A Course on Social Engineering
Phishing & Email Analysis

Automated NLP-based Email Analysis

Akbar Namin
Texas Tech University

Spring 2021

Phishing Emails

- The detection of phishing is a hard problem
  - It is very easy to create and setup a replica of good and legitimate Websties
  - Phishing Websites and emails are hard to be differentiated from legitimate and genuine ones
- A possible approach
  - Extracting information and features from emails and attack vectors
  - Hybrid: Extracting data found in emails and collecting information from external sources (URL Analysis)

Spam Detection Techniques

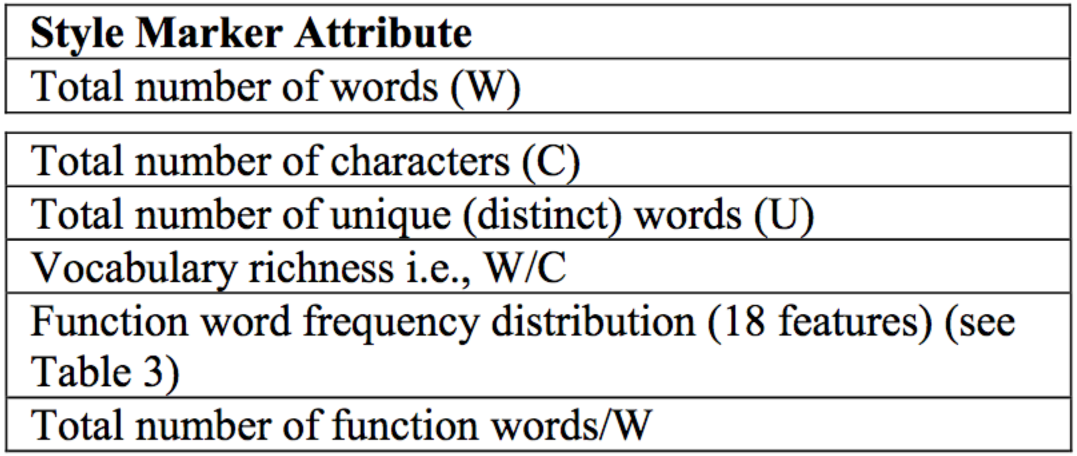
- The detection of phishing is a hard problem
  - An effective system of review spam detection should satisfy the following orthogonal requirements [12]:
    1. The system should be accurate enough to reduce the chances of misclassifying ham as spam or vice versa.
    2. The system should be efficient enough to process the large volumes of reviews currently being posted to the Internet.
    3. The system should be equipped with a component capable of handling spammers’ obfuscation strategies, which make fake reviews look like the legitimate ones.
    4. The system should be able to detect different types of fake reviews. Finally, the sys-tem should be able to manage the uncertainty involved in review spam detection by providing probabilistic estimations of the detection results

Phishing Emails – Structure of Phishing Emails

- The common structure of phishing emails [6]:
  - Spoofing of banks/retailers' Websites
    - Gaining the trust of the users through resembling legitimate Websites
  - Different link in the text than actual legitimate destination
    - Redirecting www.bankofamerica.com (as a URL) to www.msweethome.ch/sign-in/
  - Usage of general terms/words in addressing recipients
    - General phishing emails and contents targeting more victims
  - Usage of well-defined/interesting situational contexts to attract users
    - The use of persuasion techniques (i.e., urgency, threat, etc.) to deceit users

Phishing Emails – Email Filtering

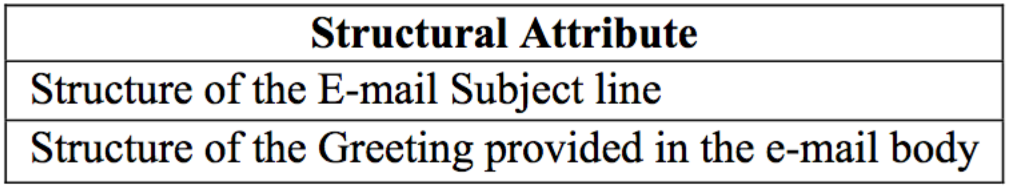
- Approaches [6]
  - Use linguistic and structural features of emails to decide about phishing emails
    - Total number of characters, Total number of unique (distinct) words, Vocabulary richness, etc.
  - E.g., Number of words, the richness of the vocabulary, the structure of the subject line, and the presence of 18 keywords



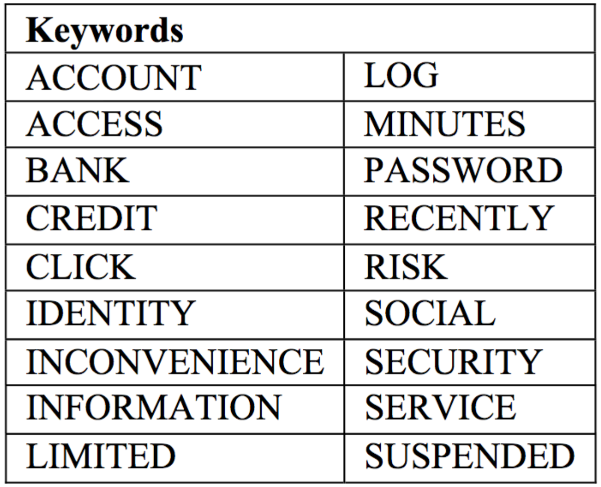
Style Marker Attribute
Total number of words (W)
Total number of characters (C)
Total number of unique (distinct) words (U)
Vocabulary richness i.e., W/C
Function word frequency distribution (18 features) (see Table 3)
Total number of function words/W

Phishing Emails – Email Filtering

- Approaches [6]
  - Use linguistic and structural features of emails to decide about phishing emails
    - Total number of characters, Total number of unique (distinct) words, Vocabulary richness, etc.
  - E.g., Number of words, the richness of the vocabulary, the structure of the subject line, and the presence of 18 keywords



Structural Attribute
Structure of the E-mail Subject line
Structure of the Greeting provided in the e-mail body



Keywords	
ACCOUNT	LOG
ACCESS	MINUTES
BANK	PASSWORD
CREDIT	RECENTLY
CLICK	RISK
IDENTITY	SOCIAL
INCONVENIENCE	SECURITY
INFORMATION	SERVICE
LIMITED	SUSPENDED

