Proselint: the linting of science prose, and the science of prose linting

Michael D. Pacer and Jordan W. Suchow

Writing is notoriously hard, even for the best writers, and it's not for lack of good advice — a tremendous amount of knowledge about the craft is strewn across usage guides, dictionaries, technical manuals, essays, pamphlets, websites, and the hearts and minds of great authors and editors. But poring over Strunk & White hardly makes one a better writer — it turns you into neither Strunk nor White. And nobody has the willpower, time, or memory to manually apply all the advice from Garner's Modern English Usage (a 1,120-page usage guide) to everything they write. The knowledge is trapped, waiting to be extracted and transformed.

We built Proselint, a Python-based linter for prose. (A linter is a computer program that, like a spell checker, scans through a document and analyzes it.) Proselint identifies violations of expert style and usage guidelines. Proselint is open-source software released under the BSD license and works with Python 2 and 3. It runs efficiently as a command-line utility or editor plugin. It outputs advice in standard formats (e.g., JSON), integrating with Sublime Text, Atom, Vim, Emacs, and other editors and services. Though in its infancy – perhaps 2% of what it could be – Proselint already includes modules on a variety of usage problems: redundancy, jargon, illogic, clichés, sexism, misspelling, inconsistency, misuse of symbols, malapropisms, oxymorons, security gaffes, hedging, apologizing, pretension, and more.

Proselint can be seen as both a language tool for scientists and a tool for language science. On the one hand, it can be used to improve writing, and it includes modules that promote clear and consistent prose in science writing. On the other, it can measure language usage and explore the factors relevant to creating a useful linter.

Proselint is unlike other language linters. First, Proselint does not focus on grammar, which is AI-complete, requiring human-level intelligence to get right. Instead, we consider usage and style. Second, existing tools for improving prose raise so many false alarms that their advice is distrusted and ignored. Proselint's motto is 'Better to be silent than wrong', aiming for a precision that makes it possible to adopt its recommendations unquestioningly. We optimize a "lintscore" metric that penalizes false positives.

Proselint is a massive undertaking, one that will require the ethos of an open source community to complete. Garner's book alone has 11,000 entries. Half are easy, assignable as a homework problem (e.g., that "very unique" compares an uncomparable adjective, or that people from Michigan prefer to be called "Michiganders", not "Michiganians"). Thirty percent are moderately challenging, requiring custom tooling. Fifteen percent are hard — projects that require advances in AI and NLP. Everything else, around five percent (the best five percent), is AI-complete.

We will discuss where Proselint is and where it is heading. We will show its installation and application, demonstrating its use on the repository of papers submitted to SciPy2016.

Proselint is fertile ground for growing an open-source community. It has trivial subproblems and lofty goals, an immediate impact and a long future.

http://proselint.com

https://github.com/amperser/proselint