Supplemental technical report

Anonymous

January 2020

This document contains supplemental technical information for the paper "A fast filtering algorithm for massive context free grammars."

1 Supplement to theoretical analysis of the TTF

2 Supplement to experiment three

The asymptotic relationship between run time of the terminal-tree filter and the number of rules in the filtered subgrammar appears roughly linear for our example grammar, supporting the argument that, in expectation, the run time is linear in the size of the subgrammar, though we lack a sufficiently large number of samples from the language to rigorously test this hypothesis.

The test CFG for this experiment was produced by converting a Systemic Functional Grammar [**?**] of English based on Edouard Hovy's SFG from the Penman project [**?**]. The resulting grammar has $|P| \approx 13,000,000$ and $|G| \approx 115$ million.

The resulting grammar has around 13 million rules. The total grammar size $|G|$, measured by the sums of the lengths of the right hand sides is $115,738,333$. Compare this to the two grammars under study by [**?**] which had $\approx 500,000$ rules each and total size 1 million and 12 million, respectively. The structure of the Halliday grammar is relatively flat and much of the rules arise from different ways of permuting otherwise equivalent RHSs. In particular, there are $8,975,867$ unique RHSs and $35,228$ unique sets of RHS elements. A consequence of this structure is that it is particularly well suited to both length-based filtering when input sentences are short (a basis for filtering not mentioned in [**?**] because their study permits grammars with ϵ-productions, while we ours does not) and b-filtering. Additionally, this structure makes the tree bear a structure much more similar to the grammar in experiment 2, than the one in experiment 1, in particular it should be closer to a full binary tree and should have few long right-branching chains.

Note that length-based filtering can not be performed on a grammar containing empty productions - a grammar where nonterminals don't necessarily yield at least one output symbol - because there is no certain relationship between the number of nonterminals on the RHS of a rule and the length of strings

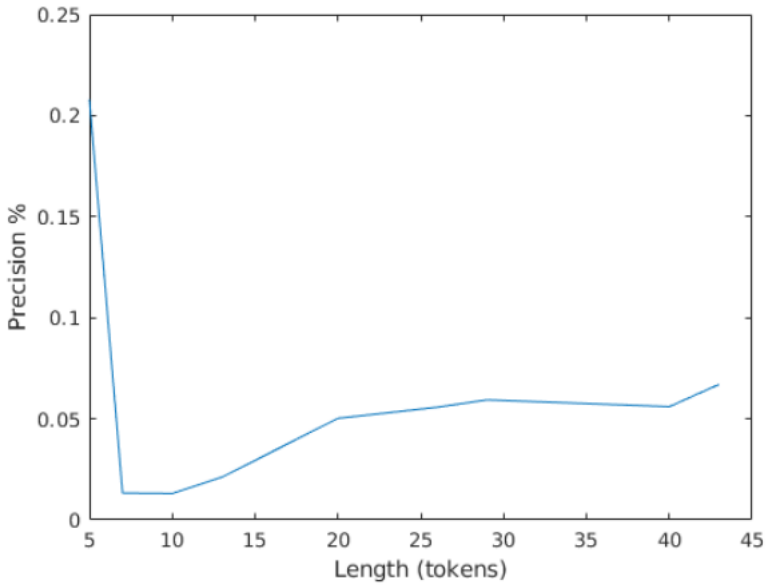

Figure 1: Precision expressed as % as measured by number of rules in the gold-standard divided by number of rules in the subgrammar.

generable from it. The grammar (and its accompanying vocabulary) we have constructed is incomplete and can't handle all possible English constructions. For that reason, the performance is analyzed on a handpicked set of 10 sentences which can be found in the technical report on our github.

In order to get a sense of how many rules of this grammar are truly applicable for each input, we filter first by length, then by content, and finally, by removing nonterminals which were made unproductive or unreachable by the previous two steps of filtering. Below is a graph relating length of the input to the number of rules left after all stages of filtering.

As you can see, our grammars respond incredibly well to content filtering, achieving a factor of 100 reduction in grammar size on even the longest test sentence.

