apple / **turicreate**

Branch: **master** ▾ **turicreate** / userguide / text_classifier / **introduction.md** Find file Copy path

znation Initial commit of Turi Create 4.0. e8e1f45 8 days ago

1 contributor

74 lines (57 sloc) 2.65 KB

Text classification

Text classification - commonly used in tasks such as sentiment analysis - refers to the use of natural language processing (NLP) techniques to extract subjective information such as the polarity of the text, e.g., whether or not the author is speaking positively or negatively about some topic.

In many cases, it can help keep a pulse on users' needs and adapt products and services accordingly. Many applications exist for this type of analysis:

- **Forum data**: Find out how people feel about various products and features.
- **Restaurant and movie reviews**: What are people raving about? What do people hate?
- **Social media**: What is the sentiment about a hashtag, e.g. for a company, politician, etc?
- **Call center transcripts**: Are callers praising or complaining about particular topics?

In addition, text classification can also be used to identify features (or aspects) of entities that are mentioned, and then estimate the sentiment for each aspect. For example, when studying reviews about mobile phones you may be interested in how people feel about aspects such as battery life, screen resolution, size, etc.

Introductory Example

Let's now take a look at a simple example of sentiment analysis where the task is to predict whether it contains positive or negative sentiment. For instance, the model will predict "positive sentiment" for a snippet of text -- whether it is a movie review or a tweet -- when it contains words like "awesome" and "fantastic". Likewise, having many words with a negative connotation will yield a prediction of "negative sentiment".

```
import turicreate as tc

# Load data
data = tc.SFrame('ratings_data.csv')

# Create a model
model = tc.sentence_classifier.create(data, 'rating', features=['text'])

# Make predictions & evaluation the model
predictions = model.predict(data)
results = model.evaluate(data)
```

How it works

Training a text classifier is really important when you want to tune the model to your data set to take advantage of vocabulary that is particular to your application. The text classifier in Turi Create is currently a simple combination of two components:

- feature engineering: a bag-of-words transformation
- statistical model: a LogisticClassifier is used to classify text based on the above features

The bag-of-words and a logistic regression classifier is a very strong baseline for this particular task and works on a wide variety of datasets.

After creating the above model, you can inspect the underlying classifier model using:

```
classifier = model.classifier
```