Perturbation learning for general-purpose text validation

Anonymous NAACL submission

Abstract

Language learners and generative models alike are often in need of text validation: checking how natural a certain sentence sounds within a given language or style. In this paper, we propose an approach to training a statistical validation model on a text corpus with no supervision. This is achieved by applying random perturbations to sentences from the corpus and training a recurrent neural network to discriminate between the original sentences and the perturbed ones. Choosing the right perturbation model, however, is far from trivial: the resulting validation model has to generalize beyond the specific perturbation we introduced and be able to recognize previously unseen kinds of deviations from the norm it learned from the corpus. We develop several perturbation models, demonstrate and compare their generalization ability.

1 Background

2 Related work

3 Methodology

We hypothesise that a discriminator trained to detect sentences that have been randomly perturbed can generalize to perform text validation. To that end, we introduce several *perturbation models*.

3.1 Word-level perturbations

3.2 Character-level perturbations

3.3 Word-form perturbations

This kind of perturbation is performed using `pymorphy2` (Korobov, 2015) and includes two types of transformations, based on morphological analysis and generation.

- During *random lemmatization*, each token in a sentence is either lemmatized with some probability (we use 50% probability) or left as it is.
- *Random inflection* is similar to *random lemmatization*, but instead of replacing a token with its normal form, we take some other grammatical form of this word. For nouns, adjectives and personal pronouns, we randomly change case; for verbs, person is changed. Tokens with other parts of speech remain unchanged.

The two types of token transformation are applied separately and form two different training sets.

3.4 Markov chain perturbations

This type of perturbations differs from others in that instead of doing changes to an initially grammatical sentence, we train a generative n-gram language model to produce some ill-formed sentences. To create the language model, we used the `markovfy` [1] implementation of Markov chain.

It is worth noting that not all of the sentences generated by markov chain are ungrammatical, but a significant part of them is, since the n-gram model cannot see further than n tokens into the past. In order to increase the number of ungrammatical sentences generated by the model we suppress any generated sentences that exactly overlap the original text by 50% of the sentence's word count.

3.5 Adversarial perturbations

4 Experimental setup

5 Results

References

Mikhail Korobov. 2015. Morphological analyzer and generator for russian and ukrainian languages. In Mikhail Yu. Khachay, Natalia Konstantinova,

[1] https://github.com/jsvine/markovify

Alexander Panchenko, Dmitry I. Ignatov, and Valeri G. Labunets, editors, *Analysis of Images, Social Networks and Texts*, volume 542 of *Communications in Computer and Information Science*, pages 320–332. Springer International Publishing.