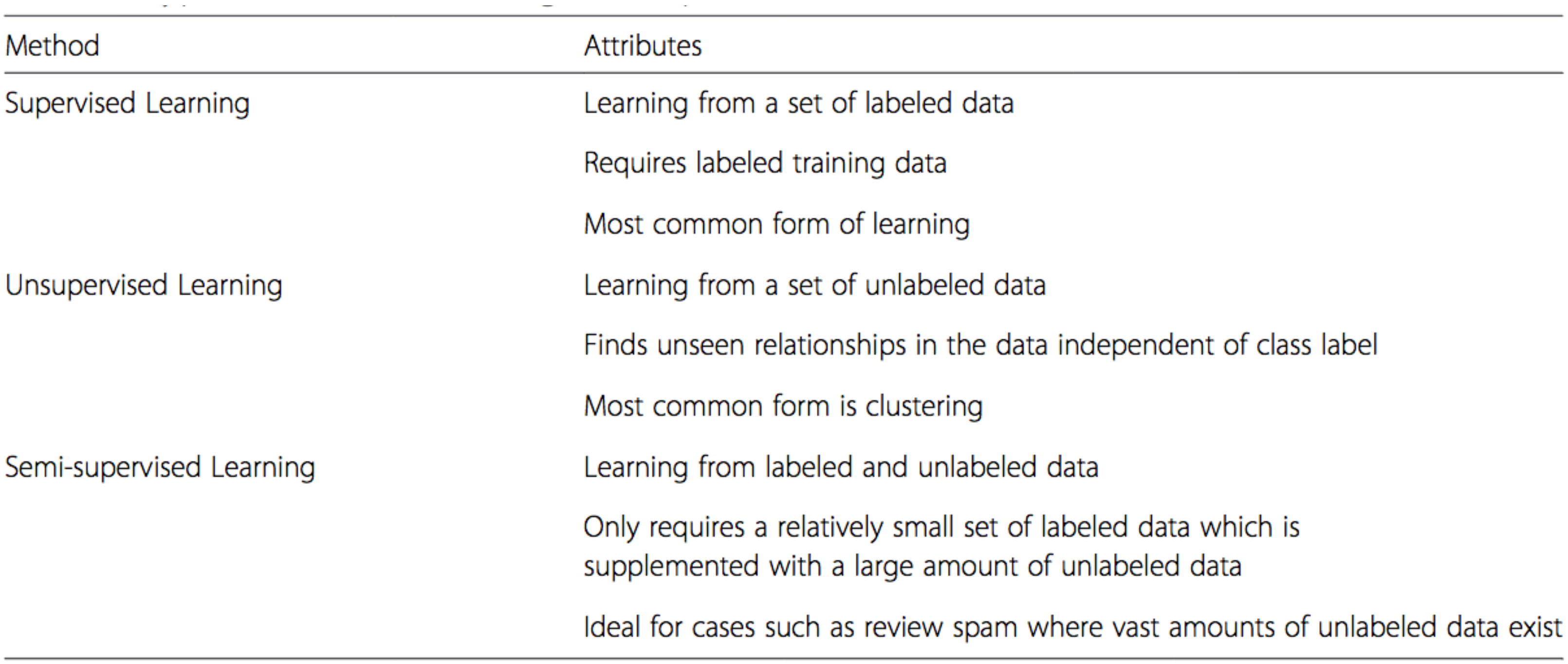
Phishing Emails – Automation

- Feature vector [6]:
  - The combination of textual information found in emails and the output of URL analysis
- A trained model based on the feature vector can decide about the legitimacy of emails.
  - Basically a machine learning classify
    - Support Vector Machine
    - Logistic Regression
    - Naïve based
    - Etc.

$$\text{Accuracy} = \frac{\text{Total number of e-mails classified correctly}}{\text{Total number of samples to classify}}$$

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Types of machine learning techniques
    - Supervised learning refers to the task of learning from labeled data
    - unsupervised learning (e.g., clustering) uses unlabeled data to find unseen relation-ships between instances independent of a class attribute.
    - Semi-supervised learning is a combination of the two and uses a few labeled instances in combination with a large number of unlabeled instances to train a classifier and has shown promise in the area of review spam detection



Method	Attributes
Supervised Learning	Learning from a set of labeled data
	Requires labeled training data
	Most common form of learning
Unsupervised Learning	Learning from a set of unlabeled data
	Finds unseen relationships in the data independent of class label
	Most common form is clustering
Semi-supervised Learning	Learning from labeled and unlabeled data
	Only requires a relatively small set of labeled data which is supplemented with a large amount of unlabeled data
	Ideal for cases such as review spam where vast amounts of unlabeled data exist

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - A classification problem of separating reviews into two classes: spam and non-spam
      - Opinion mining: extracting or summarizing the opinions from text by using NLP
      - Three types of spam reviews:
        - » Type 1 (untruthful opinions, fake reviews, bogus reviews): Those that deliberately mislead readers or opinion mining systems by giving undeserving positive reviews to some target objects in order to promote the objects (i.e., hyper spam) and/or by giving unjust or malicious negative reviews to some other objects in order to damage their reputation (i.e., defaming spam).
        - » Type 2 (reviews on brands only): Those that do not comment on the products in reviews specifically for the products but only the brands, the manufacturers or the sellers of the products.
        - » Type 3 (non-reviews): Those that are non-reviews, which have two main sub-types: (1) advertisements and (2) other irrelevant reviews containing no opinions (e.g., questions, answers, and random texts).

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Finding near duplicate reviews: defined as reviews with a Jaccard similarity score of >90%
      - Using w-shingling
        - » https://community.datarobot.com/t5/resources/w-shingling-language-processing-and-text-mining/ta-p/683
        - » can be used for classification, clusterization, and other problems
        1. we have to define test and train sets
        2. split both the test/train sets into bigger sets of sequential substrings of a fixed length
        3. it creates a substring beginning with each word in the string
        4. Next, we define the distance between set A and each element of B.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Using w-shingling
      - Example:
        - The test document:
        - *“the brown dog and the white horse are friends*”
        - The train document:
        - *the brown cow and the white horse are in the field eating the green grass during the day in the summer*
        - *the yellow and the white flowers close to the brown dog*
        - *brown dogs are my favorite ones*

        - A= [('the', 'brown', 'dog'), ('brown', 'dog', 'and'), ('dog', 'and', 'the'), ('and', 'the', 'white'), ('the', 'white', 'horse'), ('white', 'horse', 'are'), ('horse', 'are', 'friends')]
        - B[1]= [('the', 'yellow', 'and'), ('yellow', 'and', 'the'), ('and', 'the', 'white'), ('the', 'white', 'flowers'), ('white', 'flowers', 'close'), ('flowers', 'close', 'to'), ('close', 'to', 'the'), ('to', 'the', 'brown'), ('the', 'brown', 'dog')]
B[2]= [('brown', 'dogs', 'are'), ('dogs', 'are', 'my'), ('are', 'my', 'favorite'), ('my', 'favorite', 'ones')]
B[3]= ('and', 'the', 'white')

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Using w-shingling
      - Example:
        - » Similarity:
        - » r(A,B_i) = \frac{N(A and B_i)}{N(A or B_i)}
        - » [0.13043478 **0.14285714** 0.]

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Using Semantic Language Models (SLM)
        - » https://cmusphinx.github.io/wiki/semanticlanguagemodel/
      - language model will be extended with semantic information so as to improve the speech recognition performance
      - Integrating word relationships into language models (It introduces how word relationships and co-occurrence be integrated in language model)
      - Latent Semantic Analysis (LSA)

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Implementing Semantic Language Models (SLM)
        - » Method 1: WordNet
        - » Using WordNet to get a probability estimator.
        - » we want to get P(w_i|w), where w_i and w are assumed to have a relationship in WordNet.
        - » The formular is: P(w_i|w) = \frac{c(w_i,w|W,L)}{\sum_{w_j} c(w_j,w|W,L)}.
        - » Where, W is a window size, c(w_i,w|W,L) is the count that w_i and w appearing together within W-window.
        - » It can be got just by counting in certain corpus.
        - » What relationships can be considered:
        1. Synonym
        2. Hypernym
        3. Hyponym
        4. hierarchical distance between words

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Implementing Semantic Language Models (SLM)
        - Method 2: Co-Occuring
        - Using the co-occurrence is the same as WordNet.
        - we only need to replace c(w_i,w|W,L) with c(w_i,w|W), which means that w_i and w don’t need to have a link in WordNet.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Implementing Semantic Language Models (SLM)
        - » Method 3: LSA
        - » The high level idea of LSA is to convert words into concept representations
        - » it assumes that if the occurrence pattern of words in documents is similar then the words are similar.

