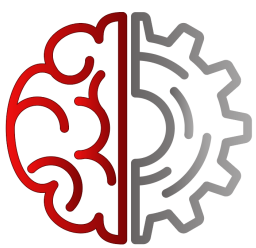
Natural Language Processing in E-Commerce
A Case Study
Product Classification in E-Commerce using Distributional Semantics

Vivek Gupta"[1]
Harish Karnick* , Ashendra Bansal' & Pradhuman Jhala'

School of Computing, University of Utah"
CSE Department, IIT Kanpur*
Flipkart.com' (India E-Commerce)



Data Science Club
University of Utah

[1]work done at IIT Kanpur

Outline

E-Commerce environment today

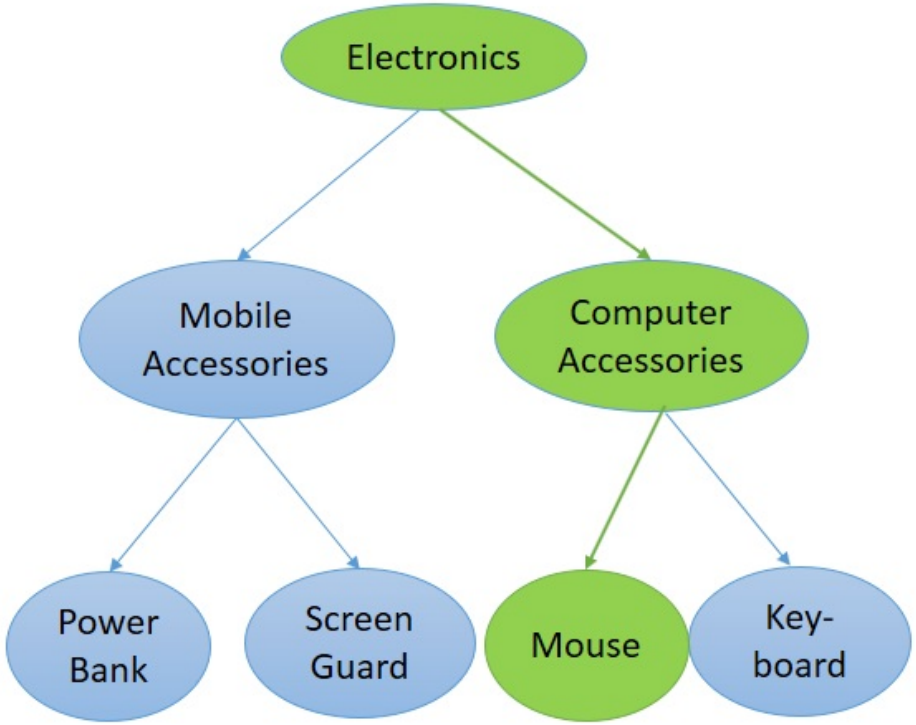
- Shift from single seller-many buyers to platform where many buyers-many sellers meet.
- Problem of product discovery: millions of products but very few actually seen by users due to search, navigation and display constraints.
- First step in improving search and navigation: use a catalogue (like a library). Tag product with a path label from a tree of class labels (ontology). This cataloguing can help in search and navigation and is also useful for internal logistics.
- Sellers typically give a textual description for their product.

Problem Statement

Task : To predict an appropriate taxonomy path for a given product in a predefined taxonomy from the product description

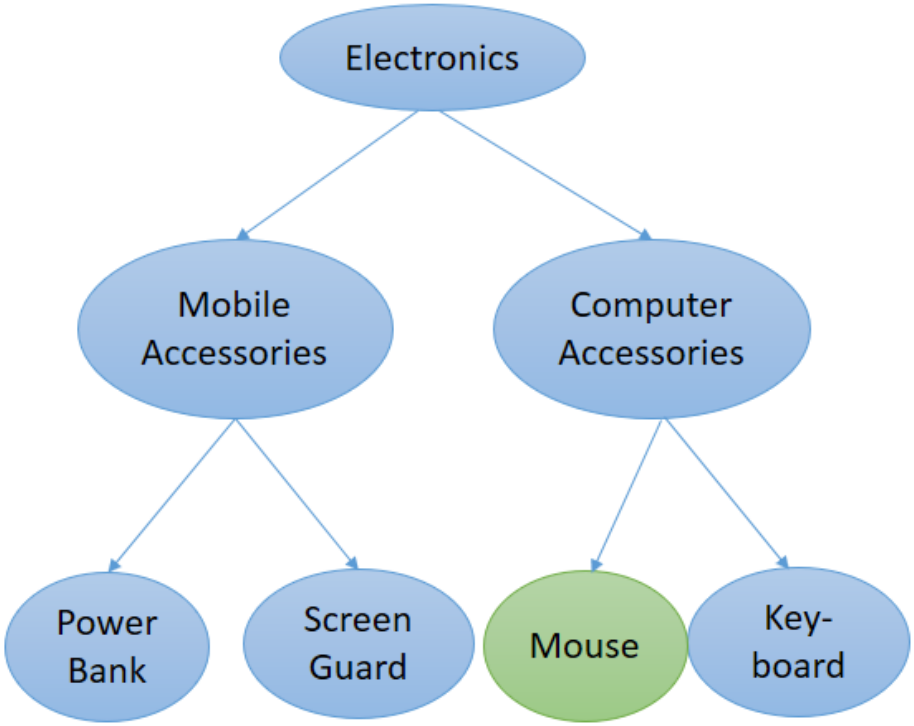
Title : 3-Button Optical Mouse from HP with scroll wheel

Description : 3-button mouse has a scroll wheel that lets you easily scroll through files. With optical sensor, it works on almost any surface. Have USB connectivity, you can easily connect it to your laptop or desktop. Design that gives it a stylish shape and also makes it comfortable to use.



