Oleksandr Romanko, Ph.D.
Senior Research Analyst, Risk Analytics, Watson Financial Services, IBM Canada
Adjunct Professor, University of Toronto

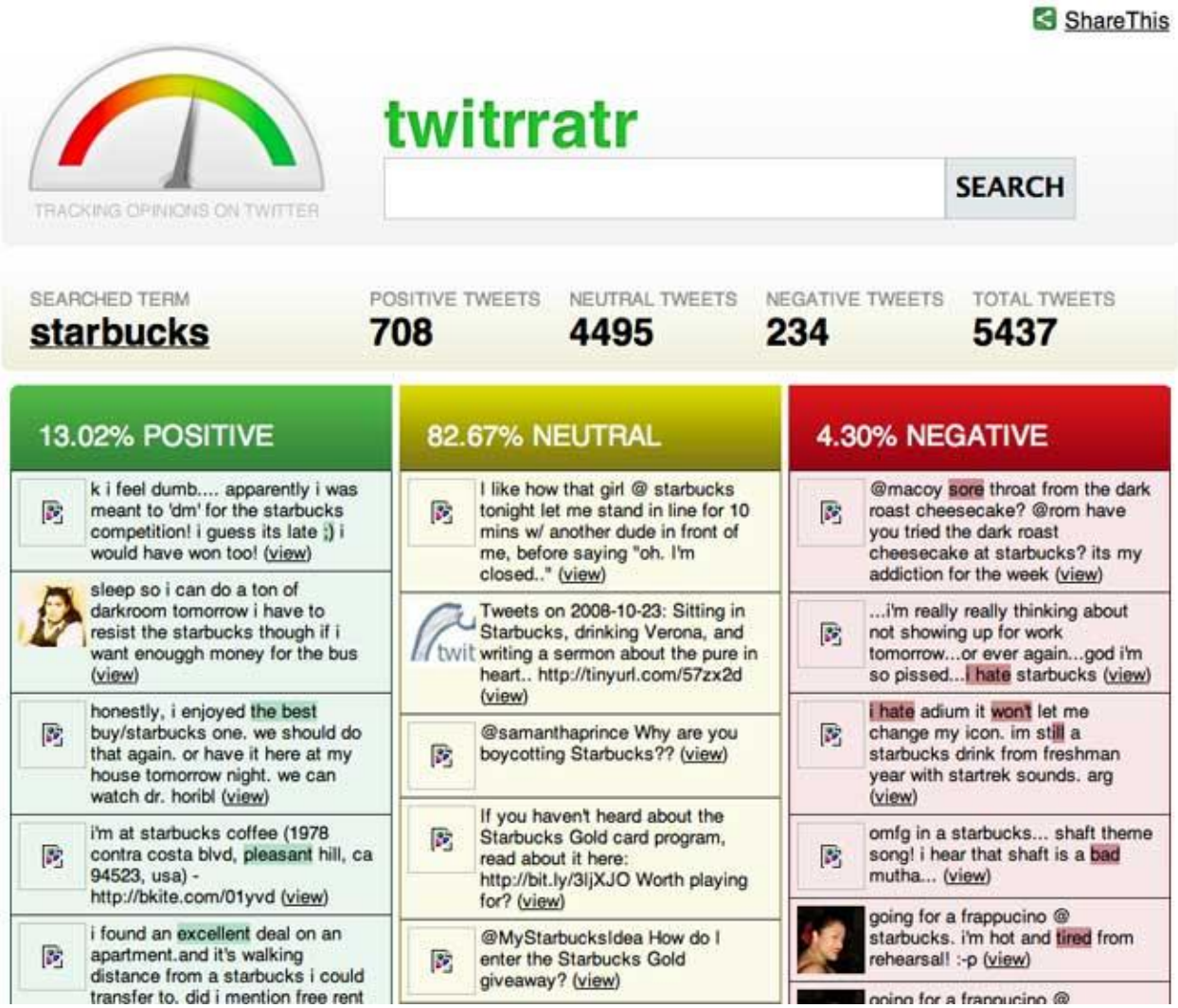
Yuwei Feng
Teaching Assistant, University of Toronto

