Sentiment Analysis

By
Ethem Utku AKTAS
Ibrahim Faruk YALCINER

Problem Analysis

- Domain of any natural language problem is vast.
  - Turkish Language has ~104k words. (wikipedia)
  - Size of the set of all possible sentences is innumerable.
  - There are also many variations of a word in a sentence
    - Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizdenmişsinizcesine
      (Hemencecik başarısızlaştırıcı hâline getiremeyeceğimiz kişilerden biriymişsiniz gibi)
  - A tweet can have couple of sentences.
- Semantic and Syntactic Quality: Tweeter is not particularly famous for good language use.
  - “Acaba siz bu "Hasirt the Blackboard" mantiginda kesilen mangirlarla ilgilenir misiniz sayin büyük patron? @garanti @GarantiyeSor”

Problem Analysis

- Sarcasm: Even though everything may seem positive.
  - “@GarantiyeSor kart geldide 20 gün diyorsunuz ya hani 2 ayin sonunda geldi sükürler olsun.”
  - “@GarantiyeSor Garantide yeni uygulama: arayip müzik dinletiyorlar. Joy FM ile ortak mi oldular acaba?”
- Irregular usage
  - Foreign Languages: “Kılayntlarımızla (Client) olan rileyşınımızı (relation) bir çek (Check) edelim.”
  - Popular Made up Words and Abbreviations: Kanki, falan oldum, deermişimm sen de yeeermişinn

Problem Analysis

- English Character Usage: saç kör ölür → sac kor olur
- Unintentional tpyos
-

ML Solution

- Machine Learning is a great tool for this task, since the listed problems are really hard to formulate-model let alone the language itself.
- People have been working on these problems for a long time (NLP)
- ML proved to be very useful for such a task in recent years.
  - Google Search:

First Trial

- We have two set of data, 21 Features and Text
  - TF-IDF: term frequency–inverse document frequency used to generate features from Text with 2-grams.
  - 21 Features: Scaled with min-max approach
- Two learners: one for each set
- Cross Validation with 10 Fold
- Our first trials proved to be very poor with just the data, no matter which Learning Algorithm is chosen. But among them Support Vector Regressor gave best results.

First Trial

- But their average for each instance proved to give better results.
- First improvement: Ensemble Averaging
  - Text Learner (TL) MSE = 0.184
  - 21 Features Learner (21FL) MSE = 0.245
  - Average Ensemble with 0.3 * TL + 0.7 * 21FL = 0.161

What can we do?

- Can we add new features to 21
  - Most intuitive features are already included to 21.
  - NLP related task, which would take a lot of time.
- Can we remove any features from 21 for better generalization
  - With Feature selection with RFE module
  - Reduced 21 to most significant 5 w.r.t. training data
  - Proved to be not very useful for current case.

What can we do?

- Can we generate new data?
  - We can’t since we don’t know the exact criteria during labeling of existing training data.
- All though, we can approximate
  - Very burdensome task
  - Can we find new data:
    - SentiTurkNet: Most used words are labeled for their polarity. Originated from SentiWordNet.
    - Pretend they are single word tweets.

Relax the problem by simplifying input data

- Map Turkish characters to English alphabet
  - Many Turkish tweets are written with English alphabet
  - Find word roots, “güzel”, “güzeldi” , “güzeldiler” all have same polarity.
  - *Zemberek*: NLP tools for Turkish Language including Morphological Analysis, Tokenization, Language Identification etc. We have used it to get roots of the words.
  - Although have the problem of mapping opposite polarity words to same.
    - Olur: agreeing on something (+) → olur
    - Ölür: Dies (-) → olur

Relax the problem by simplifying input data

- Remove all digits and punctuations from text. They rarely provide polarity information (unless “I give you 10/10” or “give me a high 5”).
- Most used punctuations to represent emotional polarity are emojis, question mark (?), exclamation mark (!) , ellipsis (…). And they are already accounted in 21 Features.

Bagging

- Reduce Variance with Ensembling
- We have some what more robust data of SentiTurkNet and erroneous data training data.
  - Create bunch of learners each including SentiTurkNet + a portion of erroneous data.
  - Use Average Ensemble to merge results.

Results

- 21FL MSE = 0.245
- 20 SVR Learners with 50% of training data + SentiTurkNet data; $\min_i(SVR_i) = 0.165$
- Average with 0.3 coefficient to 21FL, rest is distributed to each SVR Learner evenly; MSE = 0.149