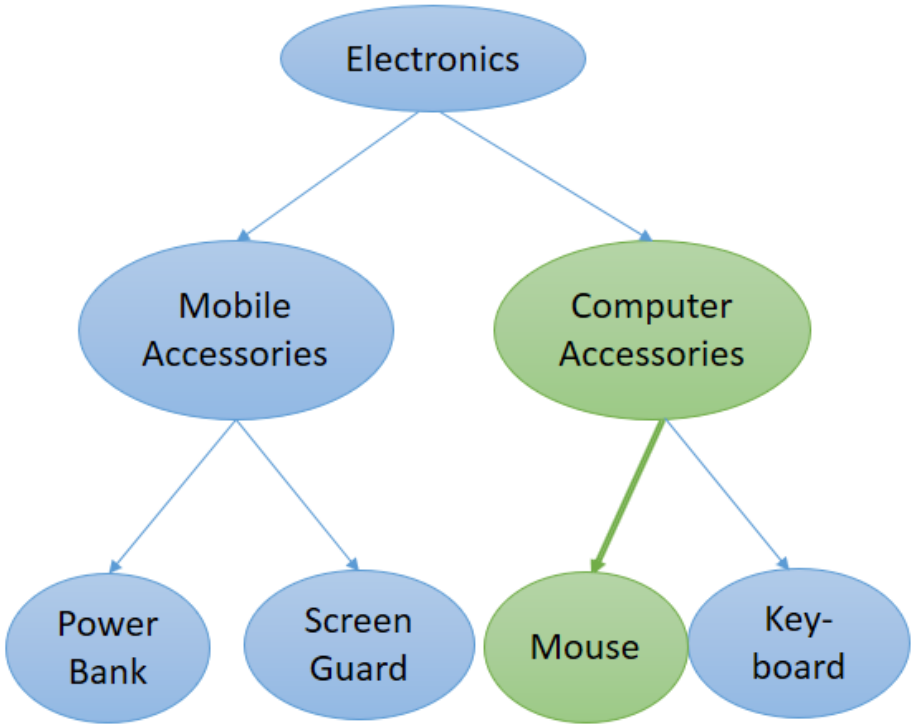
Challenges in Product Categorization

1. Hierarchical taxonomy imposes constraints on activation of labels



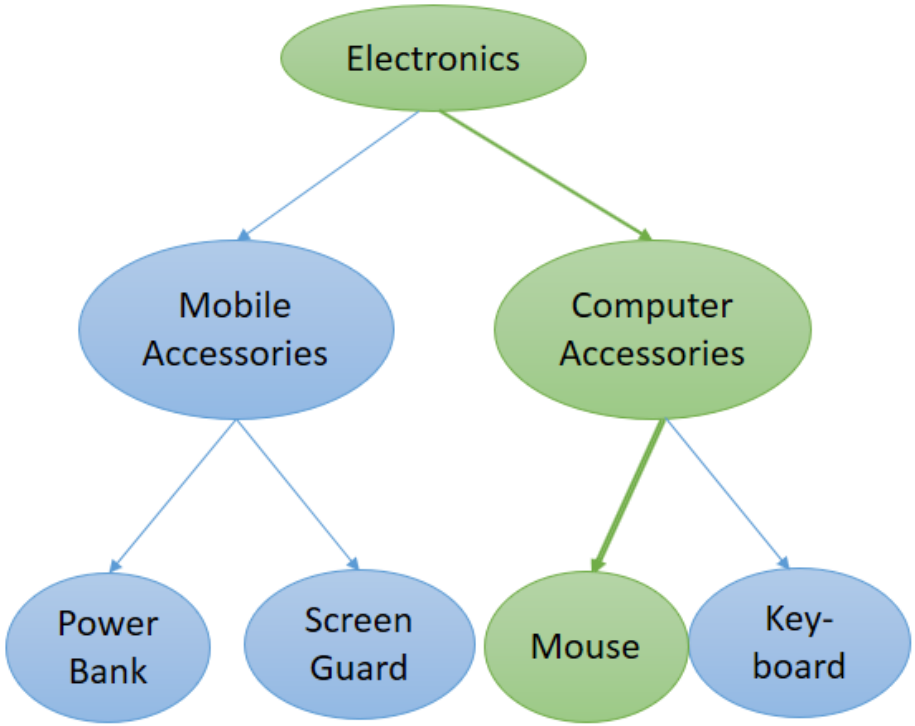
Challenges in Product Categorization

1. Hierarchical taxonomy imposes constraints on activation of labels



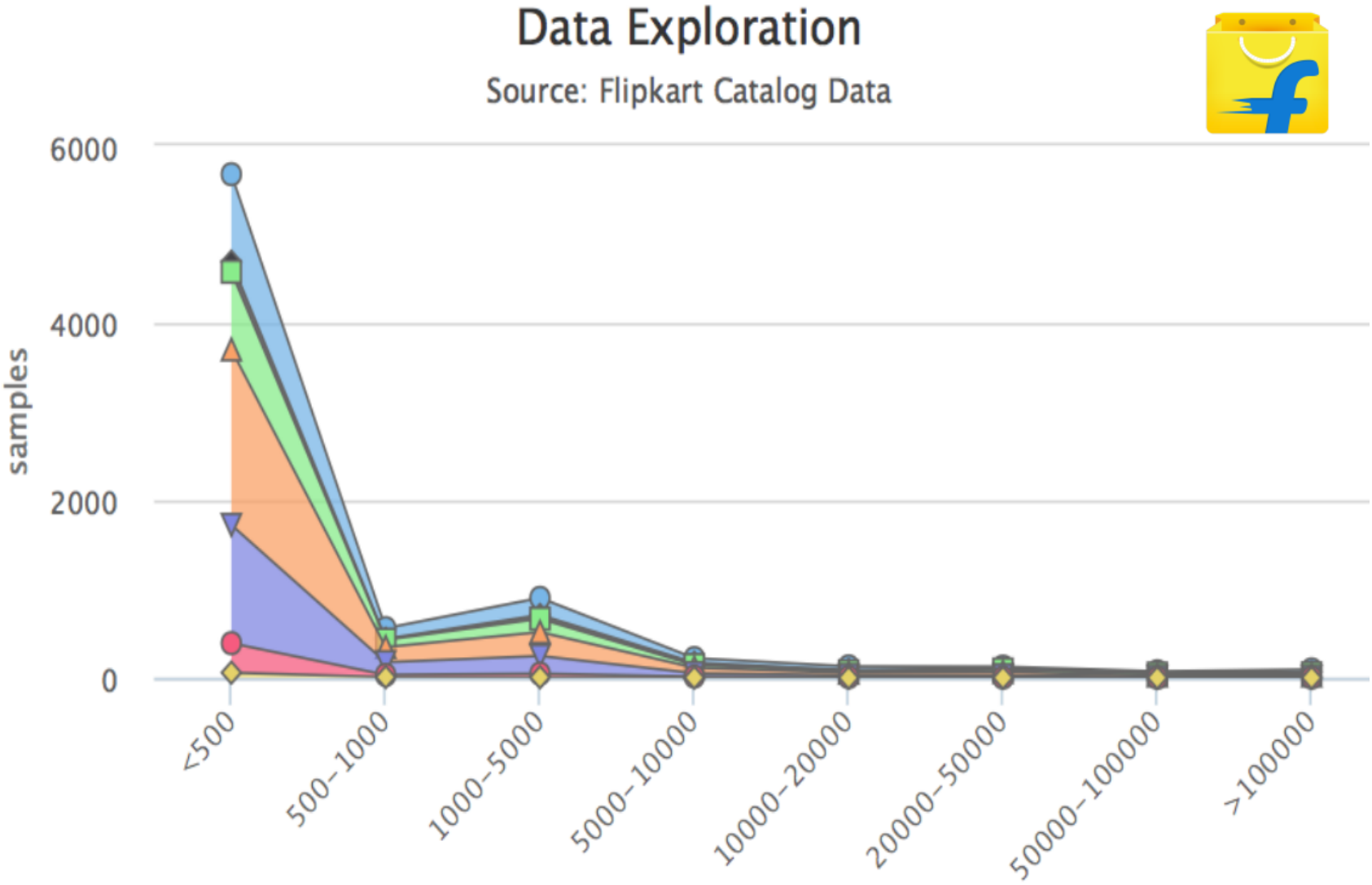
Challenges in Product Categorization

1. Hierarchical taxonomy imposes constraints on activation of labels



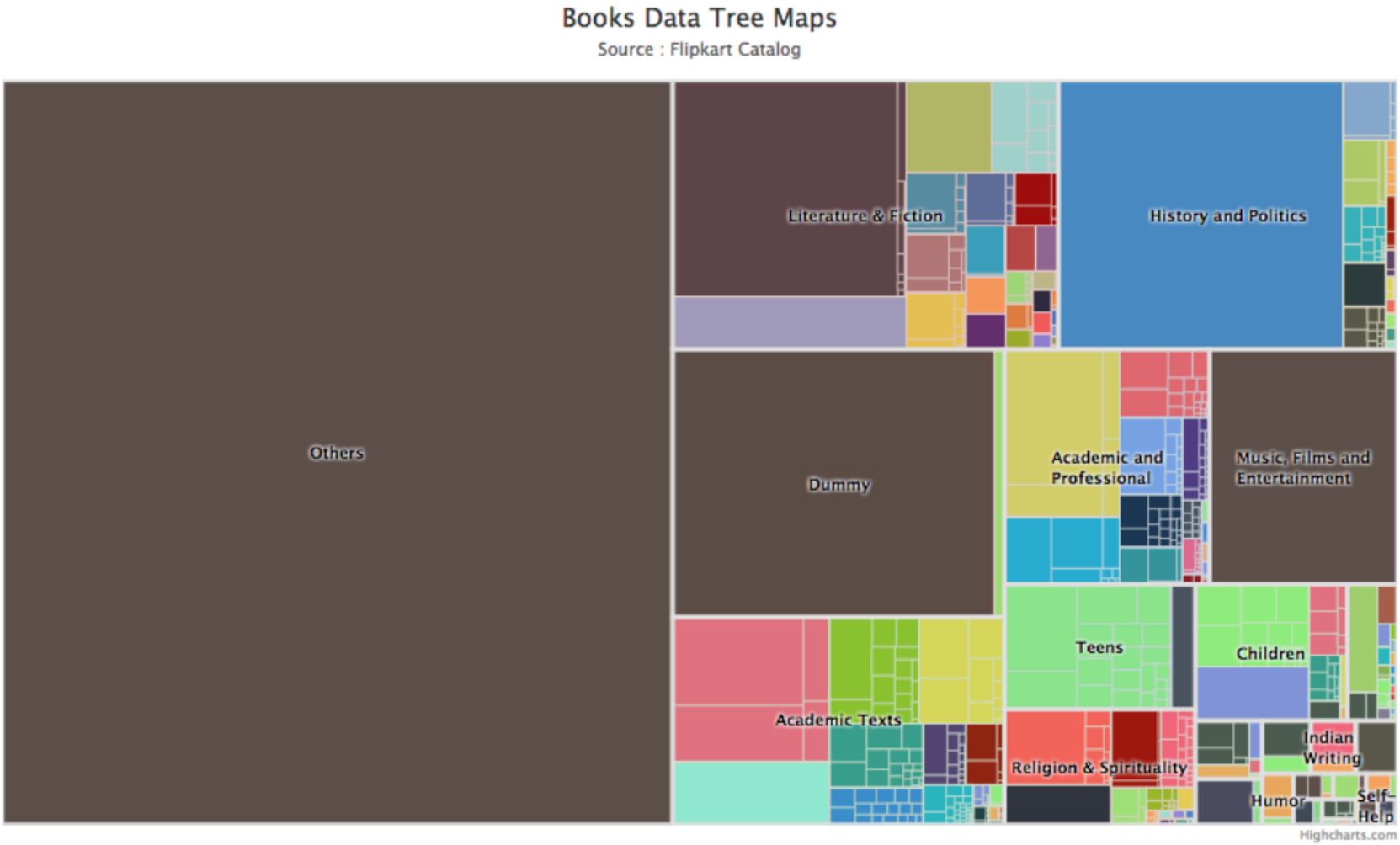
Challenges in Product Categorization

1. Categories have unbalanced data with a skewed long tailed distribution.



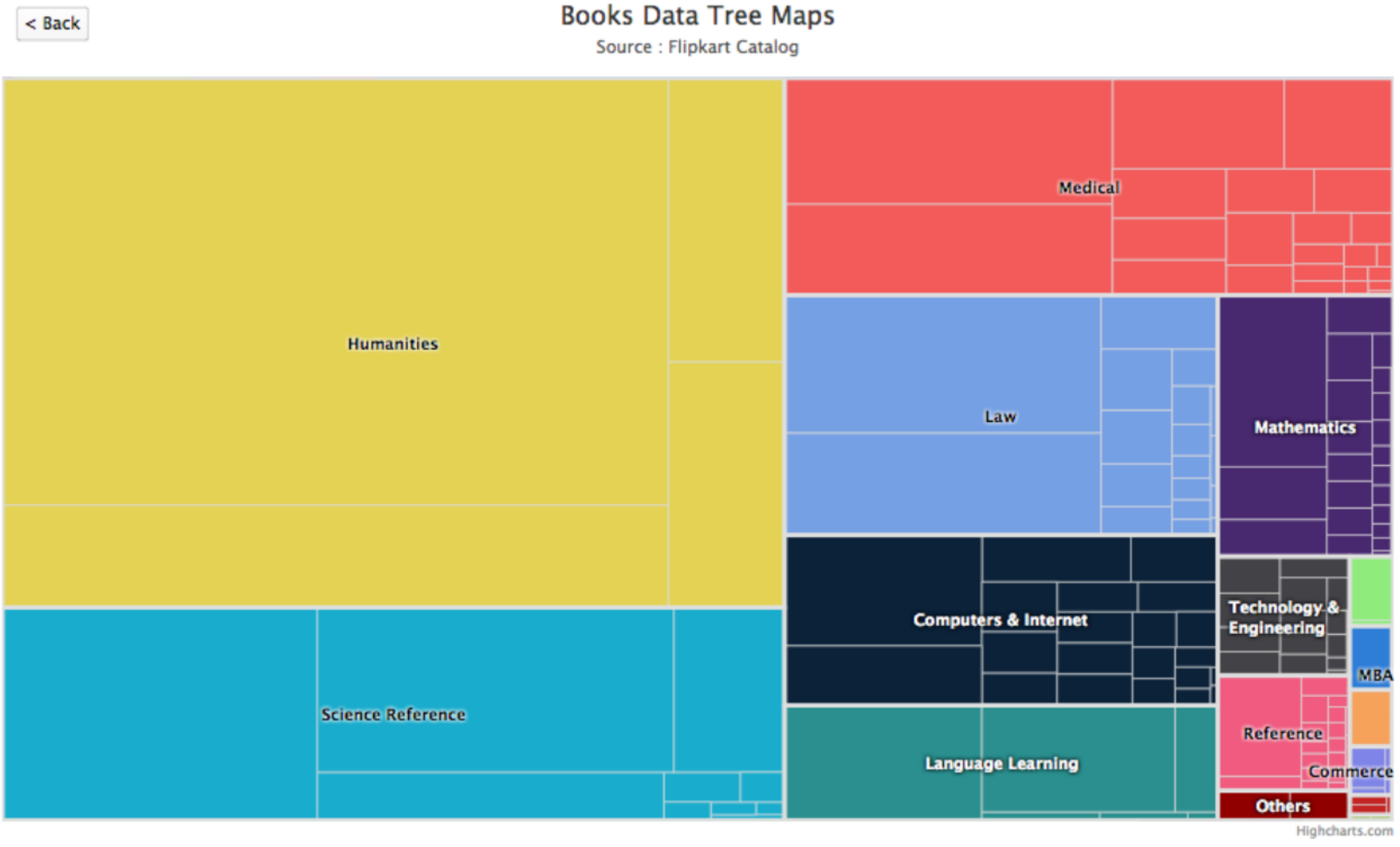
Challenges in Product Categorization

1. Categories have unbalanced data with a skewed long tailed distribution.



Challenges in Product Categorization

1. Categories have unbalanced data with a skewed long tailed distribution.



Outline

Challenges

P1 : Machine Learning models can't understand textual data

S1 : Represent text data in D dimension floating point numbers

P2 : Textual documents have lot of inherent noise and redundancy

S2 : Learning feature representation prune to noise and redundancy

Bag of Words

1. Each document d_i is represented in $v_{d_i} \in R^{|V|}$, $|V|$ is vocabulary size.
2. Each element in v_{d_i} denotes presence or absence of each word.
3. Drawback
    - High Dimensionality
    - Ignore word ordering
    - Ignore word context
    - Sparse Representation
    - No relative Weightage

Term Frequency Inverse Document Frequency

1. Each document d_i is represented in $v_{d_i} \in R^{|V|}$, $|V|$ is vocabulary size.
2. Each element in v_{d_i} product of term frequency and inverse document frequency tfidf(t,d) = tf(t,d) $\times$ idf(t,d), here idf(t,d) = $\log \frac{N+1}{n_k}$
3. Gives weights to terms which are less frequent and hence important
4. Drawback
   - High Dimensionality
   - Ignore word ordering
   - Ignore word context
   - Sparse Representation

Distributed Representation of Words

1. Each word $w_i \in V$ is represented using a vector $v_{w_i} \in R^k$
2. Hypothesis : Similar words occurring in similar context are closer in vector space.
3. Vectors v_{w_i} encode the semantics of word w_i.
4. Various models and algorithms
    1. Word2Vec CBoW (mikolov et. al.)
    2. Word2Vec SGNS (mikolov et. al.)
    3. Glove (Pennington et. al.)

