TEXT CLASSIFICATION FOR REAL-WORLD PROBLEMS

Diky Hadna | Software Engineer | PT Digital Bangsa – Inkuiri.com

TEXT CLASSIFICATION

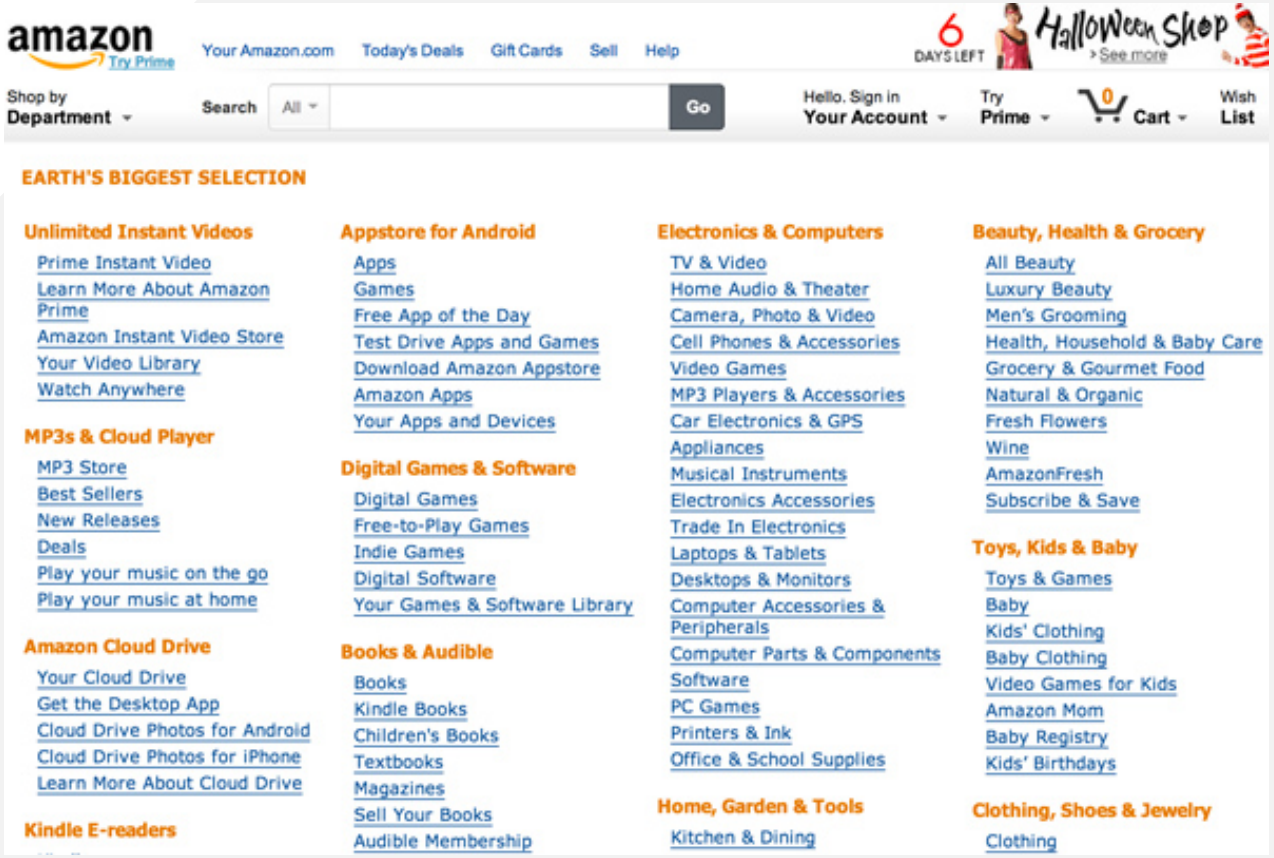
- Task of assigning predefined categories to free-text documents.
- Part of NLP (Natural Language Processing), part of ML (Machine Learning).
- Aliases: *Text categorization, topic categorization, topic classification & topic spotting.*

