**Text classification from scratch**

**Authors:** Mark Omernick, Francois Chollet
**Date created:** 2019/11/06
**Last modified:** 2020/05/17
**Description:** Text sentiment classification starting from raw text files.

## **Introduction**

This example shows how to do text classification starting from raw text (as a set of text files on disk). We demonstrate the workflow on the IMDB sentiment classification dataset (unprocessed version). We use the TextVectorization layer for word splitting & indexing.

## **Setup**

import tensorflow as tf

import numpy as np

## Load the data: IMDB movie review sentiment classification

Let's download the data and inspect its structure.

!curl -O https://ai.stanford.edu/~amaas/data/sentiment/aclImdb\_v1.tar.gz

!tar -xf aclImdb\_v1.tar.gz

 % Total % Received % Xferd Average Speed Time Time Time Current

 Dload Upload Total Spent Left Speed

100 80.2M 100 80.2M 0 0 45.3M 0 0:00:01 0:00:01 --:--:-- 45.3M

The aclImdb folder contains a train and test subfolder:

!ls aclImdb

!ls aclImdb/test

!ls aclImdb/train

README imdb.vocab imdbEr.txt [34mtest[m[m [34mtrain[m[m

labeledBow.feat [34mneg[m[m [34mpos[m[m urls\_neg.txt urls\_pos.txt

labeledBow.feat [34mpos[m[m unsupBow.feat urls\_pos.txt

[34mneg[m[m [34munsup[m[m urls\_neg.txt urls\_unsup.txt

The aclImdb/train/pos and aclImdb/train/neg folders contain text files, each of which represents on review (either positive or negative):

!cat aclImdb/train/pos/6248\_7.txt

Being an Austrian myself this has been a straight knock in my face. Fortunately I don't live nowhere near the place where this movie takes place but unfortunately it portrays everything that the rest of Austria hates about Viennese people (or people close to that region). And it is very easy to read that this is exactly the directors intention: to let your head sink into your hands and say "Oh my god, how can THAT be possible!". No, not with me, the (in my opinion) totally exaggerated uncensored swinger club scene is not necessary, I watch porn, sure, but in this context I was rather disgusted than put in the right context.<br /><br />This movie tells a story about how misled people who suffer from lack of education or bad company try to survive and live in a world of redundancy and boring horizons. A girl who is treated like a whore by her super-jealous boyfriend (and still keeps coming back), a female teacher who discovers her masochism by putting the life of her super-cruel "lover" on the line, an old couple who has an almost mathematical daily cycle (she is the "official replacement" of his ex wife), a couple that has just divorced and has the ex husband suffer under the acts of his former wife obviously having a relationship with her masseuse and finally a crazy hitchhiker who asks her drivers the most unusual questions and stretches their nerves by just being super-annoying.<br /><br />After having seen it you feel almost nothing. You're not even shocked, sad, depressed or feel like doing anything... Maybe that's why I gave it 7 points, it made me react in a way I never reacted before. If that's good or bad is up to you!

We are only interested in the pos and neg subfolders, so let's delete the rest:

!rm -r aclImdb/train/unsup

You can use the utility tf.keras.preprocessing.text\_dataset\_from\_directory to generate a labeled tf.data.Dataset object from a set of text files on disk filed into class-specific folders.

Let's use it to generate the training, validation, and test datasets. The validation and training datasets are generated from two subsets of the train directory, with 20% of samples going to the validation dataset and 80% going to the training dataset.

Having a validation dataset in addition to the test dataset is useful for tuning hyperparameters, such as the model architecture, for which the test dataset should not be used.

Before putting the model out into the real world however, it should be retrained using all available training data (without creating a validation dataset), so its performance is maximized.