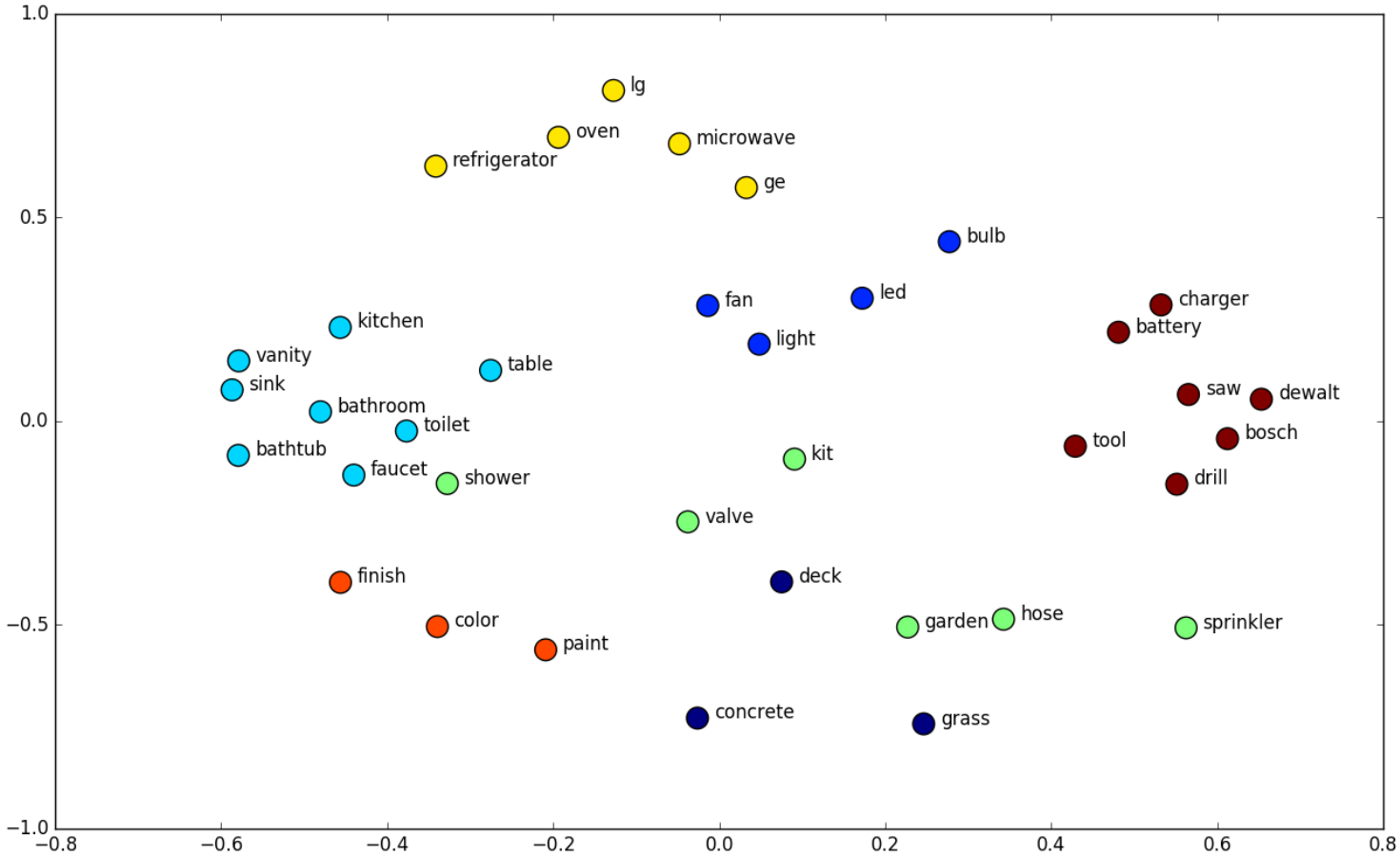


Figure: Semantically similar word, occur closely in vector space

Distributed Representation of Document[2]

1. Each document $d_i \in D$ is represented using a vector $v_{d_i} \in R^k$.
2. Vector v_{d_i} encode the semantics of the document d_i.
3. Various models and algorithms
    1. Doc2Vec PV-DM (mikolov et. al.)
    2. Doc2Vec PV-DBoW (mikolov et. al.)
    3. Glove parvec (Pennington et. al.)
4. Document vector are embedded in same space as word vectors

[2] Quoc V Le and Tomas Mikolov. "Distributed representations of sentences and documents". In: *arXiv preprint arXiv:1405.4053* (2014).

Outline

Tree Based Approach

1. High-level classifier serves as *gate* to lower level classifiers *experts* [Shen et al.].

2. Experts are complex classifier like svm and gates are simple classifiers like KNN

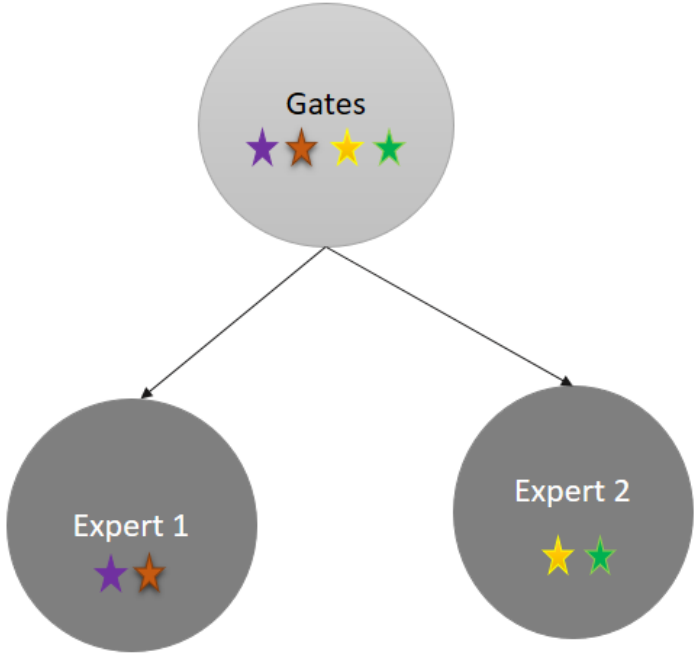


Figure: Gates Expert Classification Model. Each color star represent an class

Label Embedding Approach

1. Orthogonal label projection using Kernel methods.
2. Independently train classifiers on projected attributes in parallel.