        1. To build the LAS model, a co-occurrence matrix W will be built first, where w_{ij} is a weighted count of word w_j and document d_j. w_{ij} = G_i L_{ij} C_{ij} where, C_{ij} is the count of w_i in document d_j; L_{ij} is local weight; G_i is global weight. Usually, L_{ij} and G_i can use TF/IDF.
        2. Then, SVD Analysis will be applied to W, then W = U S V^T where, W is a M*N matrix, (M is the Vocabulary size, N is document size); U is M*R, S is R*R and V is a R*N matrix. R is usually a predefined dimension number between 100 and 500.
        3. After that, each word w_i can be denoted as a new vector U_i = u_i*S
        4. Based on this new vector, a distance between two words is defined: K(U_i, U_j) = \frac{u_i*S^2*u_m^T}{|u_i*S||u_m*S|}

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Implementing Semantic Language Models (SLM)
        - » Method 3: LSA

5. Therefore, we can perform a clustering to words into K clusters, C_1, C_2, …., C_K.
6. Let H_{q-1} be the history for word W_q, then we can get the probability of W_q given H_{q-1} by formula below: P(W_q|H_{q-1}) = P(W_q|W_{q-1},W_{q-2},… W_{q-n+1}, d_{q_1}) = P(W_q|W_{q-1},W_{q-2},…W_{q-n+1})*P(W_q| d_{q_1}|) where, P(W_q|W_{q-1},W_{q-2},…W_{q-n+1}) is n-gram model; P(d_{q_1}|W_q) is the LSA model.

and, P(W_q|d_{q_1}) = P(U_q|V_q) = K(U_q, V_{q_1})/Z(U,V) K(U_q, V_{q_1}) = \frac{U_q*S*V_{q-1}^T}{|U_q*S^{1/2}*||V_{q-1}*S^{1/2}|}

Z(U,V) is normalized factor.

5. We can apply word smoothing to the model based K-Clustering as follows: P(W_q| d_{q_1}) = \sum_{k=1}^{K} P(W_q|C_k)P(C_k|d_{q_1}) where, P(W_q|C_k), P(C_k|d_{q_1}) can be computer use the distance measurement given above by a normalized factor.
6. In this way, N-gram and LSA model are combined into one language model.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Using a logistic regression model and when tested using 10-fold cross validation, an Area Under the receiver operating characteristic Curve (AUC)
        - Finding duplicate reviews is a trivial task

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Feature identification and construction
      - There are three main types of information related to a review:
        1. the content of the review,
        2. the reviewer who wrote the review, and
        3. the product being reviewed.
      - Three types of features:
        1. review centric features: characteristics of reviews
        2. reviewer centric features: characteristics of reviewers
        3. product centric features: information about the product
      - For some features, we divide products and reviews into three types based on their average ratings (rating scale: 1 to 5): Good (rating ≥ 4), bad (rating ≤ 2.5) and Average, otherwise

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Feature identification and construction
      1. review centric features: characteristics of reviews
         - » Number of feedbacks (F1),
         - » number of helpful feedbacks (F2)
         - » percent of helpful feedbacks (F3) that the review gets
         - » Length of the review title (F4)
         - » length of review body (F5). longer reviews tend to get more helpful feedbacks and customer's attention.
         - » Position of the review in the reviews of a product sorted by date, in both ascending (F6) descending (F7) order.
         - » We also use binary feature to indicate if a review is the first review (F8) or the only review (F9).
         - » Percent of positive (F10) and negative (F11) opinion-bearing words in the review, e.g., "beautiful", "great", "bad" and "poor". Many researchers have compiled such lists for opinion or sentiment classification.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Feature identification and construction
      1. review centric features: characteristics of reviews
         - » Cosine similarity (F12) of the review and product features (which are obtained from the product description page at the amazon.com site). This feature is useful for detecting type 3 reviews, particularly advertisements.
         - » Percent of times brand name (F13) is mentioned in the review. This feature was used for reviews which praise or criticize the brand.
         - » Percent of numerals (F14), capitals (F15) and all capital (F16) words in the review. These features are useful for detecting non-reviews. Excessive use of numerals signifies too much technical detail. Capitals and all capitals signify poorly written and unrelated reviews.
         - » Rating (F17) of the review and its deviation (F18) from product rating. Feature indicating if the review is good, average or bad (F19). b.Binary features indicating whether a bad review was written just after the first good review of the product and vice versa (F20, F21). A spammer might have written such reviews to do damage control.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Feature identification and construction

        1. review centric features: characteristics of reviews
        2. reviewer centric features: characteristics of reviewers
           - » Ratio of the number of reviews that the reviewer wrote which were the first reviews (F22) of the products to the total number of reviews that he/she wrote, and ratio of the number of cases in which he/she was the only reviewer (F23).
           - » Rating related features: average rating given by reviewer (F24), standard deviation in rating (F25) and a feature indicating if the reviewer always gave only good, average or bad rating (F26). The first two features are obvious. The third is for reviewers who write only type 3 spam, they tend to give the same rating to all products they review to save time.
           - » Binary features indicating whether the reviewer gave more than one type of rating, i.e. good, average and bad. There are four cases: a reviewer gave both good and bad ratings (F27), good rating and average rating (F28), bad rating and average rating (F29) and all three ratings (F30). These four features are for the cases where a reviewer praises products of some brand, but criticizes the products of a competitor brand.
           - » Percent of times that a reviewer wrote a review with binary features F20 (F31) and F21 (F32).
        3. product centric features: information about the product

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Opinion mining [7]
      - Feature identification and construction
        1. review centric features: characteristics of reviews
        2. reviewer centric features: characteristics of reviewers
        3. product centric features: information about the product
           - » Price (F33) of the product.
           - » Sales rank (F34) of the product. Amazon assigns sales rank to “now selling products”, which is updated every hour. The sales rank is calculated based on some combination of recent and historic sales of the product. These features were helpful since spams could be concentrated on cheap/expensive or less selling products.
           - » Average rating (F35) and standard deviation in ratings (F36) of the reviews on the product.

Detecting Spams/Fake Reviews/Phishing Emails

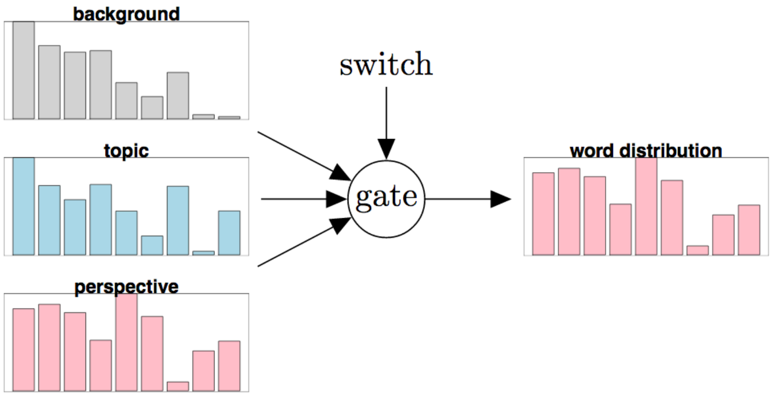
- Review centric spam detection [1]
  - Supervised learning
    - Generality of deception detection (cross-domain deceptive data) [8]
      - Existing supervised learning algorithms are usually narrowed to one specific domain
      - They rely heavily on domain-specific vocabulary.
      - To address this, create a cross domain dataset that included three types of reviews from three domains (e.g., hotel, restaurant and doctor)
      - The classification framework is based on using the Sparse Additive Generative Model (SAGE), a generative Bayesian approach [9]
        - » SAGE (Sparse Additive Generative Model): A generative Bayesian approach which can be viewed as an combination of topic models and generalized additive models. Unlike other derivatives of topic models, SAGE drops the Dirichlet-multinomial assumption and adopts a Laplacian prior, triggering sparsity in topic-word distribution.
        - » SAGE constructs multi-faceted latent variable models by simply adding together the component vectors rather than incorporating multiple switching latent variables in multiple facets.