When using the validation\_split & subset arguments, make sure to either specify a random seed, or to pass shuffle=False, so that the validation & training splits you get have no overlap.

batch\_size = 32

raw\_train\_ds = tf.keras.preprocessing.text\_dataset\_from\_directory(

 "aclImdb/train",

 batch\_size=batch\_size,

 validation\_split=0.2,

 subset="training",

 seed=1337,

)

raw\_val\_ds = tf.keras.preprocessing.text\_dataset\_from\_directory(

 "aclImdb/train",

 batch\_size=batch\_size,

 validation\_split=0.2,

 subset="validation",

 seed=1337,

)

raw\_test\_ds = tf.keras.preprocessing.text\_dataset\_from\_directory(

 "aclImdb/test", batch\_size=batch\_size

)

print(

 "Number of batches in raw\_train\_ds: %d"

 % tf.data.experimental.cardinality(raw\_train\_ds)

)

print(

 "Number of batches in raw\_val\_ds: %d" % tf.data.experimental.cardinality(raw\_val\_ds)

)

print(

 "Number of batches in raw\_test\_ds: %d"

 % tf.data.experimental.cardinality(raw\_test\_ds)

)

Found 25000 files belonging to 2 classes.

Using 20000 files for training.

Found 25000 files belonging to 2 classes.

Using 5000 files for validation.

Found 25000 files belonging to 2 classes.

Number of batches in raw\_train\_ds: 625

Number of batches in raw\_val\_ds: 157

Number of batches in raw\_test\_ds: 782

Let's preview a few samples:

# It's important to take a look at your raw data to ensure your normalization

# and tokenization will work as expected. We can do that by taking a few

# examples from the training set and looking at them.

# This is one of the places where eager execution shines:

# we can just evaluate these tensors using .numpy()

# instead of needing to evaluate them in a Session/Graph context.

for text\_batch, label\_batch in raw\_train\_ds.take(1):

 for i in range(5):

 print(text\_batch.numpy()[i])

 print(label\_batch.numpy()[i])

b'I\'ve seen tons of science fiction from the 70s; some horrendously bad, and others thought provoking and truly frightening. Soylent Green fits into the latter category. Yes, at times it\'s a little campy, and yes, the furniture is good for a giggle or two, but some of the film seems awfully prescient. Here we have a film, 9 years before Blade Runner, that dares to imagine the future as somthing dark, scary, and nihilistic. Both Charlton Heston and Edward G. Robinson fare far better in this than The Ten Commandments, and Robinson\'s assisted-suicide scene is creepily prescient of Kevorkian and his ilk. Some of the attitudes are dated (can you imagine a filmmaker getting away with the "women as furniture" concept in our oh-so-politically-correct-90s?), but it\'s rare to find a film from the Me Decade that actually can make you think. This is one I\'d love to see on the big screen, because even in a widescreen presentation, I don\'t think the overall scope of this film would receive its due. Check it out.'

1

b'First than anything, I\'m not going to praise I\xc3\xb1arritu\'s short film, even I\'m Mexican and proud of his success in mainstream Hollywood.<br /><br />In another hand, I see most of the reviews focuses on their favorite (and not so) short films; but we are forgetting that there is a subtle bottom line that circles the whole compilation, and maybe it will not be so pleasant for American people. (Even if that was not the main purpose of the producers) <br /><br />What i\'m talking about is that most of the short films does not show the suffering that WASP people went through because the terrorist attack on September 11th, but the suffering of the Other people.<br /><br />Do you need proofs about what i\'m saying? Look, in the Bosnia short film, the message is: "You cry because of the people who died in the Towers, but we (The Others = East Europeans) are crying long ago for the crimes committed against our women and nobody pay attention to us like the whole world has done to you".<br /><br />Even though the Burkina Fasso story is more in comedy, there is a the same thought: "You are angry because Osama Bin Laden punched you in an evil way, but we (The Others = Africans) should be more angry, because our people is dying of hunger, poverty and AIDS long time ago, and nobody pay attention to us like the whole world has done to you".<br /><br />Look now at the Sean Penn short: The fall of the Twin Towers makes happy to a lonely (and alienated) man. So the message is that the Power and the Greed (symbolized by the Towers) must fall for letting the people see the sun rise and the flowers blossom? It is remarkable that this terrible bottom line has been proposed by an American. There is so much irony in this short film that it is close to be subversive.<br /><br />Well, the Ken Loach (very know because his anti-capitalism ideology) is much more clearly and shameless in going straight to the point: "You are angry because your country has been attacked by evil forces, but we (The Others = Latin Americans) suffered at a similar date something worst, and nobody remembers our grief as the whole world has done to you".<br /><br />It is like if the creative of this project wanted to say to Americans: "You see now, America? You are not the only that have become victim of the world violence, you are not alone in your pain and by the way, we (the Others = the Non Americans) have been suffering a lot more than you from long time ago; so, we are in solidarity with you in your pain... and by the way, we are sorry because you have had some taste of your own medicine" Only the Mexican and the French short films showed some compassion and sympathy for American people; the others are like a slap on the face for the American State, that is not equal to American People.'