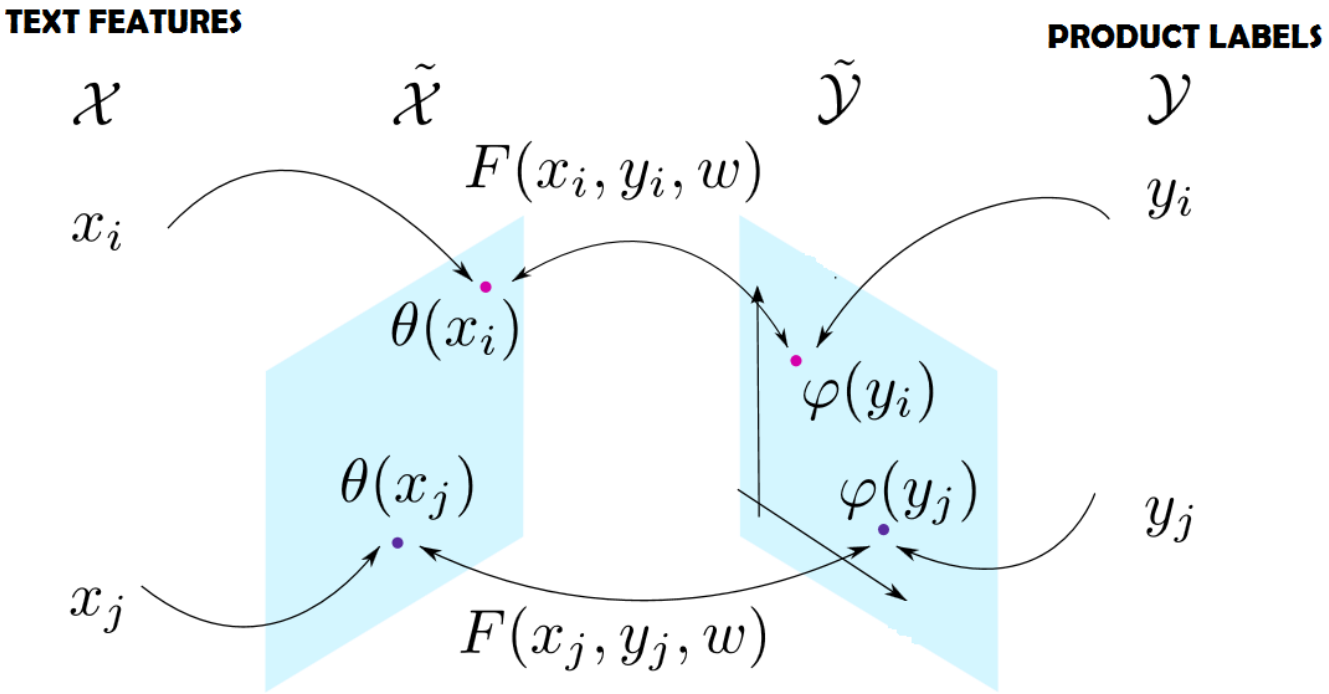


Figure: Orthogonal label projection using K-PCA

Outline

Weighted Average Word Vectors

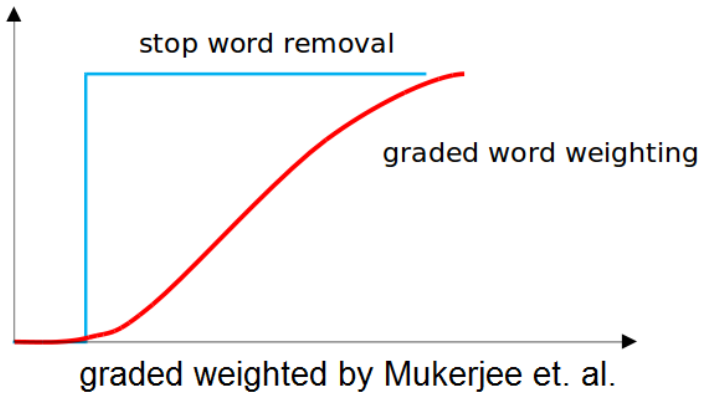
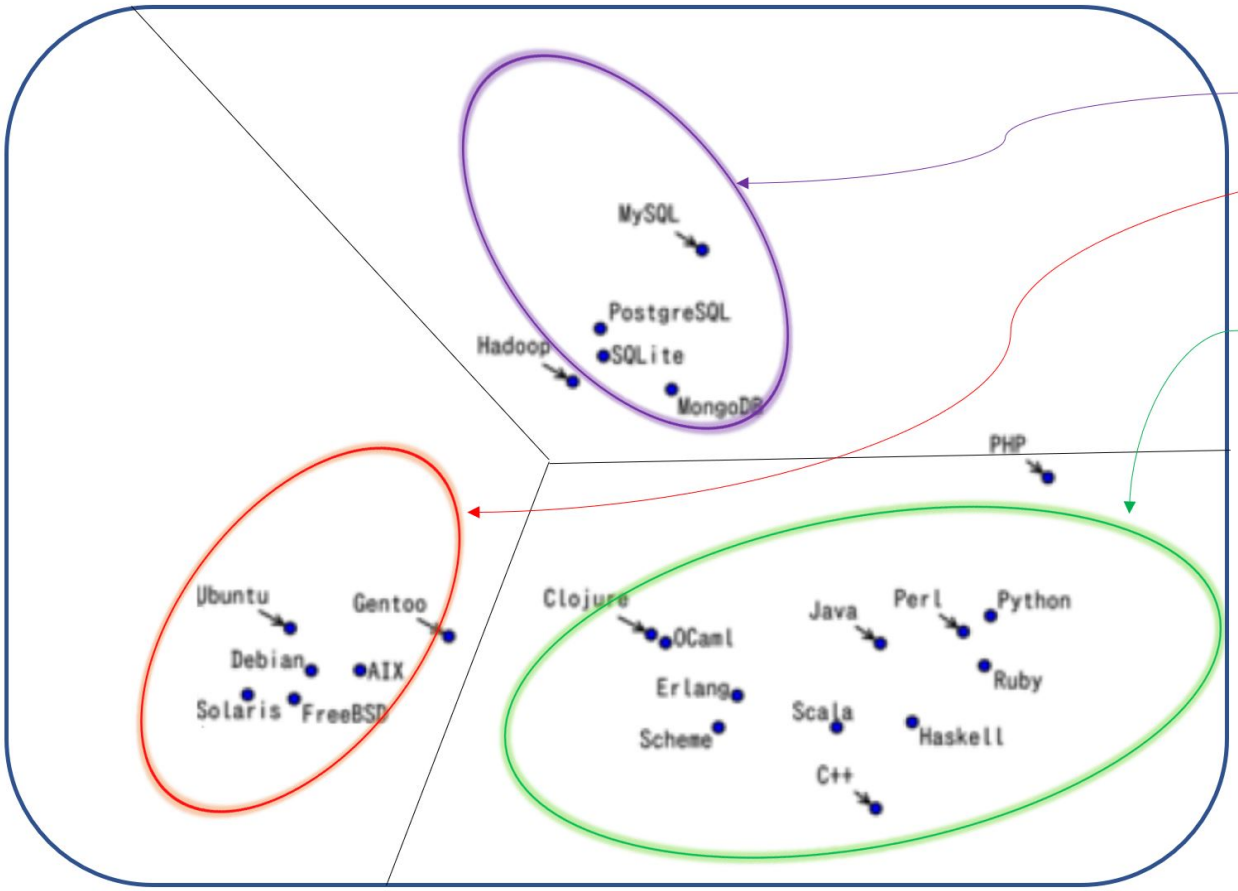


Figure: graded weighted averaging Mukerjee et al. (words are not equal)

- Methods better than simple averaging : Varying weights capture the relative importance of words.
- Standard assumption that all words within a document have same semantic topic applies.

Drawback of Average Vectors

Problem with Averaging



"Below are examples to write Hadoop Map-Reduce programs in Python on Ubuntu and Solaris systems"

Earlier Approach:
dv = wv(Hadoop) + wv(Python) + wv(Ubuntu) + wv(Solaris)

Here, + represent addition

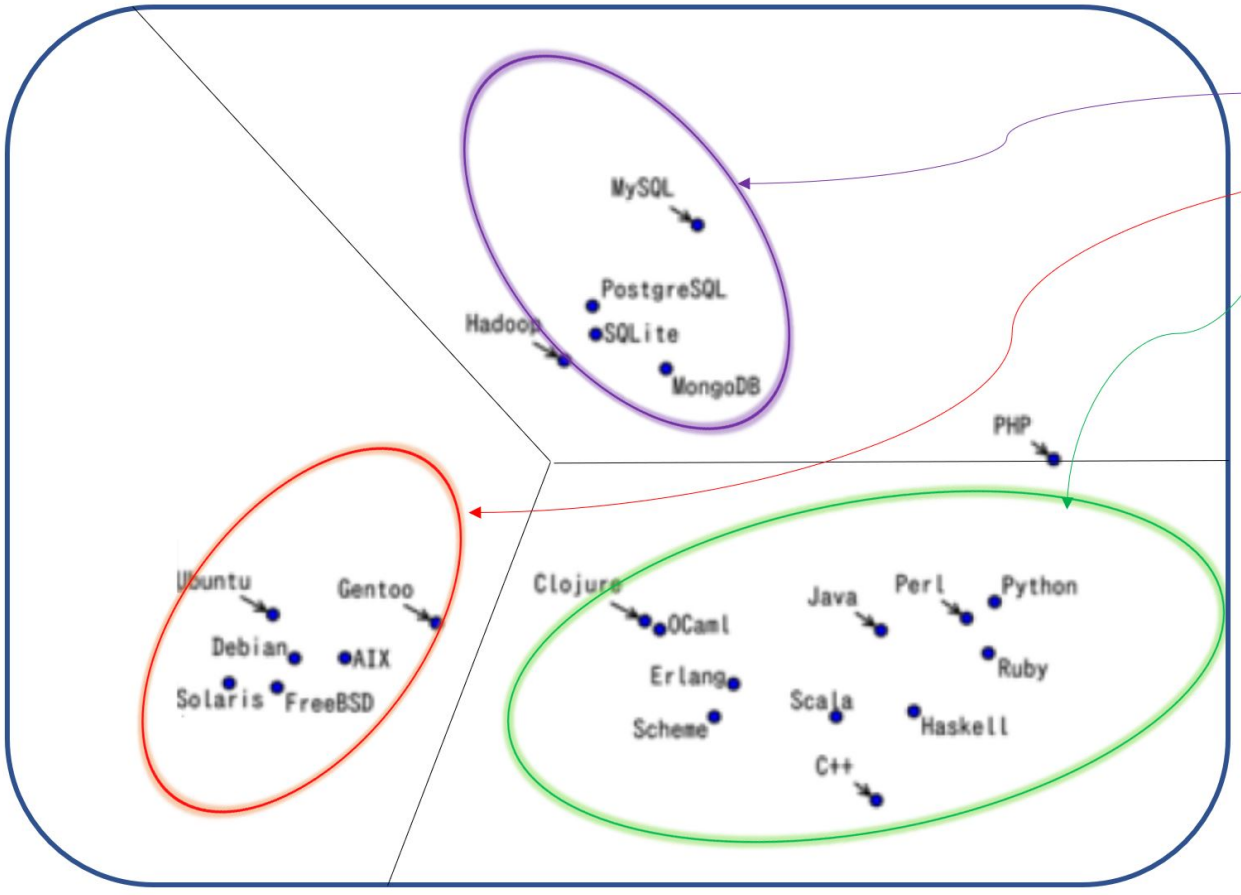
Semantically different words are directly average (TWE, NTSG)

Document and word vectors has same dimension (TWE, NTSG)

Outline

Proposed Formulation : gwBoWV

Our Approach BoWV (Gupta et al., 2016)



"Below are examples to write Hadoop Map-Reduce programs in Python on Ubuntu and Solaris systems"

BoWV Approach :

dv = wv(Hadoop) $\oplus$ wv(Python) $\oplus$ (wv(Ubuntu) + wv(Solaris)) $\oplus$

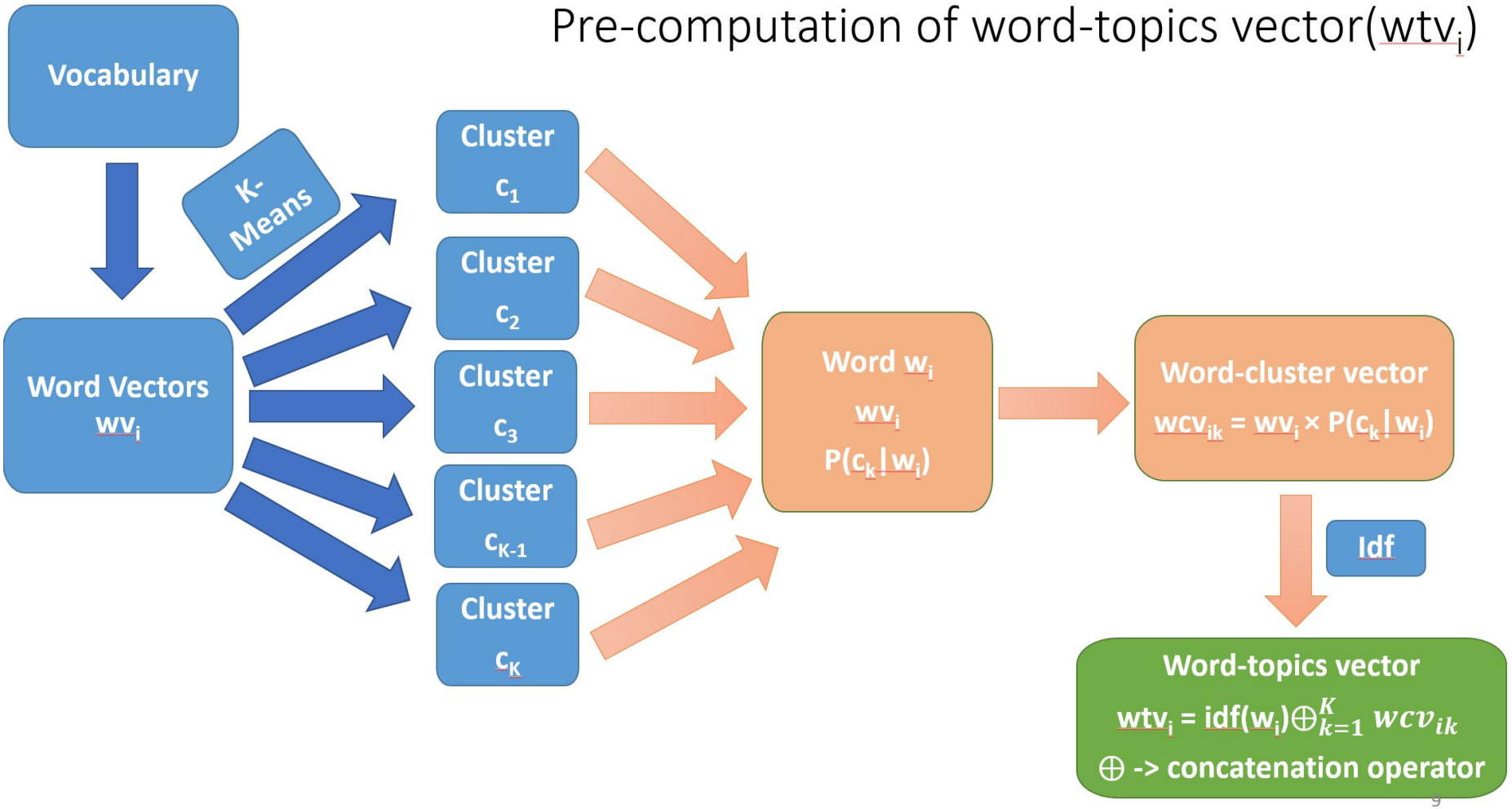
idf(Hadoop) $\oplus$ idf(Python) $\oplus$ (wv(Ubuntu) + wv(Solaris))