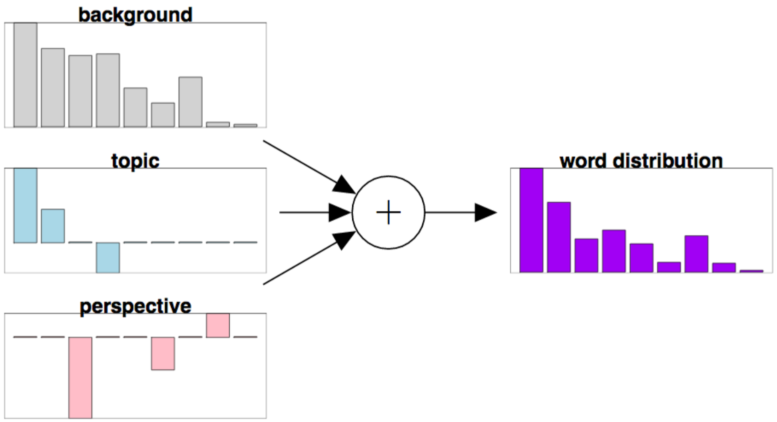
Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Generality of deception detection (cross-domain deceptive data) [8]
      - A Bayesian generative approach that can capture the multiple generative facets (i.e., deceptive vs. truthful, positive vs. negative, experienced vs. non-experienced, hotel vs. restaurant vs. doctor)
      - A combination of topic models (statistical models for discovering abstract topics in a collection of documents) and generalized additive models (linear models in which the linear predictor is dependent on unknown, smoother functions) generated using SAGE as well as SVM in their classification experiments.
      - Findings:
        - » More general features, such as LIWC and POS, to be more robust than unigram features alone when modeled using SAGE for cross-domain classification;
        - » however, when comparing the intra-domain classification (i.e., hotels reviews only) the best performance is achieved by unigram features.
        - » different linguistic features may appear in different domains, and more robust cues of deceptive opinion spam need to be identified if across domain classifier is to be created.

Phishing Emails

- Multinomial Switching model SAGE





Advantages of SAGE [8]:

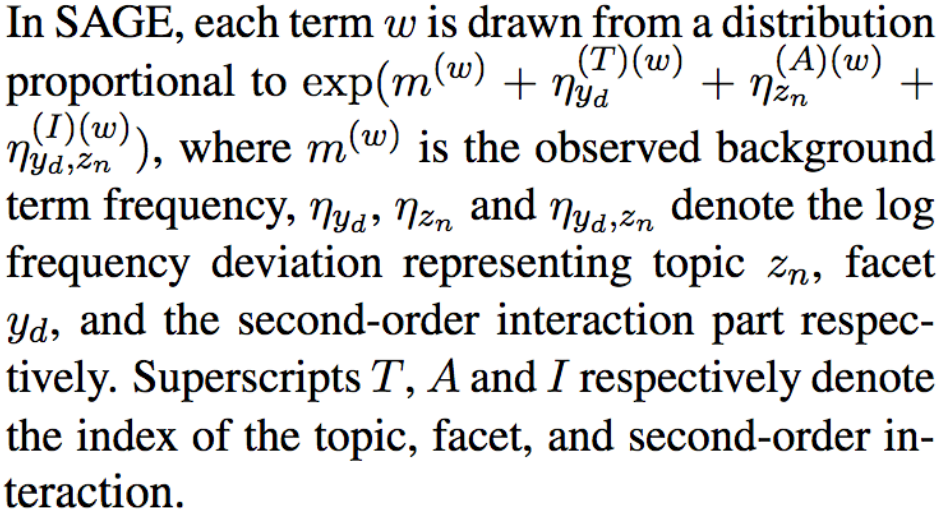
- The "additive" nature of SAGE allows a better understanding of which features contribute most to each type of deceptive review and how much each such feature contributes to the final decision jointly. If we instead use SVM, for example, we would have to train classifiers one by one (due to the distinct features from different sources) to draw conclusions regarding the differences between Turker vs Expert vs truthful reviews, positive expert vs negative expert reviews, or reviews from different domains. This would not only become intractable, but would make the conclusions less clear.
- For cross-domain classification task, standard machine learning approaches may suffer due to domain-specific properties

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
    - Supervised learning
        - Generality of deception detection (cross-domain deceptive data) [8]
            - Data sets for additive models (Domain):
                - Hotel Reviews
                - Doctor Reviews
                - Restaurant Reviews
            - Fake reviews provided by (Source)
                - Turker,
                - Employee,
                - Customers
            - Sentiment (Sentiment)
                - Positive
                - Negative

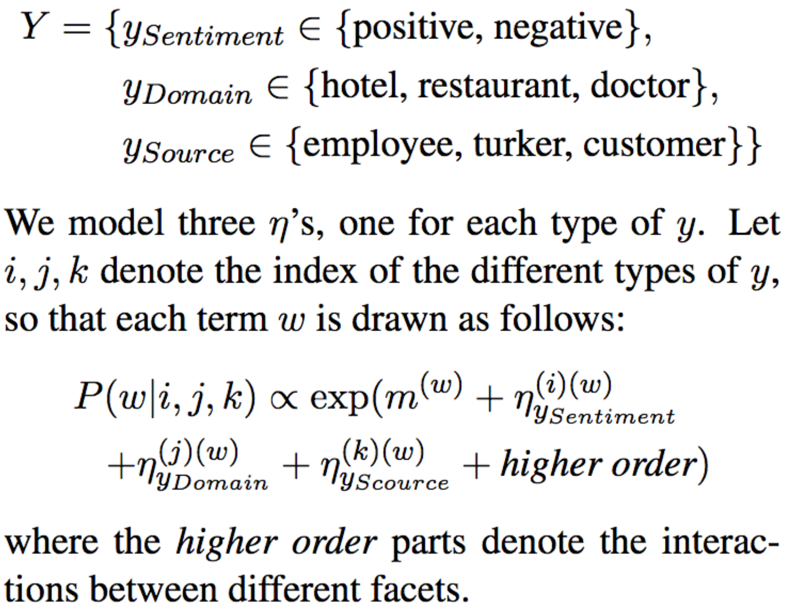
Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Generality of deception detection (cross-domain deceptive data) [8]
      - The mathematical model:



In SAGE, each term w is drawn from a distribution proportional to $\exp(m^{(w)} + \eta_{y_d}^{(T)(w)} + \eta_{z_n}^{(A)(w)} + \eta_{y_d,z_n}^{(I)(w)})$, where $m^{(w)}$ is the observed background term frequency, η_{y_d}, η_{z_n} and η_{y_d,z_n} denote the log frequency deviation representing topic z_n, facet y_d, and the second-order interaction part respectively. Superscripts T, A and I respectively denote the index of the topic, facet, and second-order interaction.