1

b'Blood Castle (aka Scream of the Demon Lover, Altar of Blood, Ivanna--the best, but least exploitation cinema-sounding title, and so on) is a very traditional Gothic Romance film. That means that it has big, creepy castles, a headstrong young woman, a mysterious older man, hints of horror and the supernatural, and romance elements in the contemporary sense of that genre term. It also means that it is very deliberately paced, and that the film will work best for horror mavens who are big fans of understatement. If you love films like Robert Wise\'s The Haunting (1963), but you also have a taste for late 1960s/early 1970s Spanish and Italian horror, you may love Blood Castle, as well.<br /><br />Baron Janos Dalmar (Carlos Quiney) lives in a large castle on the outskirts of a traditional, unspecified European village. The locals fear him because legend has it that whenever he beds a woman, she soon after ends up dead--the consensus is that he sets his ferocious dogs on them. This is quite a problem because the Baron has a very healthy appetite for women. At the beginning of the film, yet another woman has turned up dead and mutilated.<br /><br />Meanwhile, Dr. Ivanna Rakowsky (Erna Sch\xc3\xbcrer) has appeared in the center of the village, asking to be taken to Baron Dalmar\'s castle. She\'s an out-of-towner who has been hired by the Baron for her expertise in chemistry. Of course, no one wants to go near the castle. Finally, Ivanna finds a shady individual (who becomes even shadier) to take her. Once there, an odd woman who lives in the castle, Olga (Cristiana Galloni), rejects Ivanna and says that she shouldn\'t be there since she\'s a woman. Baron Dalmar vacillates over whether she should stay. She ends up staying, but somewhat reluctantly. The Baron has hired her to try to reverse the effects of severe burns, which the Baron\'s brother, Igor, is suffering from.<br /><br />Unfortunately, the Baron\'s brother appears to be just a lump of decomposing flesh in a vat of bizarre, blackish liquid. And furthermore, Ivanna is having bizarre, hallucinatory dreams. Just what is going on at the castle? Is the Baron responsible for the crimes? Is he insane? <br /><br />I wanted to like Blood Castle more than I did. As I mentioned, the film is very deliberate in its pacing, and most of it is very understated. I can go either way on material like that. I don\'t care for The Haunting (yes, I\'m in a very small minority there), but I\'m a big fan of 1960s and 1970s European horror. One of my favorite directors is Mario Bava. I also love Dario Argento\'s work from that period. But occasionally, Blood Castle moved a bit too slow for me at times. There are large chunks that amount to scenes of not very exciting talking alternated with scenes of Ivanna slowly walking the corridors of the castle.<br /><br />But the atmosphere of the film is decent. Director Jos\xc3\xa9 Luis Merino managed more than passable sets and locations, and they\'re shot fairly well by Emanuele Di Cola. However, Blood Castle feels relatively low budget, and this is a Roger Corman-produced film, after all (which usually means a low-budget, though often surprisingly high quality "quickie"). So while there is a hint of the lushness of Bava\'s colors and complex set decoration, everything is much more minimalist. Of course, it doesn\'t help that the Retromedia print I watched looks like a 30-year old photograph that\'s been left out in the sun too long. It appears "washed out", with compromised contrast.<br /><br />Still, Merino and Di Cola occasionally set up fantastic visuals. For example, a scene of Ivanna walking in a darkened hallway that\'s shot from an exaggerated angle, and where an important plot element is revealed through shadows on a wall only. There are also a couple Ingmar Bergmanesque shots, where actors are exquisitely blocked to imply complex relationships, besides just being visually attractive and pulling your eye deep into the frame.<br /><br />The performances are fairly good, and the women--especially Sch\xc3\xbcrer--are very attractive. Merino exploits this fact by incorporating a decent amount of nudity. Sch\xc3\xbcrer went on to do a number of films that were as much soft corn porn as they were other genres, with English titles such as Sex Life in a Woman\'s Prison (1974), Naked and Lustful (1974), Strip Nude for Your Killer (1975) and Erotic Exploits of a Sexy Seducer (1977). Blood Castle is much tamer, but in addition to the nudity, there are still mild scenes suggesting rape and bondage, and of course the scenes mixing sex and death.<br /><br />The primary attraction here, though, is probably the story, which is much a slow-burning romance as anything else. The horror elements, the mystery elements, and a somewhat unexpected twist near the end are bonuses, but in the end, Blood Castle is a love story, about a couple overcoming various difficulties and antagonisms (often with physical threats or harms) to be together.'

