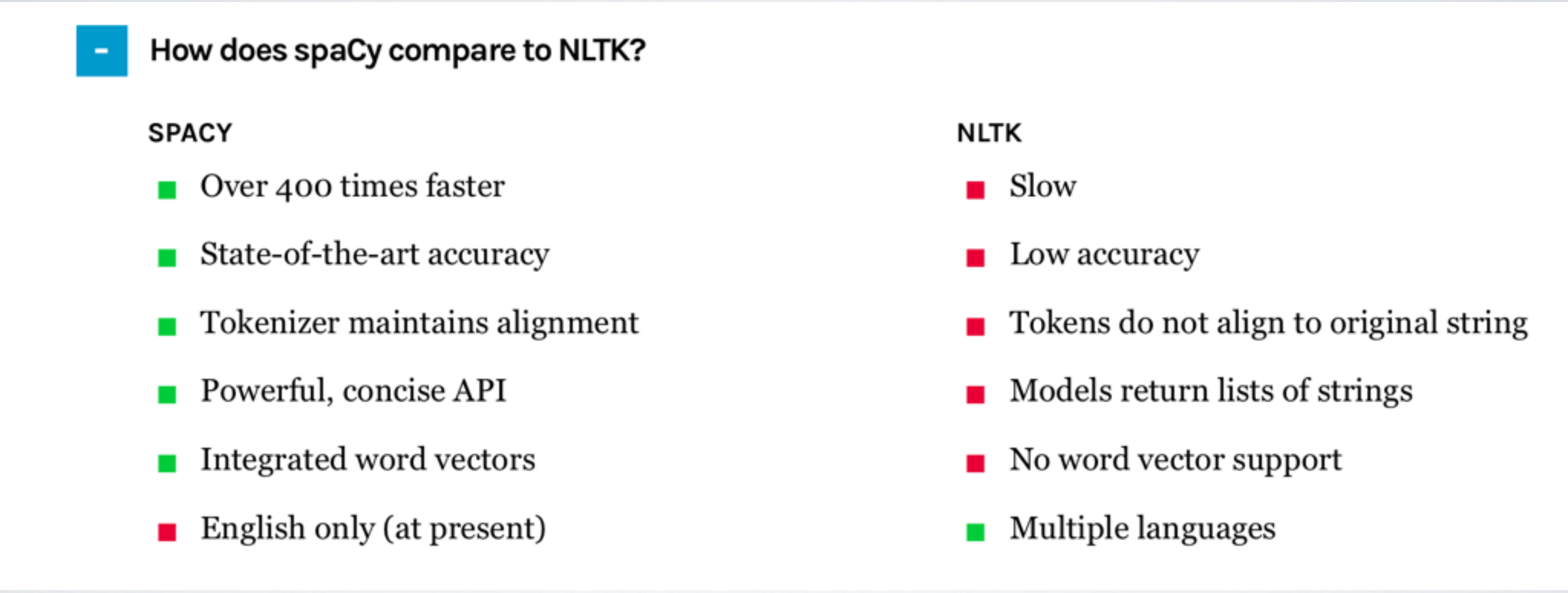
APPROACHES

- Structural : Words with deeper, ‘more complicated’ trees tend to be harder to read.

  Ex: “He, in the morning, ate an apple” ($p=0.04$) is more complex than
  “He ate an apple in the morning” ($p=0.019$)


displaCy
DEPENDENCY PARSE TREE VISUALIZATION
He ate an apple in the morning
NSUBJ
DOBJ
PREP
POBJ
He
NOUN
ate
VERB
an apple
NN
in
ADP
the morning
NN

SPLACY.IO



How does spaCy compare to NLTK?