Where, + represent addition and $\oplus$ represent concatenation

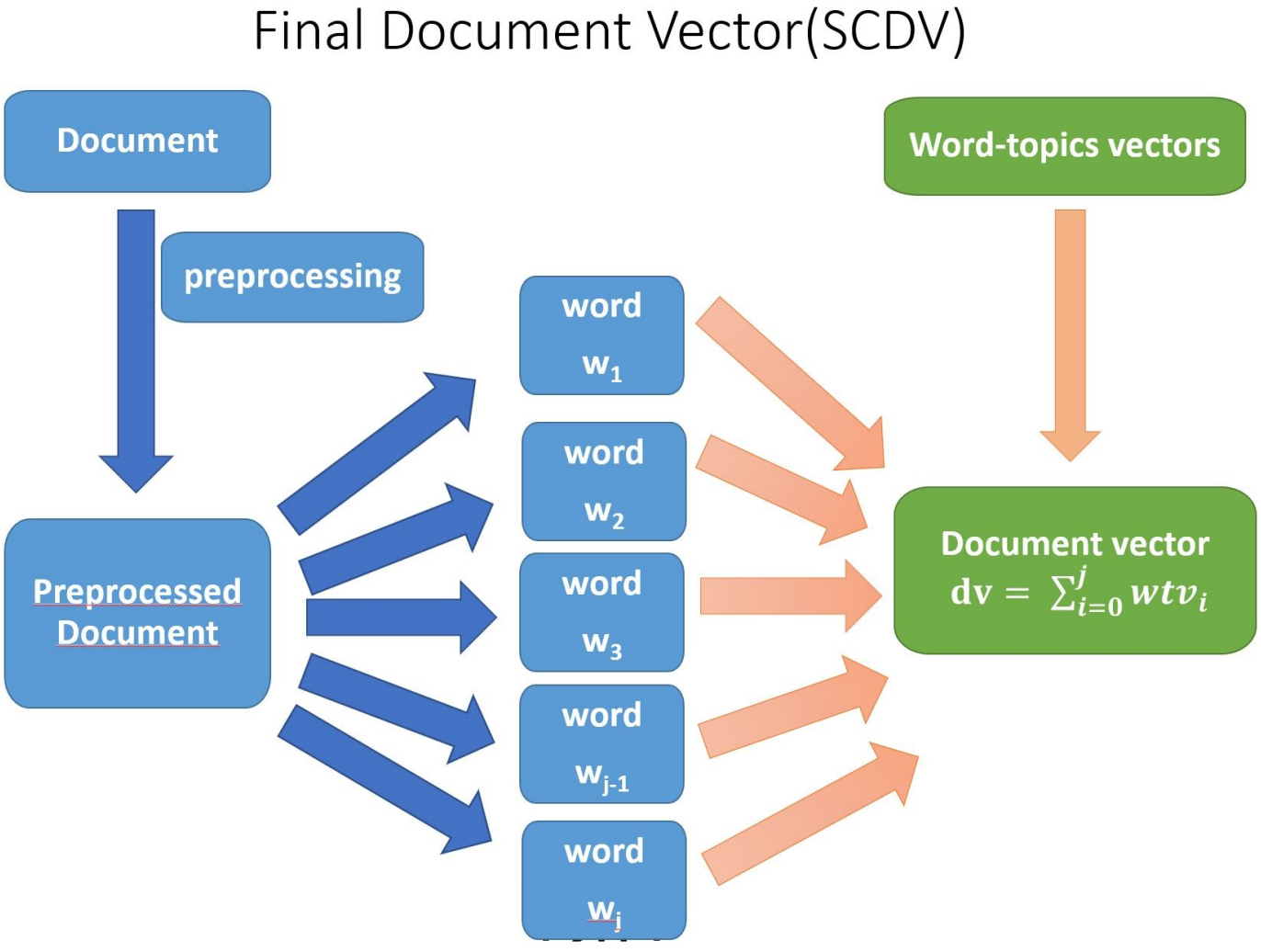
Used K-mean for clustering

Document Vector Dimension (K*d +K) > Word Vector Dimension (d)

Proposed Formulation : gwBoWV



Proposed Formulation : gwBoWV



Flowchart : gwBoWV

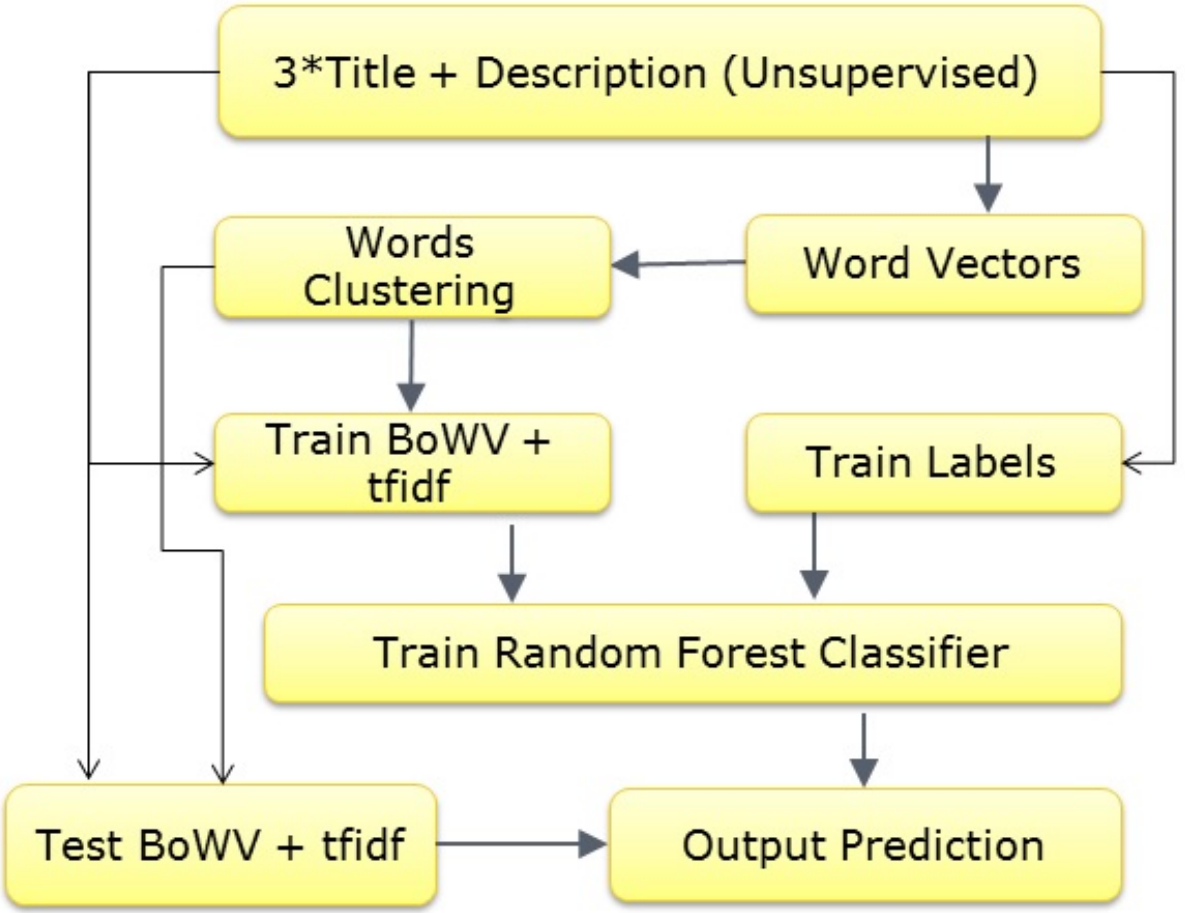


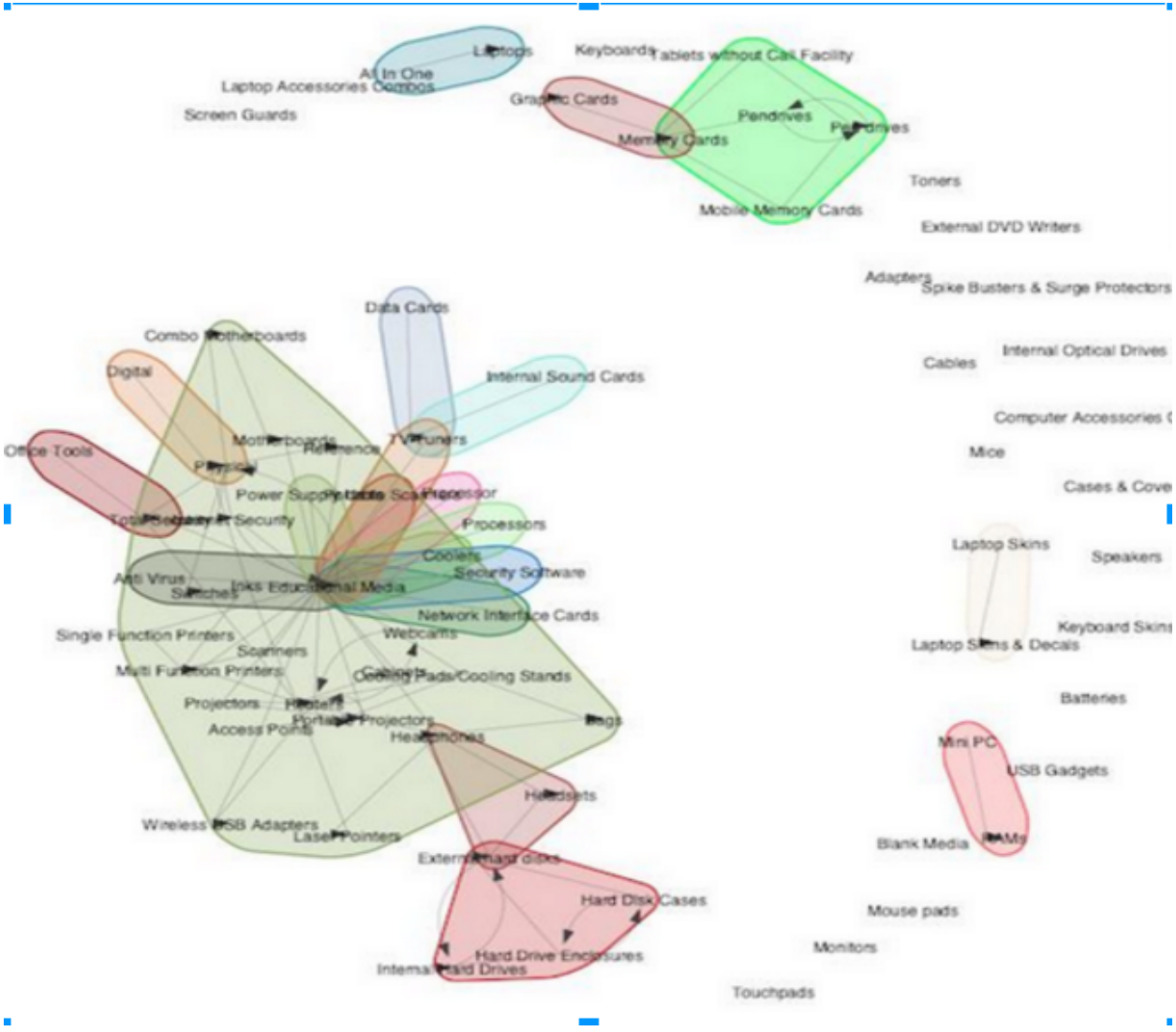
Figure: Graded weighted Bag of Word Vector Classification Approach