The SAGE model:



$$Y = \{y_{Sentiment} \in \{\text{positive, negative}\},\\ y_{Domain} \in \{\text{hotel, restaurant, doctor}\},\\ y_{Source} \in \{\text{employee, turker, customer}\}\}$$

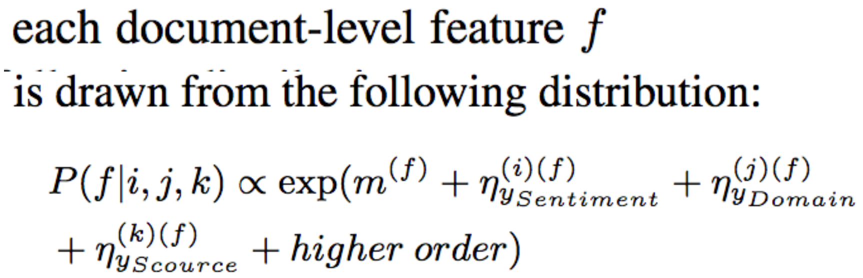
We model three η's, one for each type of y. Let i, j, k denote the index of the different types of y, so that each term w is drawn as follows:

$$P(w|i,j,k) \propto \exp(m^{(w)} + \eta_{y_{Sentiment}}^{(i)(w)}\\ + \eta_{y_{Domain}}^{(j)(w)} + \eta_{y_{Scource}}^{(k)(w)} + \textit{higher order})$$

where the *higher order* parts denote the interactions between different facets.

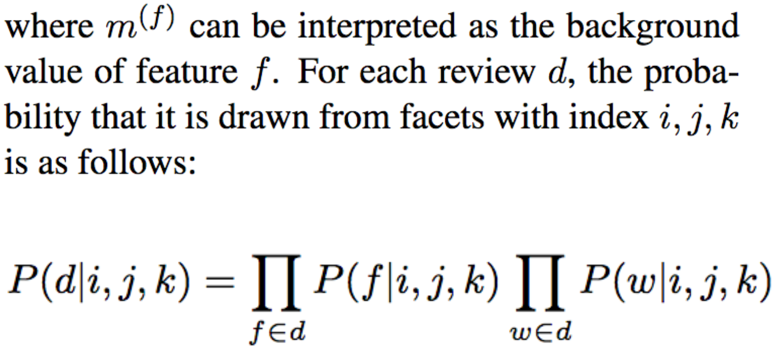
Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Generality of deception detection (cross-domain deceptive data) [8]
      - The mathematical model:



each document-level feature f is drawn from the following distribution:

$$P(f|i,j,k) \propto \exp(m^{(f)} + \eta^{(i)(f)}_{y_{Sentiment}} + \eta^{(j)(f)}_{y_{Domain}} + \eta^{(k)(f)}_{y_{Scource}} + higher\ order)$$

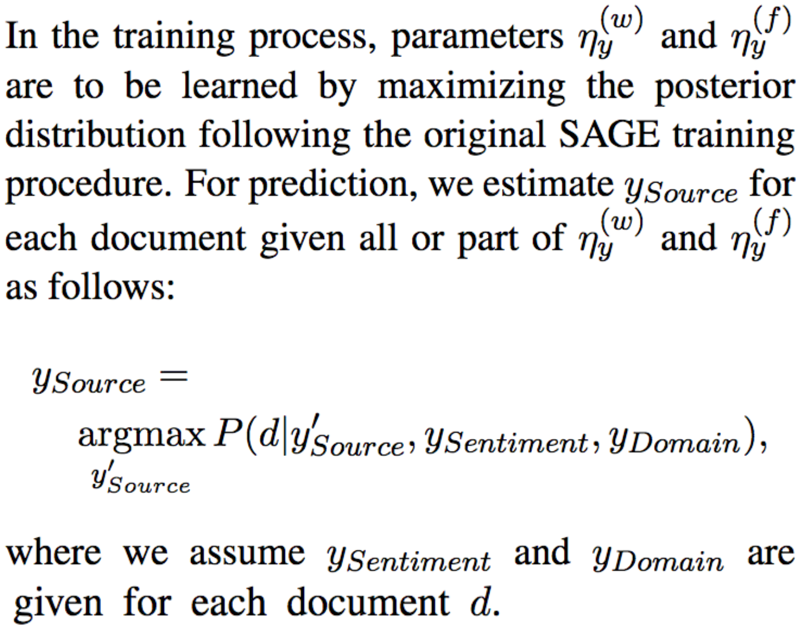


where $m^{(f)}$ can be interpreted as the background value of feature f. For each review d, the probability that it is drawn from facets with index i, j, k is as follows:

$$P(d|i,j,k) = \prod_{f \in d} P(f|i,j,k) \prod_{w \in d} P(w|i,j,k)$$

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Generality of deception detection (cross-domain deceptive data) [8]
      - The mathematical model:



In the training process, parameters $\eta_y^{(w)}$ and $\eta_y^{(f)}$ are to be learned by maximizing the posterior distribution following the original SAGE training procedure. For prediction, we estimate y_{Source} for each document given all or part of $\eta_y^{(w)}$ and $\eta_y^{(f)}$ as follows:

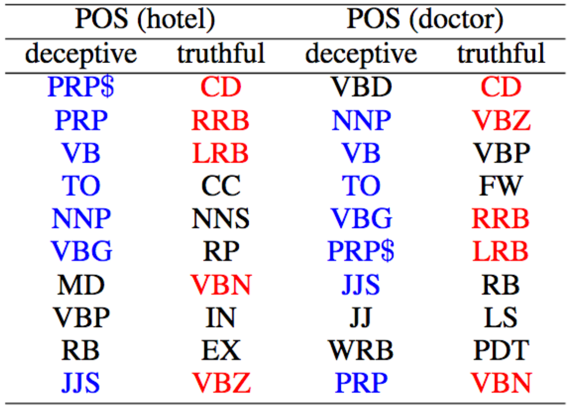
$$y_{Source} = \underset{y'_{Source}}{\operatorname{argmax}} P(d|y'_{Source}, y_{Sentiment}, y_{Domain}),$$

where we assume $y_{Sentiment}$ and y_{Domain} are given for each document d.

More math: See [9]

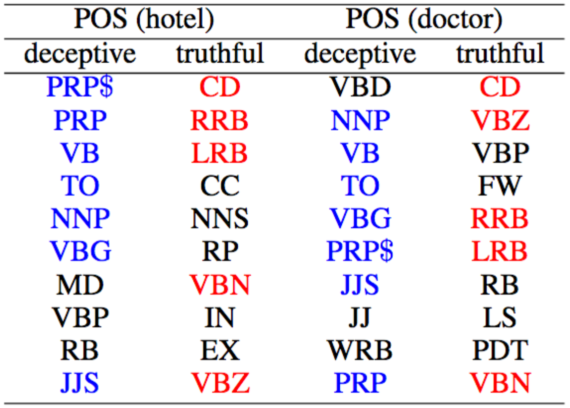
Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Generality of deception detection (cross-domain deceptive data) [8]
      - Some results:
        - » N (Noun), JJ (Adjective), IN(Preposition or subordinating conjunction) and DT(Determiner), V (Verb), RB (Adverb), PRP (Pro-nouns, both personal and possessive) and PDT(Pre-Determiner)



POS (hotel)		POS (doctor)	
deceptive	truthful	deceptive	truthful
PRP$	CD	VBD	CD
PRP	RRB	NNP	VBZ
VB	LRB	VB	VBP
TO	CC	TO	FW
NNP	NNS	VBG	RRB
VBG	RP	PRP$	LRB
MD	VBN	JJS	RB
VBP	IN	JJ	LS
RB	EX	WRB	PDT
JJS	VBZ	PRP	VBN

Table 6: Top weighted POS features for Turker vs Customer in Doctor and Hotel reviews. Blue denotes shared positive (deceptive) features and red denotes negative (truthful) features.