MIE1624H – Introduction to Data Science and Analytics

Assignment 2 – Natural Language Processing of Tweets

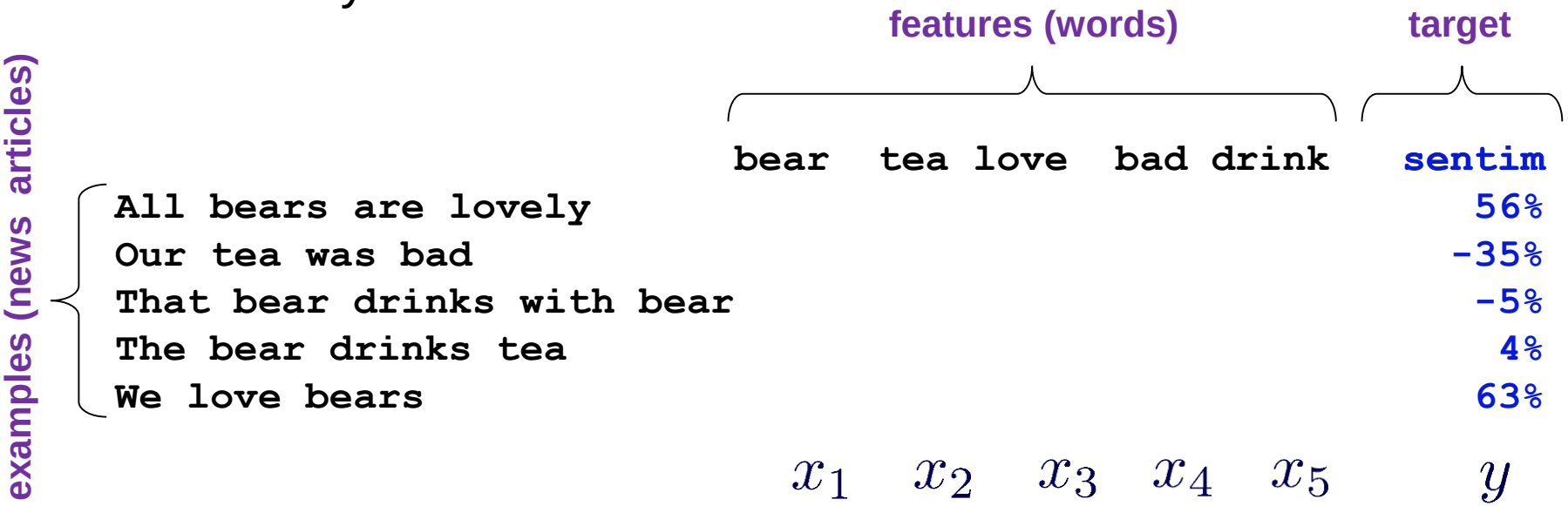
University of Toronto
November 5, 2019

Text analytics and sentiment analysis



Sentiment analysis of tweets

Natural Language Processing: features and target variable in sentiment analysis



Stop words that were removed:

- all
- are, was
- our
- that
- with
- the

Natural Language Processing: ‘bag of words’ based on Word Frequency (WF) and sentiment analysis

examples (news articles)

features (WF) | target

	bear	tea	love	bad	drink	sentim
All bears are lovely	1	0	1	0	0	56%
Our tea was bad	0	1	0	1	0	-35%
That bear drinks with bear	2	0	0	0	1	-5%
The bear drinks tea	1	1	0	0	1	4%
We love bears	1	0	1	0	0	63%

bag of words

x_1 x_2 x_3 x_4 x_5 y

Supervised machine learning algorithm:

- ❑ Linear regression
- ❑ Decision trees
- ❑ SVM regression
- ❑ k-NN regression
- ❑ Ensembles (random forests, XGBoost)
- ❑ Artificial neural nets (deep learning)

$$y = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(x_1, \ldots, x_5)$$

$$y = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \ldots + \theta_5 \cdot x_5 + \epsilon$$

