Detecting offensive tweets

Offenseval exercise

Task

SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval)

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, Ritesh Kumar

SemEval

Annual competitions on various semantic problems

1 task (often with subtasks)

1 training set (+ validation set)

1 test set

Deadline, restriction on resources which can be used

→ fair comparison

Task

Sub-task A: Offensive language identification (104 participating teams)

Sub-task B: Automatic categorization of offense types (71 participating teams)

Sub-task C: Offense target identification (66 participating teams)

Task

Sub-task A: Offensive language identification (104 participating teams)

Sub-task B: Automatic categorization of offense types (71 participating teams)

Sub-task C: Offense target identification (66 participating teams)

Task: detecting offensive tweets

Binary classification task:

Label tweets as

- Offensive (‘OFF’)
- Not offensive (‘NOT’)

Data

@USER He is a DUMBASS !!!!!	OFF
@USER @USER @USER @USER The Institution only let her tweet when she behaves. She is a Window Licker!!!!	OFF
@USER He always shows dedication at what he does no wonder he is the best URL	NOT
@USER Reagan also signed the first gun control bill as governor of CA	NOT

A critical look at the data

Offensive?

@USER I mean it worked for gun control right? URL

@USER Oh my Carmen. He is SO FRICKING CUTE

Not offensive?

@USER Please shut up Ontario is seeing what conservatives stand for @USER is not helping you loose this province your never going anywhere.. Not they you will any way your completely out of tune with Canadian Mr. Harper oops I mean Mr. Scheer common mistake you so much alike.

Data: training - development - test

- Training data (split into training and development)
- Trial data

Downloading the data

[insert link]

[share via flash drive?]

Where to start?

What is most informative?

Can we create a simple baseline?

Towards a machine learning system

- Which information do we want to represent and how?
- Which machine learning approach is most suitable?
- Can we beat the baseline?
- Can we tune the the machine learning approach further?

Focus on lexical features

$X \rightarrow y$

Tweet $\rightarrow$ OFF / NOT

Representation of X

- Bag of words (as opposed to exploiting structure)
- Sparse count encodings (vocabulary-sized count matrix)
- Word embeddings

Embeddings

Which type of embeddings works best?

→ Underlying corpus is important!

Suggestion: Plug in embeddings trained on tweets (Li et al. 2017)
Embeddings can be downloaded here and are compatible with Gensim

Code

Experiments set up by Sophie Arnoult + modifications to work with embeddings

What we have:

- Reading in the data
- Transforming raw data into vectors
- Simple classifiers: Naive Bayes, SVM
- Evaluation
- Grid search for parameter tuning

Code - from text to vectors

- Sparse count vectors with CountVectorizer (sklearn):
  - Documents → matrix of token counts
  - Sparse representation using scipy.sparse.csr_matrix
  - Default size: vocabulary size found in the data
- Embeddings:
  - Centroid of all words in a tweet (excluding OOV words)
  - If no tweets: vector of zeros (of the same length)

Running the code

cd pynlp

Change paths for input data in task/offenseval.py (or use command line arguments)

Run: python -m tasks.offenseval

Code - what we could add

- SVM with the suitable settings for dealing with the embedding representations: dense, 100-500 dimensions (as compared to sparse, vocabulary sized vectors)
- Other algorithms - neural net (tutorial for CNN)
- Experiment with different types of embeddings
- Error analysis - what goes wrong
  - Add code to write out predictions
- Analysis of most informative features

Code walk-through

Let’s beat the baseline!