POS (hotel)		POS (doctor)	
deceptive	truthful	deceptive	truthful
PRP$	CD	VBD	CD
PRP	RRB	NNP	VBZ
VB	LRB	VB	VBP
TO	CC	TO	FW
NNP	NNS	VBG	RRB
VBG	RP	PRP$	LRB
MD	VBN	JJS	RB
VBP	IN	JJ	LS
RB	EX	WRB	PDT
JJS	VBZ	PRP	VBN

Table 6: Top weighted POS features for Turker vs Customer in Doctor and Hotel reviews. Blue denotes shared positive (deceptive) features and red denotes negative (truthful) features.

Detecting Spams/Fake Reviews/Phishing Emails

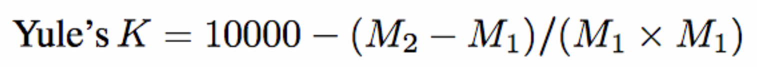
- Review centric spam detection [1]
  - Supervised learning
    - Detecting Deceptive Reviews Using Lexical and Syntactic Features [10]
      - Based on Stylometric [11]
      - The features are categorized as either lexical features or syntactic features.
        - 234 stylometric features; the features are categorized into lexical (character-based and word-based) and syntactic features.
      - Lexical features are character/word based features
      - Syntactic features represent the writing style of the reviewers at the sentence level, such as occurrences of punctuations or function words.
      - using a hybrid set of both the lexical and syntactic features and comparing this with using either lexical or syntactic features alone, a machine learning classifier can perform better.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Detecting Deceptive Reviews Using Lexical and Syntactic Features [10]
      - lexical features help to learn about the preferred use of each characters and words of a reviewer.
      - Lexical features can be divided into two categories, character-based and word-based features.
      - Character-based: They contain character count (N), ratio of digits to N, ratio of letters to N, ratio of uppercase letters to N, ratio of spaces to N, ratio of tabs to N, occurrences of alphabets (A-Z) (26 features), and occurrences of special characters: < > ^ % I { } [l \ / # ~ + - ;-* & @ $ (20 features).
      - Lexical word-based features: token count (T), average sentence length in terms of characters, average token length, ratio of characters in words to N, ratio of short words (1-3 characters) to T, ratio of word length frequency distribution to T (16 features), ratio of types to T, vocabulary richness to parametrize vocabulary richness in each of reviews.
      - Vocabulary richness measures the distribution of word frequencies. Yule proposed a measure, so called the Yule's K value:

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Detecting Deceptive Reviews Using Lexical and Syntactic Features [10]
      - Yule's K value:

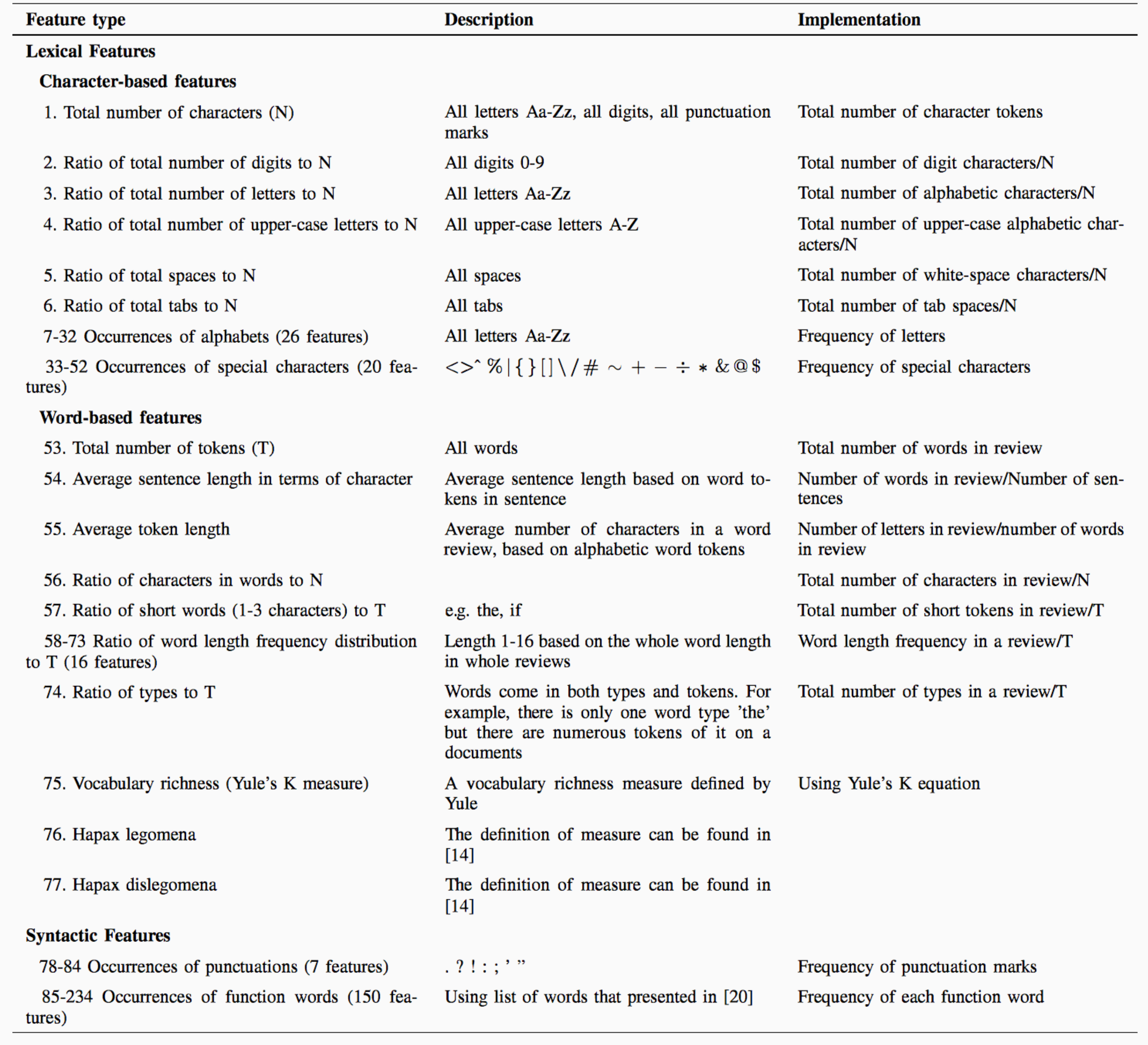


$$\text{Yule's } K = 10000 - (M_2 - M_1)/(M_1 \times M_1)$$

      - Where M1 is the number of all word forms a text consists of and M2 is the sum of the products of each observed frequency to the power of two and the number of word types observed with that frequency.
      - The larger Yule's K can result the smaller diversity of the vocabulary.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Supervised learning
    - Detecting Deceptive Reviews Using Lexical and Syntactic Features [10]
      - Syntactic features represent the writing style of reviewers at the sentence level. Syntactic features include: occurrences of punctuations that include. ? ! : ; , " (7 features).