TEXT CLASSIFICATION USAGE

- Content/product tagging: *e-commerce, news, online directory*
- Sentiment analysis: *positive or negative*
- Spam detection: *email, web comment section*
- Chatbot: *intent detection*

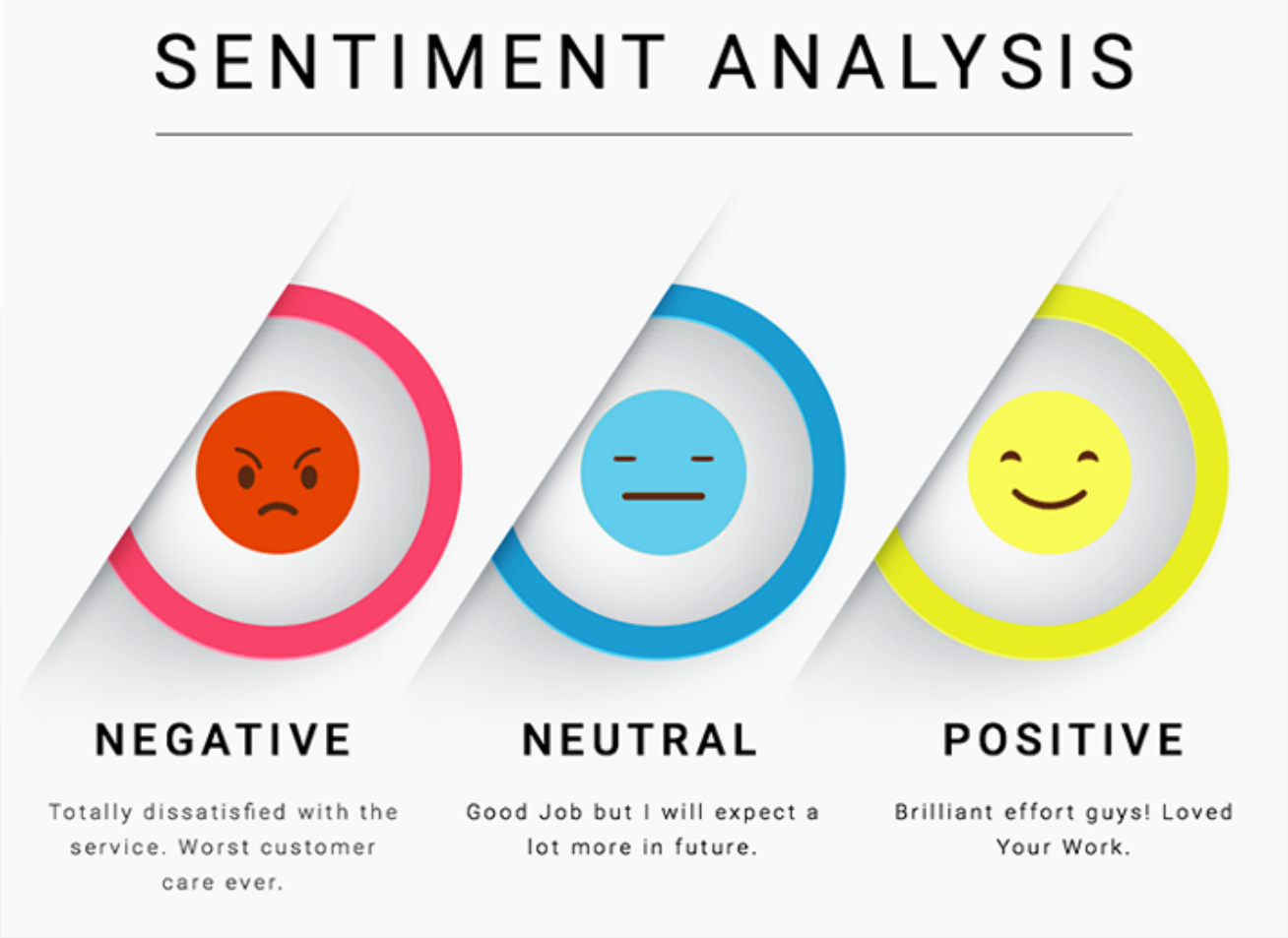
CONTENT/PRODUCT TAGGING

- Commonly used in content-focused site (e-commerce, news, online directory)
- Users usually put the content, and system can suggest the labels/ categories