Outline

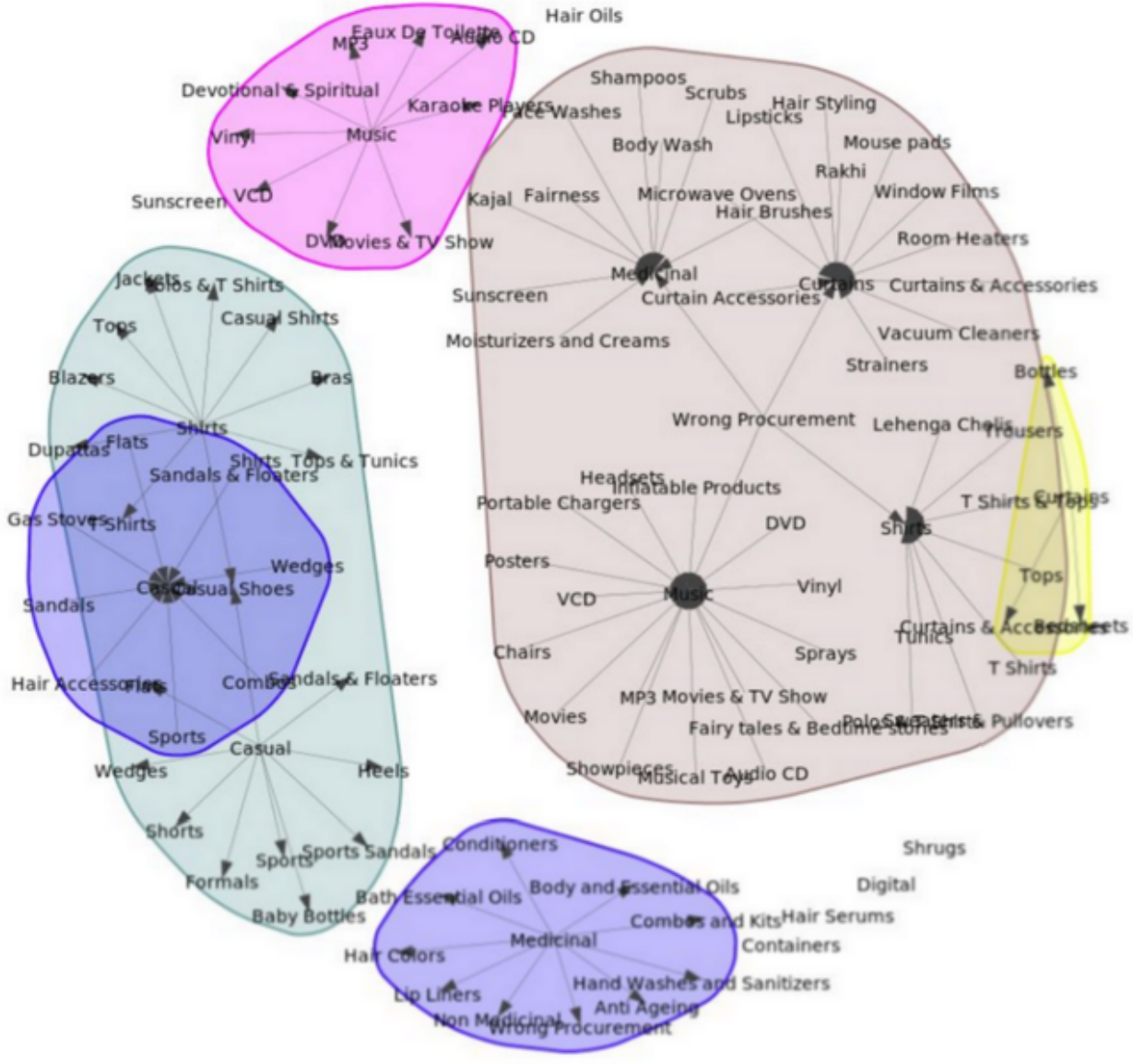
Connected Component Grouping Approach

- Adopted two level classification approach of Shen and Ruvini et al.
- Use misclassification between categories to form connected graph
- Use weakly connected component (BCC) to find connected category.
- Use of weakly connected component instead of strongly connected component improve sensitivity

Connected Groups using tf-idf



Connected Groups using gwBOVW



Outline

Proposed : Ensemble of Multitype Predictors I

Train multiple level one classifiers parallely to predict product paths, labels and depth-wise labels.[3]

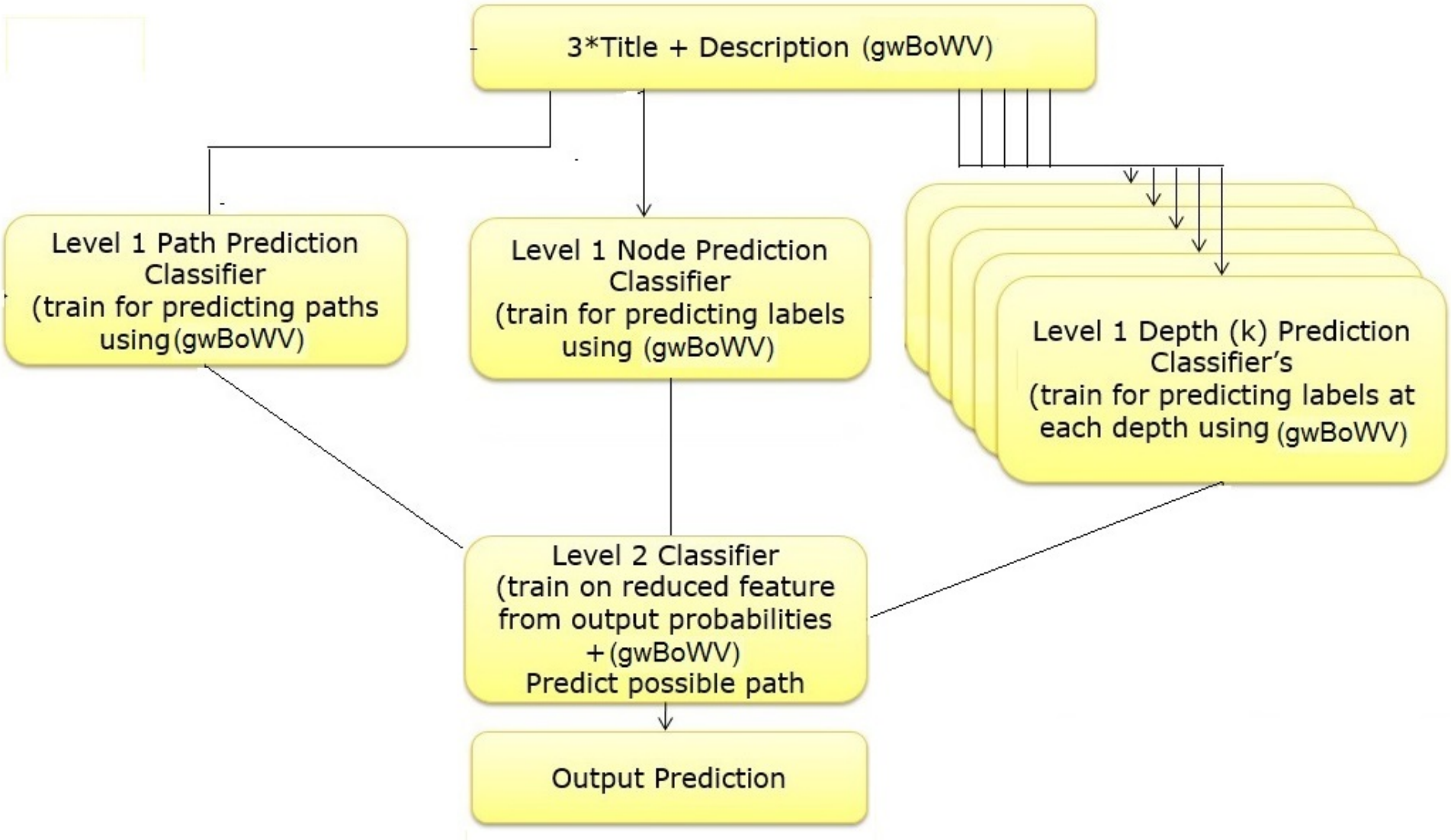
- Path-Wise Classifier takes each possible *path* as a possible class and train a classifier (PP).
- Node-Wise Prediction Classifier takes each possible *node* as a possible class label and train a classifier (NP).
- Depth-Wise Node Prediction Classifiers
  - Train multiple classifiers (DNP_i) one for each depth (i) using nodes that occur at that depth (i) as class labels
  - Data samples which have no node at depth k are train using *dummy* labels (10% sample)

[3] Proc. of COLING'2016

Proposed : Ensemble of Multitype Predictors I

- Concatenate the output probabilities over classes for all level one classifiers.
- Used variance based supervised feature selection to reduce dimension.
- Used final representation for training another classifier at level two to predict path

Proposed : Ensemble of Multitype Predictors



Emperical Study

Goal of this empirical study is to address the following questions :

1. Compare the gwBoVW representation with other embedding representations in literature.
2. Does the proposed multi prediction type classifier ensemble reduce classification error.

Outline

Datasets Description

- Used seller product descriptions and title samples from Flipkart.com
- Flipkart organizes items in two root product categories Non-Books and Books.
- Non-Book data was more discriminative with average description length of around 10-15 words, whereas most book data descriptions range in length between 200 to 300 words.
- Weight title three times the description give more importance to title.
- Used base classifier as random forest which is robust to data skewness.

Data Description I

1. Used Flipkart Book and NonBook dataset
2. Book data have 1.6 million training and 1.1 million testing samples with 3000 paths.
3. Non-Book data have 5 million training and 3.5 million testing samples with 3000 paths

Level	**#Categories**	**%Sample Path**
1	21	34.9%
2	278	22.64%
3	1163	25.7%
4	970	12.9%
5	425	3.85%
6	18	0.10%

Challenges with Datasets I

1. Due to the large and slightly noisy hierarchical taxonomy, the category nodes may not be mutually exclusive.
2. Samples of category paths with different categories at near top (root) and similar categories at leaf nodes i.e. reduplication of the same path with synonymous labels.
3. The quality of title and description varies considerably among seller due to difference in vocabulary.

Challenges with Datasets II

4. Most titles or descriptions are smaller and do not describe the category item fully.
   Title : *Macroman Striped Men's Track Pants*
   The above product can belong to any of the below taxonomy paths :-
   - *Apparels* $\rightarrow$ *Men* $\rightarrow$ *Innerwear and Sleepwear* $\rightarrow$ *Track Pants*
   - *Apparels* $\rightarrow$ *Men* $\rightarrow$ *Sportswear* $\rightarrow$ *Track Pants*
   - *Apparels* $\rightarrow$ *Men* $\rightarrow$ *Sports and Gym Wear* $\rightarrow$ *Track Pants*
5. Also, there were labels like *Others* and *General* at various depths of the taxonomy tree which carry no specific semantic meaning.
6. A special label called *wrong procurement* was removed manually for consistency.