Feature type	Description	Implementation
Lexical Features		
Character-based features		
1. Total number of characters (N)	All letters Aa-Zz, all digits, all punctuation marks	Total number of character tokens
2. Ratio of total number of digits to N	All digits 0-9	Total number of digit characters/N
3. Ratio of total number of letters to N	All letters Aa-Zz	Total number of alphabetic characters/N
4. Ratio of total number of upper-case letters to N	All upper-case letters A-Z	Total number of upper-case alphabetic characters/N
5. Ratio of total spaces to N	All spaces	Total number of white-space characters/N
6. Ratio of total tabs to N	All tabs	Total number of tab spaces/N
7-32 Occurrences of alphabets (26 features)	All letters Aa-Zz	Frequency of letters
33-52 Occurrences of special characters (20 features)	<>^ % | { } [ ] \ / # ∼ + − ÷ ∗ & @ $	Frequency of special characters
Word-based features		
53. Total number of tokens (T)	All words	Total number of words in review
54. Average sentence length in terms of character	Average sentence length based on word tokens in sentence	Number of words in review/Number of sentences
55. Average token length	Average number of characters in a word review, based on alphabetic word tokens	Number of letters in review/number of words in review
56. Ratio of characters in words to N		Total number of characters in review/N
57. Ratio of short words (1-3 characters) to T	e.g. the, if	Total number of short tokens in review/T
58-73 Ratio of word length frequency distribution to T (16 features)	Length 1-16 based on the whole word length in whole reviews	Word length frequency in a review/T
74. Ratio of types to T	Words come in both types and tokens. For example, there is only one word type 'the' but there are numerous tokens of it on a documents	Total number of types in a review/T
75. Vocabulary richness (Yule's K measure)	A vocabulary richness measure defined by Yule	Using Yule's K equation
76. Hapax legomena	The definition of measure can be found in [14]	
77. Hapax dislegomena	The definition of measure can be found in [14]	
Syntactic Features		
78-84 Occurrences of punctuations (7 features)	. ? ! : ; ' ”	Frequency of punctuation marks
85-234 Occurrences of function words (150 features)	Using list of words that presented in [20]	Frequency of each function word

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - Benefits:
        - » unsupervised spam detection model is able to address the problem of “missing features in an individual review” related to untruthful review detection.
        - » A semantic language model to estimate the overlapping of semantic contents among reviews, and hence to identify untruthful reviews.
        - » Able to take “substituted” terms into account when estimating the semantic content similarity among reviews.
        - » Address the knowledge acquisition problem by developing a high-order concept association mining method to extract context-sensitive concept association knowledge.
        - » This knowledge can then be utilized by the proposed semantic language model to ascertain possible “concept substitutions” in untruthful reviews.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - Modules:
        - » Module 1: The selection of the detection scope (e.g., all the reviews of a product)
        - » Module 2: Use of APIs to retrieve consumer reviews
        - » Module 3: if APIs are not provided, then general search engines such as Google and dedicated crawler programs will be invoked to retrieve the online reviews.
        - » Module 4: Document preprocessing procedures (e.g., stop-word removal, Part-of-Speech (POS) tagging, and stemming), are applied
        - » Module 5: Then the high-order concept association mining module is invoked to extract the prominent concepts and their high-order associations for each product category. These association relationships are used to bootstrap the performance of the semantic language model to detect untruthful reviews using the obfuscation strategies exercised by spammers. The text mining module is invoked periodically and executed as a background task.
        - » Module 6: non-review detection is performed by a supervised SVM classifier
        - » Module 7: untruthful review detection is carried out independently by an unsupervised probabilistic language model.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - The deriving idea: indirectly estimate the degree of “untruthfulness” based on the “overlapping” of the semantic contents of reviews.
      - If the semantic contents of a review are mainly generated from another review, this suggests that both reviews may not sincerely reflect writers’ true opinions
    - Probabilistic ranking of suspicious fake reviews and facilitate users’ final decisions about the truthfulness of reviews.
    - General approach:
      1. Extend the well-known Language Modeling (LM) to develop a semantic-based smoothing method to estimate the likelihood of semantic content generation among reviews.
      2. Apply Kullback-Leibler (KL) divergence to measure the distance of the language models that represent individual reviews.
    - The semantic language models should take into account relationships such as (“love”→“like”) when estimating the semantic similarity of reviews (use of WordNet)
      - E.g., “….. and I loved it” vs. “.... and I liked it”

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - For term association relationships not defined in WordNet (e.g., “fabulous”→“fantastic”), a text mining method needs to be applied to dynamically discover the term association relationships to detect the spammers’ obfuscation actions.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - "language model" is widely used to refer to a probability distribution M which represents the statistical regularities in the generation of a language
      - A language model is a probabilistic function that assigns a probability to a string t drawn from some vocabulary T
      - It has been applied to estimate the relevance of a document d with respect to a query q in Information Retrieval

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - The basic unigram language model is defined by:

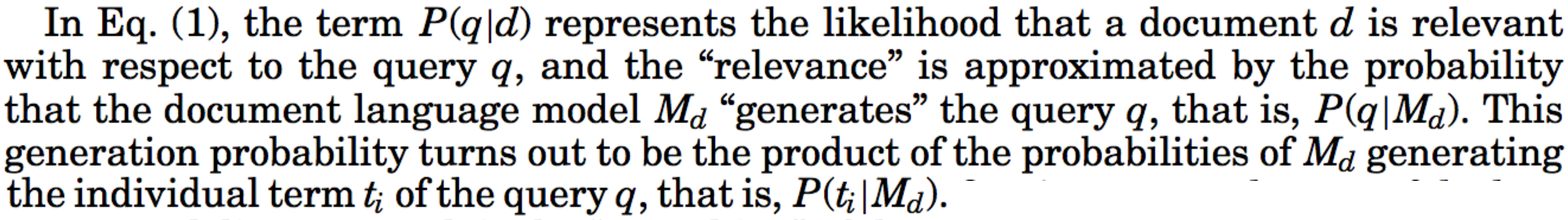
$$P(q|d) \propto P(q|M_d) = \prod_{t_i \in q} P(t_i|M_d) \tag{1}$$

$$P(t_i|M_d) = (1-\lambda)P_{ML}(t_i|M_d) + \lambda P_{ML}(t_i|M_D) \tag{2}$$

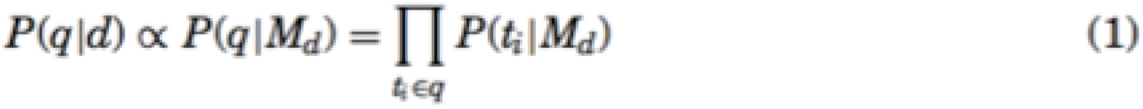
$$P_{ML}(t_i|M_D) = \frac{tf(t_i, D)}{|D|}. \tag{3}$$

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - The basic unigram language model is defined by:



In Eq. (1), the term $P(q|d)$ represents the likelihood that a document d is relevant with respect to the query q, and the "relevance" is approximated by the probability that the document language model M_d "generates" the query q, that is, $P(q|M_d)$. This generation probability turns out to be the product of the probabilities of M_d generating the individual term t_i of the query q, that is, $P(t_i|M_d)$.