1

b"I was talked into watching this movie by a friend who blubbered on about what a cute story this was.<br /><br />Yuck.<br /><br />I want my two hours back, as I could have done SO many more productive things with my time...like, for instance, twiddling my thumbs. I see nothing redeeming about this film at all, save for the eye-candy aspect of it...<br /><br />3/10 (and that's being generous)"

0

b"Michelle Rodriguez is the defining actress who could be the charging force for other actresses to look out for. She has the audacity to place herself in a rarely seen tough-girl role very early in her career (and pull it off), which is a feat that should be recognized. Although her later films pigeonhole her to that same role, this film was made for her ruggedness.<br /><br />Her character is a romanticized student/fighter/lover, struggling to overcome her disenchanted existence in the projects, which is a little overdone in film...but not by a girl. That aspect of this film isn't very original, but the story goes in depth when the heated relationships that this girl has to deal with come to a boil and her primal rage takes over.<br /><br />I haven't seen an actress take such an aggressive stance in movie-making yet, and I'm glad that she's getting that original twist out there in Hollywood. This film got a 7 from me because of the average story of ghetto youth, but it has such a great actress portraying a rarely-seen role in a minimal budget movie. Great work."

1

## Prepare the data

In particular, we remove <br /> tags.

from tensorflow.keras.layers.experimental.preprocessing import TextVectorization

import string

import re

# Having looked at our data above, we see that the raw text contains HTML break

# tags of the form '<br />'. These tags will not be removed by the default

# standardizer (which doesn't strip HTML). Because of this, we will need to

# create a custom standardization function.

def custom\_standardization(input\_data):

 lowercase = tf.strings.lower(input\_data)

 stripped\_html = tf.strings.regex\_replace(lowercase, "<br />", " ")

 return tf.strings.regex\_replace(

 stripped\_html, "[%s]" % re.escape(string.punctuation), ""

 )

# Model constants.

max\_features = 20000

embedding\_dim = 128

sequence\_length = 500

# Now that we have our custom standardization, we can instantiate our text

# vectorization layer. We are using this layer to normalize, split, and map

# strings to integers, so we set our 'output\_mode' to 'int'.

# Note that we're using the default split function,

# and the custom standardization defined above.

# We also set an explicit maximum sequence length, since the CNNs later in our

# model won't support ragged sequences.

vectorize\_layer = TextVectorization(

 standardize=custom\_standardization,

 max\_tokens=max\_features,

 output\_mode="int",

 output\_sequence\_length=sequence\_length,

)

# Now that the vocab layer has been created, call `adapt` on a text-only

# dataset to create the vocabulary. You don't have to batch, but for very large

# datasets this means you're not keeping spare copies of the dataset in memory.

# Let's make a text-only dataset (no labels):

text\_ds = raw\_train\_ds.map(lambda x, y: x)

# Let's call `adapt`:

vectorize\_layer.adapt(text\_ds)

## Two options to vectorize the data

There are 2 ways we can use our text vectorization layer:

**Option 1: Make it part of the model**, so as to obtain a model that processes raw strings, like this:

text\_input = tf.keras.Input(shape=(1,), dtype=tf.string, name='text')

x = vectorize\_layer(text\_input)

x = layers.Embedding(max\_features + 1, embedding\_dim)(x)

...

