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

Text validation is the problem of discriminating between text that naturally belongs to a certaing domain (language or a subdomain of language, such as a certain author's style) and .

Neural network-based approaches have the additional benefit that the validation function $f(s)$ (s - sentence) is differentiable ($\frac{df}{ds}$ can be easily calculated) and thus can be used as perceptual loss (Johnson et al., 2016) to train a generative neural network that outputs natural-sounding text.

2 Methodology

We hypothesise that there exists a mechanism of applying random perturbations to sentences such that a discriminator trained to detect sentences that have been perturbed from intact ones can be used to detect mistakes more generally. To that end, we introduce several *perturbation models*. For each of them, we train a binary classifier (*validation model*), test its performance on a holdout validation dataset and then on the datasets used to train other *validation models*. Our hypothesis can be considered confirmed if a *validation model* trained with *perturbation model* correctly detect sentences modified with other *perturbation models*.

2.1 Perturbation models

2.1.1 Word-order perturbations

The first model we employ is *random word flip*: a randomly selected word in the sentence is moved to a randomly selected location in the sentence. All words and locations have equal probability to be selected.

Shuffle perturbation means reordering the entire sentence according to a random permutation.

2.1.2 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.

2.1.3 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.

2.2 Validation model

3 Experimental setup

4 Results

References

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer.

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

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