SPACY	NLTK
🟩 Over 400 times faster	🟥 Slow
🟩 State-of-the-art accuracy	🟥 Low accuracy
🟩 Tokenizer maintains alignment	🟥 Tokens do not align to original string
🟩 Powerful, concise API	🟥 Models return lists of strings
🟩 Integrated word vectors	🟥 No word vector support
🟥 English only (at present)	🟩 Multiple languages

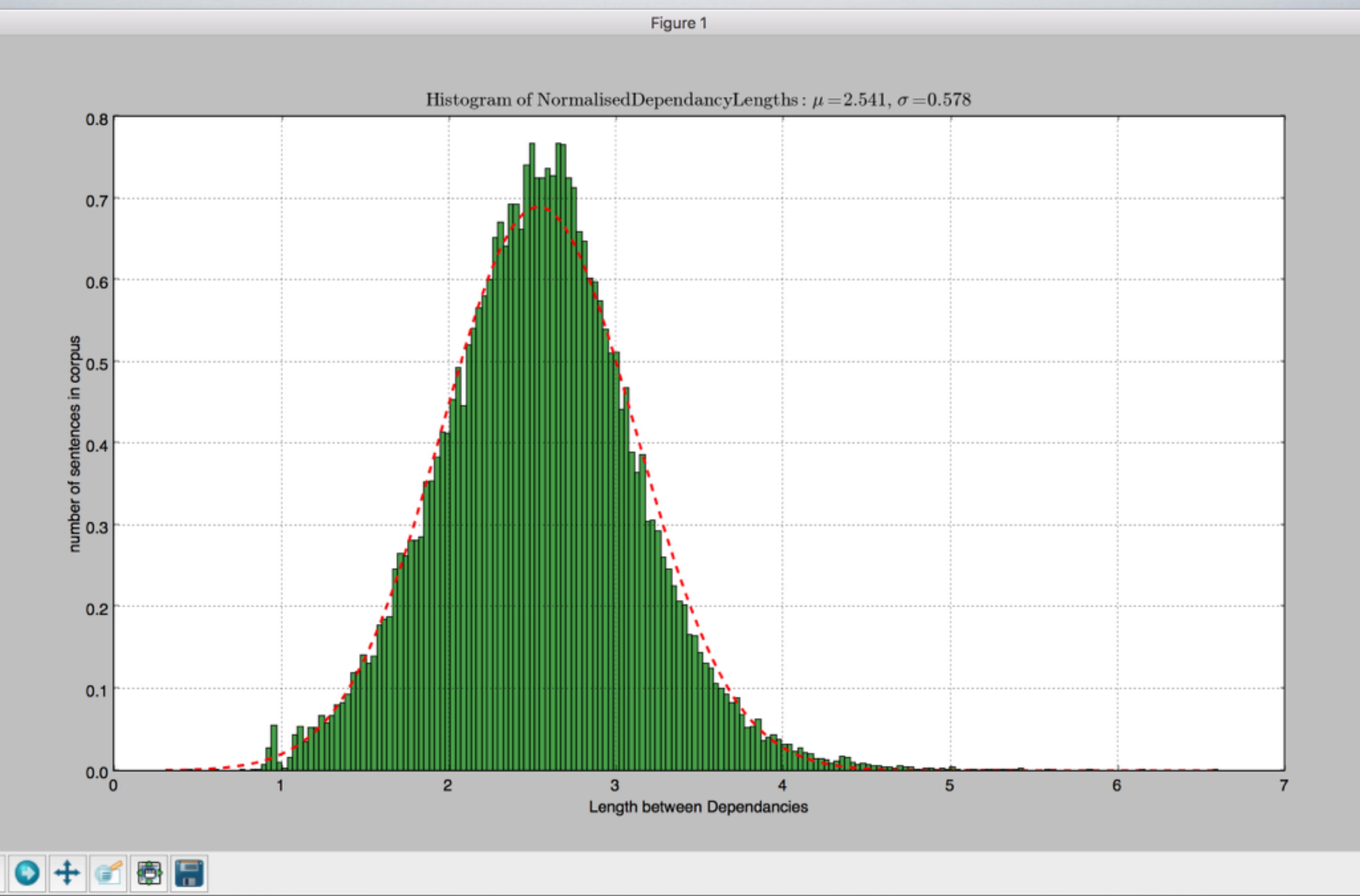


Figure 1
Histogram of NormalisedDependancyLengths : $\mu = 2.541$, $\sigma = 0.578$
number of sentences in corpus
Length between Dependancies
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
1
2
3
4
5
6
7