Outline

Non-Book Top 1 Path Predictions

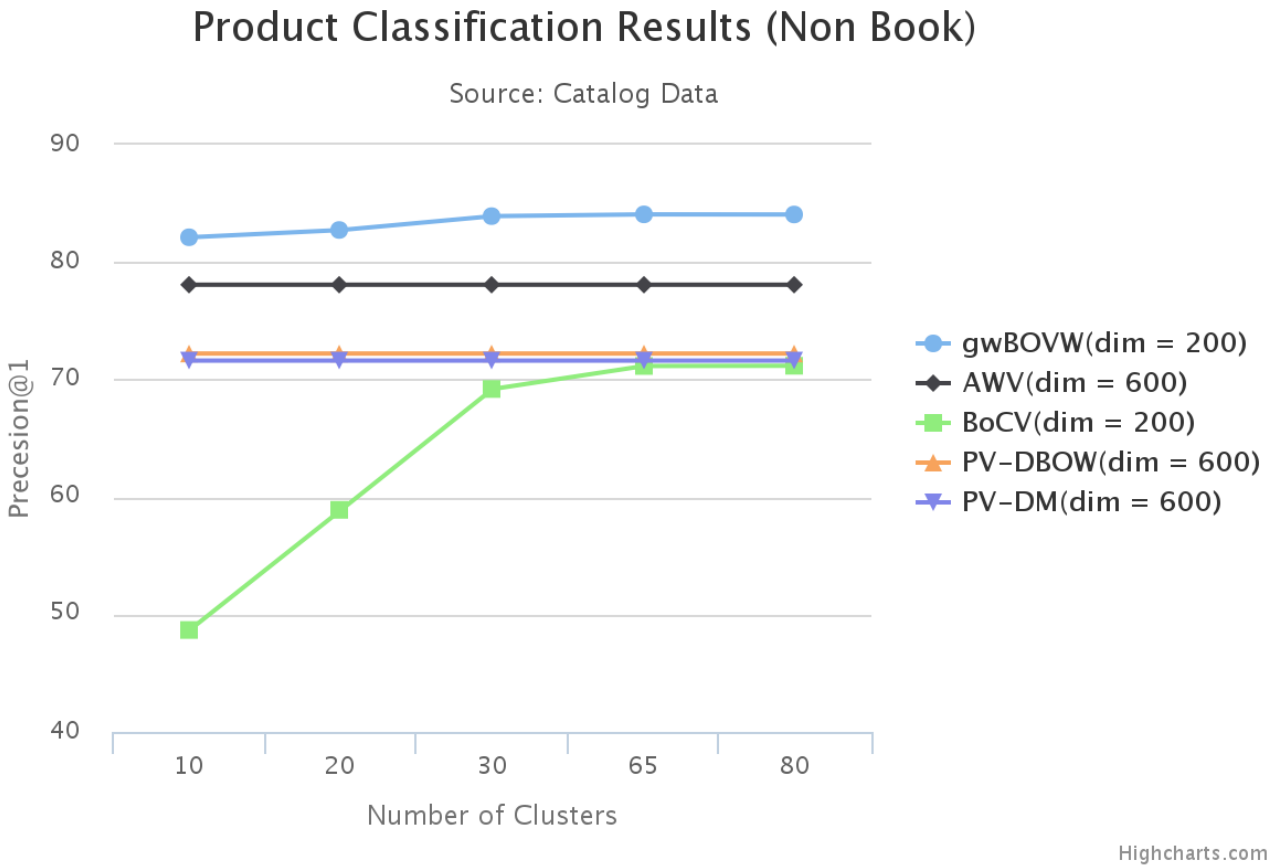


Figure: Comparison of prediction accuracy for path prediction for (gwBoWV) Graded Weighted bag of Words Vector, (BoCV) Bag of Cluster Vector distribution histogram and Average Word Vector with varying Number

Multiple Path Prediction

1. Considering the nature of the data which has multiple similar paths predicting a single path is not appropriate.
2. Solution was to predict more than one path or better a ranked list of 3 or 6 paths with score.

Effect of Vocabulary Dimensions on tf-idf Baseline

$\mathbf{WV}_{dim}$	**%CP**	**%LR**
2000	86.01%	91.94%
4000	**87.63%**	**92.98%**

Table: *Result of top 1 path prediction for multiple tfidf with varying dimension on #Training Samples = 5.0 millions, #Test Samples = 3.5 millions.*

$\mathbf{WV}_{dim}$	**%CP**	**%LR**
2000	94.04%	96.85%
4000	**94.78%**	**97.33%**

Table: *Result of top 6 paths prediction for multiple tfidf with varying dimension on #Training Samples = 5.0 million, #Test Samples = 3.5 million.*

Effect of Vector Dimensions and Size on BOWV I

Table: *Result of top 1 and 6 paths prediction for gwBoWV*

#Clus	**WV**$_{dim}$	**%CP**	**%LR**
40	50	89.45%	94.31%
40	100	**89.91%**	**94.57%**

#Clus	**WV**$_{dim}$	**%CP**	**%LR**
40	50	96.43%	98.27%
40	100	**96.67%**	**98.39%**

Book Dataset Results

1. Books were more confusing with tangled, improper hierarchy and text.
2. Used ensemble of multi-type prediction (path, nodes, depthwise nodes) as described earlier for classifications.
3. Used Mutual Information Criteria (MIC) based (ANOVA i.e. analysis of variance) for dimensionality reduction.

Book Dataset Results

1. We reduced graded weighted bag of words vectors for book dataset.

Red-Dim	**%CP**	**%LR**
3500	75.18 %	87.06 %
4000	**75.24 %**	**87.07 %**
4500	75.18 %	87.01 %

Table: *Results from reduced gwBoVW for Top 6 path prediction (Orignal Dimension : 8080)*

Book Dataset Results I

We obtain improved results for both the evaluation metrics using Label Ensemble Technique

Method	**%CP**	**%LR**
*tfidf	75.00%	86.86%
path	74.83%	86.60%
depth	75.34%	87.08%
node	74.68%	86.65%
comb	**77.26%**	**88.86%**

Table: *Comparision Result from various approaches for Top 6 prediction (*baseline)*

Book Dataset Results II

To improve results further we applied MIC based dimensionality reduction on combined output probabilities and trained another classifier on it (original dimension = 7975)

Red-Dim	**%CP**	**%LR**
2500	79.09%	90.18%
2600	**79.36%**	**90.23%**
2700	79.42%	90.32%

Table: *Results from varying reduced dimension on Top 6 prediction*

Real Example I

Description : *harpercollins continues with its commitment to reissue maurice sendaks most beloved works in hardcover by making available again this 1964 reprinting of an original fairytale by frank r stockton as illustrated by the incomparable maurice sendak in the ancient country of orn there lived an old man who was called the beeman because his whole time was spent in the company of bees one day a junior sorcerer stopped at the hut of the beeman the junior sorcerer told the beeman that he has been transformed if you will find out what you have been transformed from i will see that you are made all right again said the sorcerer could it have been a giant or a powerful prince or some gorgeous being whom the magicians or the fairies wish to punish the beeman sets out to discover his original form. the beeman of orn. the beeman of orn. the beeman of orn.*
Actual Class : books-tree $\rightarrow$ children $\rightarrow$ knowledge and learning $\rightarrow$ animals books $\rightarrow$ reptiles and amphibians

Real Example II

Predictions, **Probability**

books-tree $\rightarrow$ children $\rightarrow$ knowledge and learning $\rightarrow$ animals books $\rightarrow$ reptiles and amphibians , 0.28

books-tree $\rightarrow$ children $\rightarrow$ fun and humor, 0.72

Real Example III

Description : *a new york times science reporter makes a startling new case that religion has an evolutionary basis for the last 50000 years and probably much longer people have practiced religion yet little attention has been given to the question of whether this universal human behavior might have been implanted in human nature in this original and thought provoking work nicholas wade traces how religion grew to be so essential to early societies in their struggle for survival how an instinct for faith became hardwired into human nature and how it provided an impetus for law and government the faith instinct offers an objective and non polemical exploration of humanitys quest for spiritual transcendence. the faith instinct how religion evolved and why it endures. the faith instinct how religion evolved and why it endures. the faith instinct how religion evolved and why it endures*

Actual Class : books-tree $\rightarrow$ academic texts $\rightarrow$ humanities

Real Example IV

Predictions, **Probability**

books-tree $\rightarrow$ academic texts $\rightarrow$ humanities 0.067

books-tree $\rightarrow$ religion and spirituality $\rightarrow$ new age and occult $\rightarrow$ witchcraft and wicca 0.1

books-tree $\rightarrow$ health and fitness $\rightarrow$ diet and nutrition $\rightarrow$ diets 0.1

books-tree $\rightarrow$ dummy 0.4

Conclusions

1. Presented a novel compositional technique to form efficient document vectors.
    1. Capture importance and distinctiveness of words across documents by using graded weighting.
    2. Embedded document vectors in higher dimensional space than original word vectors.
2. Uses an ensemble of multiple type classifiers to decrease classification error on Flipkart E-commerce catalog data-set.


Thank
You!

Algorithm : gwBoWV representation

Algorithm 1: Graded Weighted Bag of Word Vectors

Data: Documents D_n, n $= 1 \ldots$ N

Result: Document vectors $gwBo\vec{V}W_{D_n}$, n $= 1 \ldots$ N

Train SGNS model to obtain word vector representation (wv_n) using all document D_n , n $= 1 \ldots$ N;

Calculate idf values for all words $(idf(w_j))$, j $= 1 \ldots |V|, |V|$ represent vocabulary size;

Use kmean algorithm for ing each word (w_j) using their word-vectors $(w\vec{v}_j)$ in K s;

for *i* $\in$ *(1 ... N)* **do**

    initialize vector $c\vec{v}_k = \vec{0}$, here k $= 1 \ldots$ K;

    initialize frequency $icf_k = 0$, here k $= 1 \ldots$ K;

    while *not at end of document* D_i **do**

        read current word w_j and obtain wordvec $w\vec{v}_j$;

        obtain index $k = idx(w\vec{v}_j)$ for wordvec $w\vec{v}_j$;

        update vector $c\vec{v}_k \mathrel{+}= w\vec{v}_j$;

        update frequency $icf_k \mathrel{+}= idf(w_j)$;

    end

    obtain $Bo\vec{V}W_{D_i} = \bigoplus_{k=1}^{K} c\vec{v}_k$, ,here $\oplus$ represent concatenation;

    obtain $gwBo\vec{V}W_{D_i} = Bo\vec{V}W_{D_i} \bigoplus_{k=1}^{K} icf_k$, here $\oplus$ represent concatenation;

end

Algorithm : Modified Connected Component

Algorithm 2: Modified Connected Component Grouping

Data: Set of Categories $C = \{c_1, c_2, c_3 ... c_n\}$ and threshold α

Result: Set of dense sub-graphs $CG = \{cg_1, cg_2, cg_3, cg_4 \dots cg_m\}$ representing highly connected groups

1 Train a weak classifier H on all possible categories ;

2 Compute pairwise confusion probabilities between classes using values from the confusion matrix *(CM)*.

$$Conf(c_i, c_j) = \begin{cases} CM(c_i, c_j), & \text{if } CM(c_i, c_j) \geq \alpha \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

here, Conf(c_i, c_j) may not be equal to Conf(c_j, c_i) due to non symmetric nature of Confusion Matrix CM;

3 Construct confusion graph $G = (E, V)$ with vertices (V) as confused categories and edges (E_{ij}) from i j with weight = Conf(c_i, c_j).;

4 Apply algorithm for Bi-Connected Component, Strongly Connected Component or Weakly Connected Component on G to obtain a set of dense sub-graphs $CG = \{cg_1, cg_2, cg_3, cg_4 \dots cg_m\}$.

