Explanation of the Filtering Pipeline

Tuesday 8th February, 2022

The filtering pipeline consists of two steps.

First, the documents will be modified to essentially remove excessively long or incorrect words (links, for example).

Second, on these newly modified documents, a filtering is performed to say whether or not the document is kept.

For each method presented in these two parts, one can, depending on the language, choose to use it or not.

The order of the filters is also indifferent and the methods are commutative.

All filters work with cutoffs or parameters, which are user-defined depending on the language in this file.

The code is available here and most methods are defined in this file.

Contents

1 Modification of documents

1.1 Whitespace standardization

There are many characters for whitespace. This method converts all whitespace in the sentence to the same whitespace, which is the classic space.

This method will later facilitate the segmentation of the document into words, which we will need for many methods.

1.2 Remove long words

To be able to extract the words of a document, and to rebuild it thereafter in a reversible way, one starts by splitting the document according to the new line character `"\n"`, then for each element split according to the tabulation `"\t"`, then for each sub-element split according to the whitespace `" "`. We can then perform the filtering we want on the words obtained, then reconstruct the document by joining the list in the opposite direction.

To know if a word is kept or not, we start by stripping this word according to the special characters (by removing on the left and on the right of the word all possible special characters). Thus, for the word `situation,` (with a comma), we obtain the word `situation` (without a comma) after strip.

If the length of the word obtained after stripping it is greater than the specified cutoff manually defined by the user, the word is deleted.

Of course, the stripping of words on special characters is only useful to decide whether to keep a word or not, but when reconstructing the sentence after the word filtering, it is the original words and not the stripped words that are considered.

This method is not adaptable to Chinese, which does not include spaces between words.

1.3 Remove words with incorrect substrings

For this method, we use exactly the same strategy as for the previous method to extract the words and reconstruct the sentence. Only the filtering strategy changes.

Here, we decide to remove a word if it contains any substring that we consider incorrect. These substrings are defined manually by the user, and are by default `["http", "www", ".com", "href", "//"]`. The goal is to remove links and words related to the source code of the page.

This method is not adaptable to Chinese, which does not include spaces between words.

2 Filtering of documents

2.1 Filtering on the number of words

We count the number of words in the sentence (separated by a line break, a tab or a whitespace). To estimate the number of words in a Chinese document, which does not include spaces between words, we perform a tokenization with a Sentencepiece model.

If this number is less than or greater than two cutoffs, the document is removed.

Documents that are too short are very often incorrect sentences, or contain no context for a model to learn correctly.

2.2 Filtering on the character repetition ratio

First of all, we have to choose the length of the repetitions on characters, noted n, which is an integer that will parameterize the filter.

For a document, we take the list of **character** n-grams, with n the length of the repetitions on characters, and we count their frequencies.

By noting $n_{n\text{-grams}}$ the number of different n-grams found in the document, we define $n_{\text{rep-}n\text{-grams}} := \left\lfloor \sqrt{n_{n\text{-grams}}} \right\rfloor$.

The character repetition ratio is then defined as the ratio of the sum of the frequencies of the $n_{\text{rep-}n\text{-grams}}$ most frequent n-grams by the sum of the frequencies of all n-grams.

If a document has a character repetition ratio greater than a certain cutoff, it is removed.

Note Choosing a higher or lower value for n will not indicate stronger or weaker filtering. In constructing this filter, two methods were initially tested.
The first was to calculate the character repetition ratio as the frequency of the most frequent n-gram over the frequency of all n-grams. The problem is that short sentences were much more likely to have a high character repetition ratio, since the most frequent n-gram represents a larger proportion of the sentence.
The second method was to calculate the character repetition ratio as the ratio of the sum of the frequencies of all n-grams with a frequency of at least 2 to the sum of the frequencies of all n-grams. The problem this time is that very long documents, but not necessarily including repetitions, tended to have a high character repetition ratio, since these texts inherently have a large diversity of n-grams. It is dangerous to remove such documents, since they can be entire book chapters written without any grammatical errors, and therefore very useful for training the model.
We therefore chose an intermediate method, trying to find a normalization function (here the square root which works good in practice) allowing to reduce this effect.
When choosing the parameters, we must also remember that taking a lower n will still favor larger character repetition ratio for long documents (despite the normalization function described above that helps to reduce this effect), which is not intended. It is generally better to increase n, so that false positives are short documents (which we want to discard anyway) rather than long ones. However, a low n can be useful for Chinese, where a character can designate a whole word.