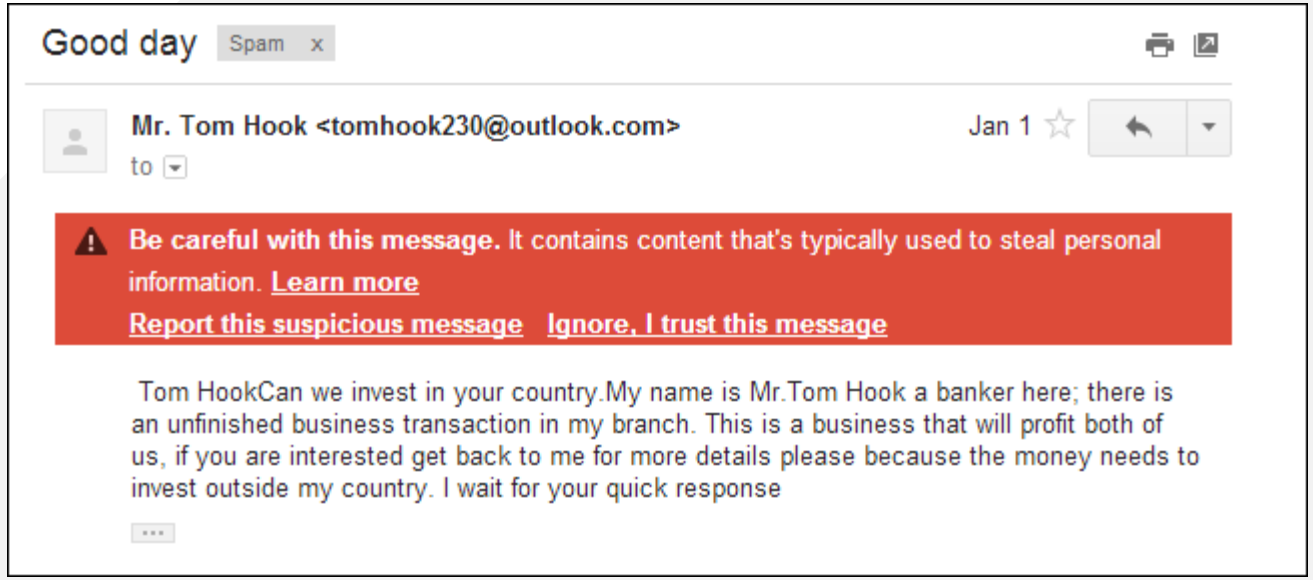
SENTIMENT ANALYSIS

- A process to identify or categorizing opinion sentiment polarity
- Usually categorized to positive and negative