Algorithm : gwBoWV representation

Algorithm 3: Graded Weighted Bag of Word Vectors

Data: Documents D_n, n $= 1 \ldots$ N

Result: Document vectors $gwBo\vec{V}W_{D_n}$, n $= 1 \ldots$ N

Train SGNS model to obtain word vector representation (wv_n) using all document D_n , n $= 1 \ldots$ N;

Calculate idf values for all words $(idf(w_j))$, j $= 1 \ldots |V|, |V|$ represent vocabulary size;

Use kmean algorithm for ing each word (w_j) using their word-vectors $(w\vec{v}_j)$ in K s;

for *i* $\in$ *(1 ... N)* **do**

  initialize vector $c\vec{v}_k = \vec{0}$, here k $= 1 \ldots$ K;

  initialize frequency $icf_k = 0$, here k $= 1 \ldots$ K;

  while *not at end of document* D_i **do**

    read current word w_j and obtain wordvec $w\vec{v}_j$;

    obtain index $k = idx(w\vec{v}_j)$ for wordvec $w\vec{v}_j$;

    update vector $c\vec{v}_k \mathrel{+}= w\vec{v}_j$;

    update frequency $icf_k \mathrel{+}= idf(w_j)$;

  end

  obtain $Bo\vec{V}W_{D_i} = \bigoplus_{k=1}^{K} c\vec{v}_k$, ,here $\oplus$ represent concatenation;

  obtain $gwBo\vec{V}W_{D_i} = Bo\vec{V}W_{D_i} \bigoplus_{k=1}^{K} icf_k$, here $\oplus$ represent concatenation;

end

Direct Node Prediction I

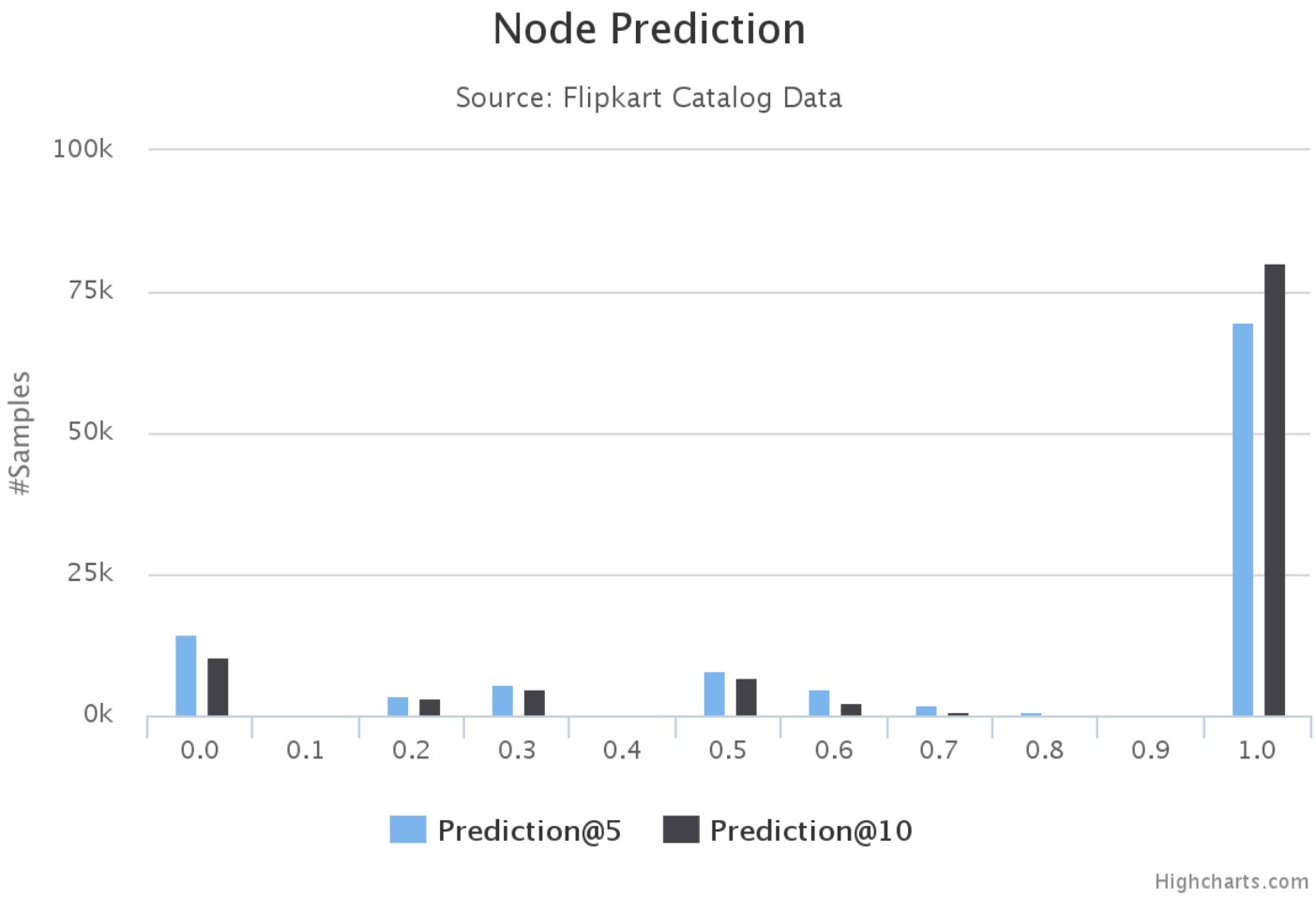


Figure: Distribution of samples with varying fraction of correct path predicted for top 5 and top 10 node prediction

Direct Node Prediction II

For path prediction we applied MIC based dimensionality reduction on output probabilities and then trained another classifier instead of training it directly on output prediction (original dimension = 6910). See Table 7

Red-Dim	**%PP**	**%CP**	**%LR**	**%LC**
1000	46.26%	72.46%	84.84%	24.85%
1500	46.88%	72.77%	84.44%	24.81%
2000	47.05%	72.26%	84.52%	25.05%
2500	47.70%	72.77%	84.58%	24.81%
3000	44.45%	73.84%	85.83%	23.74%
4000	42.05%	74.80%	86.83%	22.74%
6000	41.63%	74.82%	86.73%	22.41%

Table: *Results on varying reduced dimension on node level 2 prediction for Top 6 prediction*

Depth Wise Classification results

Depth	#Train	#Test	#Labels	Cov@1	Cov@2	Cov@3
1	1.5	1.1	21	65.77%	82.25%	87.48%
2	1.4	1.0	278	51.74%	69.40%	74.47%
3	1.2	0.8	970	58.53%	74.46%	78.18%
4	1.0	0.5	1163	77.86%	87.41%	88.72%
5	0.7	0.4	425	91.12%	94.33%	94.71%
6	0.5	0.3	18	99.74%	99.85%	99.85%

Table: *Depth Wise Prediction using multiple classifier for each depth to predict top 1, top 2 and top 3 labels. Cov@k denote accuracy for correct label prediction in top k predicted label.*

Depth Wise Level 2 Prediction

1. To predict path we train another classifier which used concatenation of output probabilities of 1^{st} classifiers at each depth as training feature vectors for 2^{nd} classifiers with path as probable classes.
2. To improve results we applied MIC based dimensionality reduction on combined output probabilities and then trained another classifier instead of training it directly. We obtained best results on a reduced dimension of 2600 (original dimension = 7975). See Tables 9

Red-Dim	**%PP**	**%CP**	**%LR**	**%LC**
2500	46.42%	79.09%	90.18%	25.50%
2600	46.10%	**79.36%**	**90.23%**	25.38%
2700	45.48%	79.42%	90.32%	25.18%
3000	43.98%	78.18%	89.44%	24.29%
4000	42.31%	76.59%	88.17%	23.20%

Table: *Results from varying reduced dimension on depth level 2 prediction for Top 6 prediction*

Algorithm : Training Two Level Ensemble

Algorithm 4: Two Level Ensemble Training

Data: Catalog Hierarchical Tree (T) of depth K and training data $D =$ (d,p_d) where d is product description and p_d is taxonomy path

Result: Set of level one Classifiers C $= \{PP, NP, DNP_1 \ldots DNP_K\}$ and level two classifier FPP.

1 Obtain $gw\vec{BoVW}_d$ features for each product description d ;

2 Train Path-Wise Prediction Classifier (PP) with possible classes as product taxonomy paths (p_d);

3 Train Node-Wise Prediction Classifier (NP) with possible classes as nodes in the taxonomy tree i.e. (n_d). Here each description will be tagged by multiple node labels appearing in the path.

4 **for** *k* $\in$ *(1 ... K)* **do**

5 Train Depth-Wise Node Classifier for depth k (DNP_K) with class labels as nodes that are possible at depth k i.e. (n_k)

6 **end**

7 Obtain output probabilities $\vec{P_X}$ over all classes for each level one classifier X i.e. $\vec{P_{PP}}$, $\vec{P_{NP}}$ and $\vec{P_{DNP_k}}$, here k $= 1 \ldots$ K.;

8 Obtain feature vector $\vec{FV}_d$ for each description as:

$$\vec{FV}_d = gw\vec{BoWV}_d \bigoplus \vec{P_{PP}} \bigoplus \vec{P_{NP}} \bigoplus_{i=k}^{K} \vec{P_{DNP_k}} \tag{2}$$

Here $\bigoplus$ represents the concatenation operation.;

9 Reduce feature dimension of ($\vec{RFV}_d$) using suitable supervised feature selection technique based on mutual information criteria;

10 Train Final Path-Wise Prediction Classifier ($\vec{FPP}_d$) using RFV_d as the feature vector and possible product taxonomy paths (p_d) as class labels.

Testing Two Level Ensemble

Algorithm 5: Two Level Ensemble Testing

Data: Catalog Hierarchical Tree (T) of depth K and testing data $D =$ (d,p_d) where d is the product description, p_d is the path in the taxonomy tree. Set of level one Classifiers C $= \{PP, NP, DNP_1 \ldots DNP_K\}$ and final level two classifier FPP

Result: top m prediction paths P_{d_i} for description d, here i $= 1 \ldots$ m

1. Obtain $gw\vec{BoVW}_d$ features for each product description d in test data;
2. Get Prediction Probabilities from all level one classifiers to obtain combined feature vector $(\vec{FV}_d)$ using Equation 2;
3. Use dimension reduction $(\vec{RFV}_d)$to obtain reduced feature vector from description d using original feature vector $(\vec{FV}_d)$;
4. Output top m paths from final prediction using output probabilities from the level 2 classifier FFP for description d.

Proposed Formulation : gwBoWV

- Cluster pre-trained word vectors using a suitable clustering algorithm.
- Add word-vectors of words from document d_i in cluster c_j to form cluster vectors cv_{ij}.
- Concatenate cluster vectors cv_{ij} for all clusters to form document vector v_{d_i}.
- Add idf value of words from document d_i belonging to cluster c_j to obtain inverse cluster frequency icf_{ij}.
- Concatenate icf_{ij} to v_{d_i} to get final document vector.

Example : gwBoWV I

1. Lets assume there are four s $C = [C_1, C_2, C_3, C_4]$, here C_i represents the i^{th}
2. Let $D_n = [w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}]$ represent document consisting of words w_1,w_2 ... w_{10} in order, whose document vectors need to be composed using word vectors $w\vec{v}_1$,$w\vec{v}_2$ ... $w\vec{v}_{10}$ respectively. Lets assume following word- assignment for document D_n

Word	
w_4, w_3, w_{10}, w_5	C_1
w_9	C_2
w_1, w_6, w_2	C_3
w_8, w_7	C_4

Table: *Document Word Assignment*

Example : gwBoWV II

3. Obtained C_i contribution in document D_n by summation of word vectors for words coming from document D_n and C_i as describe :-
   - $c\vec{v}_1 = w\vec{v}_4 + w\vec{v}_3 + w\vec{v}_{10} + w\vec{v}_5$
   - $c\vec{v}_2 = w\vec{v}_9$
   - $c\vec{v}_3 = w\vec{v}_1 + w\vec{v}_6 + w\vec{v}_2$
   - $c\vec{v}_4 = w\vec{v}_8 + w\vec{v}_7$

   Similarly we also calculate idf values of each C_i for document D_n :-
   - $icf_1 = \text{idf}(w_4) + \text{idf}(w_3) + \text{idf}(w_{10}) + \text{idf}(w_5)$
   - $icf_2 = \text{idf}(w_9)$
   - $icf_3 = \text{idf}(w_1) + \text{idf}(w_6) + \text{idf}(w_2)$
   - $icf_4 = \text{idf}(w_8) + \text{idf}(w_7)$
4. Concatenate vectors to form Bag of Word Vector of dimension (# X $\#wordvec$ dimensions) as describe

$$\vec{BoWV}(D_n) = c\vec{v}_1 \oplus c\vec{v}_2 \oplus c\vec{v}_3 \oplus c\vec{v}_4 \tag{3}$$

Example : gwBoWV III

5. Concatenate word- idf values to form graded weighted Bag of Word Vector of dimension (# x $\#wordvec$ + # dimensions) as describe

$$gwBo\vec{W}V(D_n) = c\vec{v}_1 \oplus c\vec{v}_2 \oplus c\vec{v}_3 \oplus c\vec{f}_4 \oplus icf_1 \oplus icf_2 \oplus icf_3 \oplus icf_4. \quad (4)$$