- the distributor registered this domain
- this domain was registered by the distributor
- this domain in the account was registered by the distributor
- this domain in the account was registered by the distributor in the domain
- the use of the nicknames in the decryption key is evidence that the account was the distributor of these samples

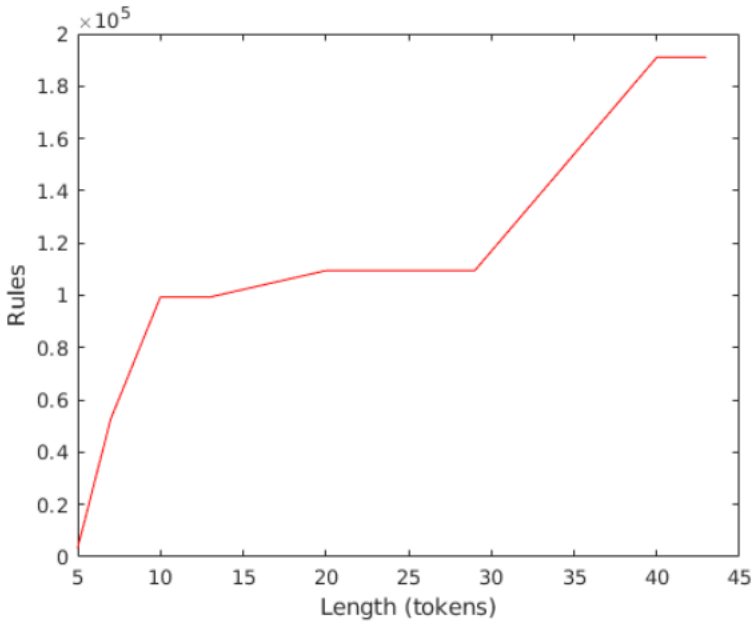

Figure 2: Number of rules resulting after content-filtering. Content-based filtering as exemplified in [**?**]'s b-filtering and our terminal-tree filtering accomplishes close to 100 fold reduction of grammar size on sentences 45 words long

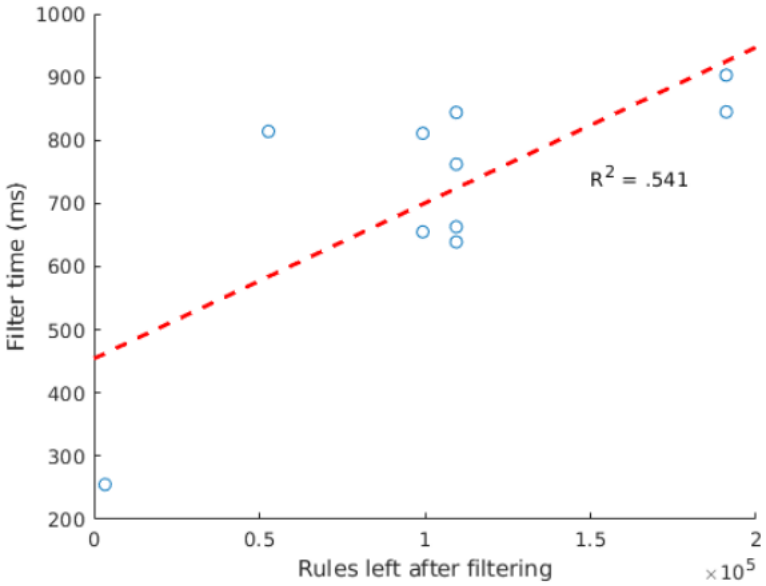

Figure 3: Number of rules resulting after content-filtering. Content-based filtering as exemplified in [**?**]'s b-filtering and our terminal-tree filtering accomplishes close to 100 fold reduction of grammar size on sentences 45 words long

- the use of the nicknames of this domain in the decryption key is evidence that the account was the distributor of these samples
- the use of the nicknames of this domain in the decryption key in the record is evidence that the account was the distributor of these samples
- the use of the nicknames of this domain in the decryption key in the record of the activity is evidence that the account was the distributor of the software
- the use of the nicknames of this domain in the decryption key in the record of the activity is evidence that the account was the distributor of the software and that it was registered by the communication of the distributors
- the use of the nicknames of this domain in the decryption key in the record of the activity is evidence that the account was the distributor of the software and that it was registered by the communication of the distributors in these discussions