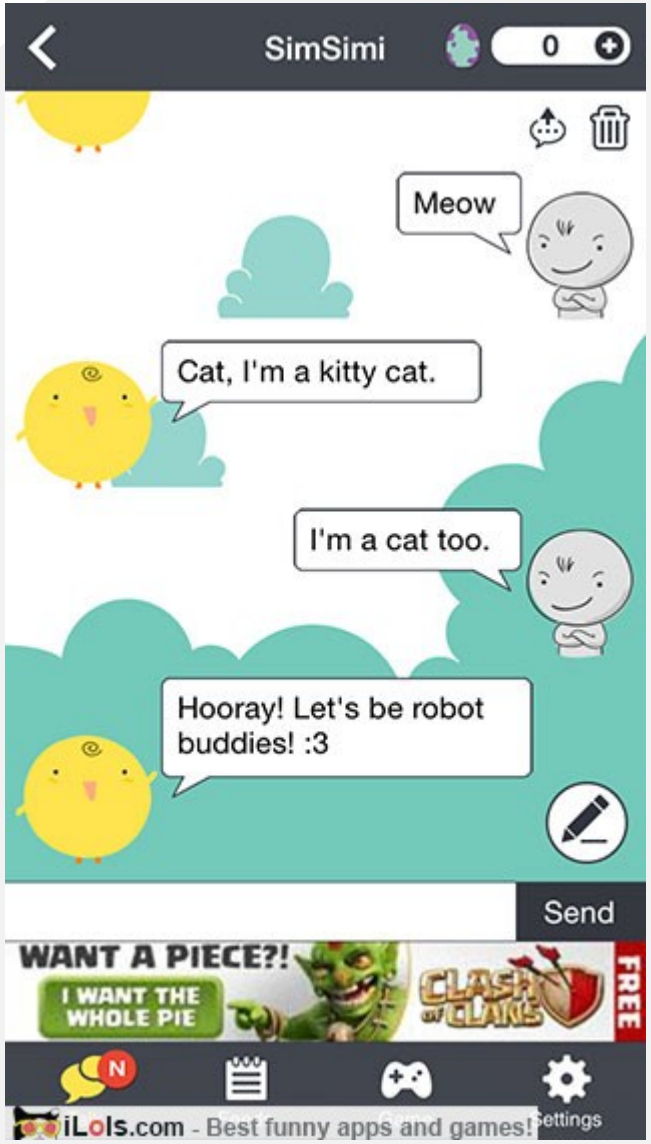
SPAM DETECTION

- Mostly used in email service
- Can also be used in blog/website comment section



CHATBOT

- Text classification is being used to classify the users' *intents*.
- Based on the *intents*, chatbot will send the response.



TEXT CLASSIFICATION PROCESSES

- Cleaning: *punctuation and unnecessary character removal*
- Filtering: *stop words removal*
- Tokenizing: *breaking any given text into pieces (called tokens)*
- Stemming & Lemmatization: *chopping the end of words and/or return it to original form*
- Weighting: *scoring the frequency of a token*
- Training: *learning process using machine learning algorithms*

CLEANING: PUNCTUATION REMOVAL

A process to remove all the unnecessary punctuations & characters.

Input: `“Hi, Marry! How are you today? You look so happy!”`

Output: `“Hi Marry How are you today You look so happy”`

FILTERING: STOP WORDS REMOVAL

Stop words refer to list of most common words which have less to no meaning.

E.g.: `a, am, the, or, in, is`

Input: `“Hi, Marry! How are you today? You look so happy!”`

Output: `“Hi, Marry! Today? Look happy!”`

TOKENIZING

A process to chop any given text into smaller pieces.

Input: “Hi, Marry! How are you today? You look so happy!”

Output: “Hi” “Marry” “How” “are” “you” “today” “you” . . .

STEMMING

Stemming: chopping the ends of the words

Input: “car” “cars” “car’s” “cars’”

Output: “car”

LEMMATIZATION

Lemmatization: return the words to its original form

Input: “The boy’s cars are different colors”

Output: “The boy car be differ color”

WEIGHTING

- Giving score to reflect how important a word is to a document in a collection of corpus.
- Common method: TF-IDF (Term Frequency – Inverse Document Frequency)

TF(t) = (Number of times term *t* appears in a document) / (Total number of terms in the document)

IDF(t) = log_e(Total number of documents / Number of documents with term *t* in it)

TRAINING

Process to train our model with the supplied dataset.

Popular algorithms:

- Naive Bayes classifier
- Support Vector Machine
- K-Nearest Neighbor

DEMO

Github: @dkhd

https://github.com/dkhd/text-classification

THANK YOU!


Diky Hadna | https://fb.me/dkyhd


dikyhadna@gmail.com


@dkyhd | https://t.me/dkyhd


@dkhd | https://medium.com/@dkhd