2.3 Filtering on the word repetition ratio

First of all, we have to choose the length of the repetitions on words, noted n, which is an integer that will parameterize the filter.

For a document, we take the list of **word** n-grams, with n the length of the repetitions on words, and we count their frequencies.

The word repetition ratio is then defined as the ratio of the sum of the frequencies of all n-grams with a frequency of at least 2 to the sum of the frequencies of all n-grams.

If a document has a word repetition ratio greater than a certain cutoff, it is removed.

Note Contrary to the filter on the character repetition ratios, we did not find a bias of this method giving systematically higher or lower scores to longer or shorter documents.

2.4 Filtering on the special characters ratio

Special characters are defined in this file. If a document has a special characters ratio greater than a certain cutoff, it is removed.

Some documents do not contain any correct sentences and mostly special characters. This filter allows you to remove them.

2.5 Filtering on the stop words ratio

We made a list of stop words for each language, available in this file. If the stop words ratio for a document is higher than a certain cutoff, it is removed.

For Chinese, which does not include spaces between words, we have to perform a tokenization to obtain subunits. However, these subunits do not necessarily form whole words, and thus may never be considered as a stop word and thus never contribute to the ratio. This same problem occurs for Vietnamese, which includes spaces between syllables. We have for example the stop word `biết bao nhiêu`. If we consider separately `biết`, then `bao`, then `nhiêu` as three different words, we will not be able to identify this stop word in a sentence. To solve this problem, for these two languages, we increase the set of words of the considered document by adding groups of two and three consecutive sub-units, joined by a space for Vietnamese and without for Chinese. This technique is also used for the filtering on the flagged words ratio in a following section.

This filter is the best method to remove automatically generated sentences by computers that do not make sense as a whole.

2.6 Filtering on the flagged words ratio

We have defined a list of flagged words for the different languages in this file.

In more detail, here is the methodology used to build a list of flagged words for a language.
Make a list of the most frequently words that appear in pornographic materials for this language.
Keep only the words associated with porn and systematically used in a sexual context.
Remove words that can be used in medical (sex, sexual), scientific (ejaculation, erection), colloquial (without referring systematically to porn) (boobs, bitch, cock, pussy, fuck), or everyday contexts (suck, swallow). Remove all insults (motherfucker, dickhead). Remove all words referring to race (white, black) or sexual orientation (gay, lesbian).

In order to build the initial list of words that frequently appear in pornographic documents, one must build a database of pornographic documents. There are two ways to do this.
The first one is to find such a database on the internet, or to build one by extracting texts from pornographic sites via a list of URLs of pornographic sites.
The second one, which is the one used for the construction of the list for English, is a circular method. We start from a pre-existing (even if not perfect) list of flagged words found on the internet. We use it to build our database containing the documents with the highest flagged words ratio.

Since English flagged words are regularly present in documents of other languages, we have also added them to those of all other languages.

For languages other than English, the lists have not yet been checked manually, and we will need native speakers to filter them in the same way as we did for English.

The purpose of this filter is not to remove erotic texts, but to remove the numerous documents consisting of an accumulation of buzzwords centered on porn, most often without having any cohesion within the same sentence, which would harmful for the learning of the model.

2.7 Filtering on the language identification prediction score

We use the fastText model to quickly obtain an estimate of the language of a document, along with a prediction confidence score. If this score is below a certain cutoff, the document is removed.

This filter removes documents in which the spoken language changes several times, as well as documents of another language, which cannot be correctly analyzed by filters that have been specified with parameters related to a particular language.

2.8 Filtering on the perplexity score

We use the KenLM models trained by Facebook on each language on Wikipedia to obtain the perplexity scores of the documents. If the perplexity score is above a certain cutoff, the document is removed.

The purpose of this filter is not necessarily to remove a large number of documents. Indeed, documents with a different structure from Wikipedia, or with a more familiar language or on the contrary very technical on specialized subjects, would be unfairly penalized. The goal here is to remove outliers which are mainly documents with unrelated words, for example a list of tags, information related to the extraction of the text of the page (date and time) or multiple repetitions of the same sentences.

3 Playing with the filtering parameters

The Spaces text-data-filtering is a demo that allows you to play with the filtering parameters to filter 5000 English documents and get familiar with the pipeline.

Once these parameters are defined, a tool allows you to enter your own document, returns its statistics, and indicates whether the document is discarded or not.

It is possible to increase the number of documents (up to 15000) and to use it for other languages. To do this, please run this code on your computer.