**Option 2: Apply it to the text dataset** to obtain a dataset of word indices, then feed it into a model that expects integer sequences as inputs.

An important difference between the two is that option 2 enables you to do **asynchronous CPU processing and buffering** of your data when training on GPU. So if you're training the model on GPU, you probably want to go with this option to get the best performance. This is what we will do below.

If we were to export our model to production, we'd ship a model that accepts raw strings as input, like in the code snippet for option 1 above. This can be done after training. We do this in the last section.

def vectorize\_text(text, label):

 text = tf.expand\_dims(text, -1)

 return vectorize\_layer(text), label

# Vectorize the data.

train\_ds = raw\_train\_ds.map(vectorize\_text)

val\_ds = raw\_val\_ds.map(vectorize\_text)

test\_ds = raw\_test\_ds.map(vectorize\_text)

# Do async prefetching / buffering of the data for best performance on GPU.

train\_ds = train\_ds.cache().prefetch(buffer\_size=10)

val\_ds = val\_ds.cache().prefetch(buffer\_size=10)

test\_ds = test\_ds.cache().prefetch(buffer\_size=10)

## Build a model

We choose a simple 1D convnet starting with an Embedding layer.

from tensorflow.keras import layers

# A integer input for vocab indices.

inputs = tf.keras.Input(shape=(None,), dtype="int64")

# Next, we add a layer to map those vocab indices into a space of dimensionality

# 'embedding\_dim'.

x = layers.Embedding(max\_features, embedding\_dim)(inputs)

x = layers.Dropout(0.5)(x)

# Conv1D + global max pooling

x = layers.Conv1D(128, 7, padding="valid", activation="relu", strides=3)(x)

x = layers.Conv1D(128, 7, padding="valid", activation="relu", strides=3)(x)

x = layers.GlobalMaxPooling1D()(x)

# We add a vanilla hidden layer:

x = layers.Dense(128, activation="relu")(x)

x = layers.Dropout(0.5)(x)

# We project onto a single unit output layer, and squash it with a sigmoid:

predictions = layers.Dense(1, activation="sigmoid", name="predictions")(x)

model = tf.keras.Model(inputs, predictions)

# Compile the model with binary crossentropy loss and an adam optimizer.

model.compile(loss="binary\_crossentropy", optimizer="adam", metrics=["accuracy"])

## Train the model

epochs = 3

# Fit the model using the train and test datasets.

model.fit(train\_ds, validation\_data=val\_ds, epochs=epochs)

Epoch 1/3

625/625 [==============================] - 32s 51ms/step - loss: 0.6288 - accuracy: 0.5835 - val\_loss: 0.3283 - val\_accuracy: 0.8610

Epoch 2/3

625/625 [==============================] - 31s 50ms/step - loss: 0.2808 - accuracy: 0.8859 - val\_loss: 0.3005 - val\_accuracy: 0.8796

Epoch 3/3

625/625 [==============================] - 31s 50ms/step - loss: 0.1450 - accuracy: 0.9467 - val\_loss: 0.3795 - val\_accuracy: 0.8726

<tensorflow.python.keras.callbacks.History at 0x137444c90>

## Evaluate the model on the test set

model.evaluate(test\_ds)

782/782 [==============================] - 7s 9ms/step - loss: 0.3999 - accuracy: 0.8650

[0.39986345171928406, 0.8649600148200989]

## Make an end-to-end model

If you want to obtain a model capable of processing raw strings, you can simply create a new model (using the weights we just trained):

# A string input

inputs = tf.keras.Input(shape=(1,), dtype="string")

# Turn strings into vocab indices

indices = vectorize\_layer(inputs)

# Turn vocab indices into predictions

outputs = model(indices)

# Our end to end model

end\_to\_end\_model = tf.keras.Model(inputs, outputs)

end\_to\_end\_model.compile(

 loss="binary\_crossentropy", optimizer="adam", metrics=["accuracy"]

)

# Test it with `raw\_test\_ds`, which yields raw strings

end\_to\_end\_model.evaluate(raw\_test\_ds)

782/782 [==============================] - 11s 13ms/step - loss: 0.4062 - accuracy: 0.8630

[0.3998638987541199, 0.8649600148200989]