$$P(q|d) \propto P(q|M_d) = \prod_{t_i \in q} P(t_i|M_d) \qquad (1)$$

Detecting Spams/Fake Reviews/Phishing Emails

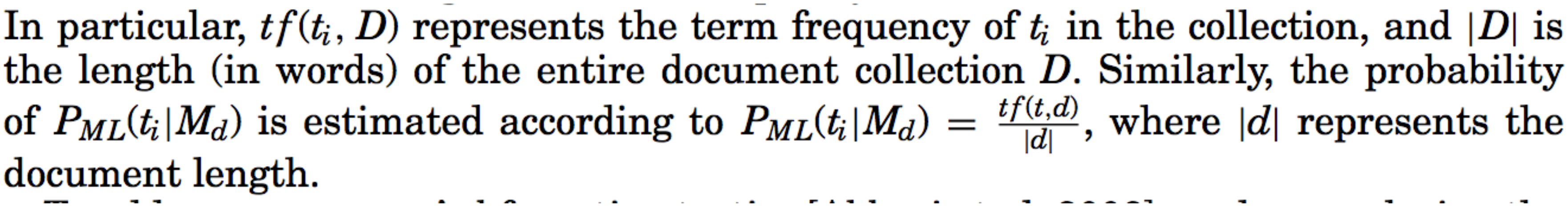
- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - “smoothing” of the term probability
        - » If a query term ti is not found in documentd, it does not necessarily mean that the document is not about ti because semantically similar terms such as synonyms could have been used to compose the document.
        - » the objective of smoothing the document model is to reduce the chance of overestimating the generation probability for terms observed in the document by applying a factor(1−λ) to the maximum likelihood language model $P_{ML}(t_i|M_d)$.
        - » To alleviate the problem of underestimating the generation probability for terms not observed in the document, a factor λ is applied to the maximum likelihood estimation of ti with respect to the entire collection D, that is, the collection language model defined $P_{ML}(t_i|M_d)$. in Eq. (2)
        - » The term λ is called the Jelinek-Mercer smoothing parameter, which usually assumes values in the range of [0.1, 0.7]

$$P(t_i|M_d) = (1 - \lambda)P_{ML}(t_i|M_d) + \lambda P_{ML}(t_i|M_D) \qquad (2)$$

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - “smoothing” of the term probability
        - » Eq. (3) defines the Jelinek-Mercer smoothing process, where the probability of an unobserved term ti is estimated according to its term frequency in the entire document collection D

$$P_{ML}(t_i|M_D) = \frac{tf(t_i, D)}{|D|}. \tag{3}$$



In particular, $tf(t_i, D)$ represents the term frequency of t_i in the collection, and $|D|$ is the length (in words) of the entire document collection D. Similarly, the probability of $P_{ML}(t_i|M_d)$ is estimated according to $P_{ML}(t_i|M_d) = \frac{tf(t,d)}{|d|}$, where $|d|$ represents the document length.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - Addressing obfuscation problem
        - » To address spammers' obfuscation tactics, such as replacing the word "like" by "love", a better way to estimate the probability of an unseen term in a document (e.g., a review) is to take into account the relationship of ("love"→"like"), as the term "like" is a synonym of the term "love" according to WordNet.
        - » The goal is to assign a more reasonable probability to an unseen term when evaluating two reviews. This can be achieved using the semantic language model defined according to Eq. (4).

$$\begin{aligned} P_{SEM}(t_i|M_d) &= \frac{\sum_{t_i,t_j \in R} P(t_i|t_j)P_{ML}(t_j|M_d)}{|R|} \\ &= \frac{\sum_{t_i,t_j \in R} P(t_j \rightarrow t_i)P_{ML}(t_j|M_d)}{|R|}, \end{aligned} \tag{4}$$

        - » Where P(tj→ti) is the certainty of the term association relationship between ti and tj, and is derived using our text mining method.

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - Addressing obfuscation problem
        - » The basic intuition of Eq. (4) is that if a term such as “like” (ti) is not found in a document (review), but the term “love” (tj)isfound in the document, and a term association such as “love”→“like” is established(according to WordNet or high-order concept mining), then the generation probability of $P_{ML}(\text{“love”}|M_d)$ can be used to estimate $P_{ML}(\text{“like”}|M_d)$.
        - » In Eq. (4) the term R represents the set of term relationships in the form of tj→ti,and |R| is the cardinality of the set R. As a number of term association relationships may be discovered via text mining, only the top n associations ranked by P(ti|tj) for each term ti are considered

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - Additional smoothing
        - » There may be unseen terms not captured in the term relationship set R.
        - » Therefore, the collection language model is still required to smooth the overall language model. Therefore, the semantic language model for review spam detection is defined according to Eq. (5). First, a Jelinek-Mercer smoothing parameter ν is applied to smooth the unseen terms in a document, as these unseen terms may be related to some other terms captured in the term relationship set R.
        - » Then, a second Jelinek-Mercer smoothing parameter μ is applied to conduct further smoothing for the unseen terms not captured by any term relationships.

$$\begin{aligned} P(t_i|M_d) = &(1-\mu)((1-\nu)P_{ML}(t_i|M_d) + \nu P_{SEM}(t_i|M_d)) \\ &+ \mu P_{ML}(t_i|M_D) \end{aligned} \qquad (5)$$

Detecting Spams/Fake Reviews/Phishing Emails

- Review centric spam detection [1]
  - Unsupervised learning
    - Problem with Supervised learning: Difficulty of producing labeled datasets
    - Text mining and probabilistic language modeling [12]
      - Estimate the distance between two probability distributions
        - » Use Kullback and Leibler (KL) divergence
        - » We apply a negative KL divergence to measure the similarity between pairs of language models, such as Md1 and Md2, representing two reviews.
        - » If the negative KL divergence of the two language models is large, then the distance of the corresponding probability distributions is considered small.
        - » It means the semantic contents of the pair of reviews are quite similar, and they are likely to be spam.
        - » The final formulation of the untruthful review detection method underpinned by LM and KL divergence is defined by:

$$\begin{aligned} Score_{KL}(d_1, d_2) &= -KL(M_{d_1} || M_{d_2}) \\ &= -\sum_{t_i \in \{d_1 \cup d_2\}} P(t_i|M_{d_1}) \times \log_2 \frac{P(t_i|M_{d_1})}{P(t_i|M_{d_2})}, \end{aligned} \tag{6}$$

        - » Where ti is a term appearing in either d1 or d2.

References

1. M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter and H. Al Najada, Survey of review spam detection using machine learning techniques
2. Dixit S, Agrawal AJ, Survey on review spam detection.
3. http://liwc.wpengine.com/how-it-works/
4. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods, Yla R. Tausczik1 and James W. Pennebaker
5. Learning to Detect Phishing Emails, I. Fette, N. Sadeh, A. Tomasic.
6. Phishing E-mail Detection Based on Structural Properties, M. Chandrasekaran, K.Narayanan and S.Upadhyaya
7. Opinion Spam and Analysis, Nitin Jindal and Bing Liu
8. Towards a General Rule for Identifying Deceptive Opinion Spam, J. Li, M. Ott, C. Cardie, E. Hovy,
9. Sparse Additive Generative Models of Text, J. Eisenstein, A. Ahmed, E.P. Xing

References

10. Detecting Deceptive Reviews Using Lexical and Syntactic Features, S. Shojaee, M. A. A. Muradt, A. BB. Azman, N. M. Sharefl and S. Nadali
11. Stylometric identification in electronic markets: Scalability and robustness, Abbasi et al.
12. Text Mining and Probabilistic Language Modeling for Online Review Spam Detection RAYMOND Y. K. LAU, S. Y. LIAO, and RON CHI-WAI KWOK
13. L. F. Gutiérrez, F. Abri, M. Armstrong, A. S. Namin, and K. S. Jones, Email Embeddings for Phishing Detection
14. F. Abri, L. F. Guti´errez, A. S. Namin, K. S. Jones, and D. R. W. Sears, Linguistic Features for Detecting Fake Reviews
15. L. F. Gutiérrez, F. Abri, M. Armstrong, A. S. Namin, K. S. Jones, and D. R. W. Sears, Ensemble learning for detecting fake reviews