Natural Language Processing: Term Frequency (TF) and sentiment analysis

$$\text{TF} = \frac{\text{Word Frequency}}{\text{number of words in document}}$$

examples (news articles)

features (TF) | target

	bear	tea	love	bad	drink	sentim
All bears are lovely	1/2	0	1/2	0	0	56%
Our tea was bad	0	1/2	0	1/2	0	-35%
That bear drinks with bear	2/3	0	0	0	1/3	-5%
The bear drinks tea	1/3	1/3	0	0	1/3	4%
We love bears	1/2	0	1/2	0	0	63%
	x_1	x_2	x_3	x_4	x_5	y

Supervised machine learning algorithm:

- ❑ Linear regression
- ❑ Decision trees
- ❑ SVM regression
- ❑ k-NN regression
- ❑ Ensembles (random forests, XGBoost)
- ❑ Artificial neural nets (deep learning)

$$y = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(x_1, \ldots, x_5)$$

$$y = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \ldots + \theta_5 \cdot x_5 + \epsilon$$

Natural Language Processing: TF-IDF and sentiment analysis

TF-IDF is short for “term frequency–inverse document frequency”

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

where **tf** is “term frequency” and **idf** is “inverse document frequency”.

$$\text{tf}(t) = \frac{\text{word } t \text{ frequency}}{\text{number of words in document}}$$

$$\text{idf}(t) = \log \frac{\text{total number of documents}}{\text{number of documents with term } t \text{ in it}} = \log \frac{N}{|\{d \in D : t \in d\}|}$$

with

- N: total number of documents in the corpus $N = |D|$
- $|\{d \in D : t \in d\}|$: number of documents where the term t appears (i.e., $\text{tf}(t, d) \neq 0$).

There are a number of variants for **tf** and **idf** weights.
For the example in next page, we use word frequency as term frequency and the following equation to calculate **idf**. This is also the default of **TfidfVectorizer** in Python.

$$\text{idf}(d, t) = \log \left[\frac{1 + N}{1 + \text{df}(d, t)} \right] + 1$$

Natural Language Processing: TF-IDF and sentiment analysis

examples (news articles)

features (TF-IDF) | target

	bear	tea	love	bad	drink	sentim
All bears are lovely	1.2	0	1.7	0	0	56%
Our tea was bad	0	1.7	0	2.1	0	-35%
That bear drinks with bear	2.4	0	0	0	1.7	-5%
The bear drinks tea	1.2	1.7	0	0	1.7	4%
We love bears	1.2	0	1.7	0	0	63%
	x_1	x_2	x_3	x_4	x_5	y

Supervised machine learning algorithm:

- Linear regression
- Decision trees
- SVM regression
- k-NN regression
- Ensembles (random forests, XGBoost)
- Artificial neural nets (deep learning)

$$y = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(x_1, \ldots, x_5)$$

$$y = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \ldots + \theta_5 \cdot x_5 + \epsilon$$

Natural Language Processing: word embeddings and sentiment analysis

examples (news articles)

	features vector (word embedding)	target
		sentim
All bears are lovely	[2.31 1.09 -1.7 … 1.08]	56%
Our tea was bad	[-3.2 1.72 1.56 …-1.78]	-35%
That bear drinks with bear	[2.66 4.09 -1.04… 2.92]	-5%
The bear drinks tea	[-1.82 8.88 1.45… 2.75]	4%
We love bears	[2.31 1.1 -1.71 … 1.08]	63%
	x_1 x_2 ... x_n	y

Supervised machine learning algorithm:

- ❑ Linear regression
- ❑ Decision trees
- ❑ SVM regression
- ❑ k-NN regression
- ❑ Ensembles (random forests, XGBoost)
- ❑ Artificial neural nets (deep learning)

$$y = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(x_1, \ldots, x_n)$$

More reading material about word embedding: https://towardsdatascience.com/word-embeddings-exploration-explanation-and-exploitation-with-code-in-python-5dac99d5d795

Natural Language Processing: word frequency (Word Cloud)