

A ANALYTICS FRAMEWORK FOR AUTO MODELLING METHODOLOGY TO COMPOSE TOPIC

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering degree in

Computer Science and Engineering

By

CHINTHAKUNTLA TEJA REDDY(Reg.No.37110154)

CHAVVA TRINADH SAI BHARGAV(Reg.No.37110131)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED TO BE UNIVERSITY) Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119**

MARCH - 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **CHAVVATRINADH SAI BHARGAV(Reg. No. 37110131)** and **CHINTHAKUNTLA TEJA REDDY(Reg. No.37110154)** who carried out the project entitled "**A Analytics Framework For Auto-Modelling Methodology To Compose Topic**" under my supervision from **August 2021 to March 2022.**

Internal Guide

Dr. JEBERSON RETNA RAJ, M.Tech.,P.h.D

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **CHINTHAKUNTLA TEJA REDDY** hereby declare that the Project Report entitled “**A Analytics Framework For Auto-Modelling Methodology To Compose Topic**” is done by me under the guidance of **DR.JEBERSON RETNA RAJ, M.Tech.,P.h.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **DR.JEBERSON RETNA RAJ, M.Tech., P.h.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Topic models simultaneously model the latent topic structure of large collections of documents and a response variable associated with each document. It is really popular to detect hot topics, which can benefit many tasks including topic recommendations, the guidance of public opinions, and so on. However, in some cases, people may want to know when to re-hot a topic, i.e., make the topic popular again. Also, it considers the continuous temporal modeling of topics, since topics are changing continuously over time. Furthermore, a weighting scheme is proposed to smooth the fluctuations in topic re-hotting prediction. Topic models simultaneously model the latent topic structure of large collections of documents and a response variable associated with each document. It is really popular to detect hot topics, which can benefit many tasks including topic recommendations, the guidance of public opinions, and so on. However, in some cases, people may want to know when to re-hot a topic, i.e., make the topic popular again. Also, it considers the continuous temporal modeling of topics, since topics are changing continuously over time. Furthermore, a weighting scheme is proposed to smooth the fluctuations in topic re-hotting prediction. model uses of topic models, which predominantly fall into the categories of understanding topics, understanding similarity, and understanding change.

Table of contents

Chapter No.	Title	Page No.
	ABSTRACT	v
	LIST OF FIGURES	vii
1.	INTRODUCTION	1
	1.1 Overview	1
	1.2 Scope of the Project	1
	1.3 Domain Overview	1
	1.3.1 Machine Learning	1
2.	LITERATURE SURVEY	6
	2.1 Literature Survey 1	6
	2.2 Literature Survey 2	7
	2.3 Literature Survey 3	8
	2.4 Literature Survey 4	9
3.	SYSTEM ANALYSIS	11
	3.1 Existing System	11
	3.2 Drawbacks of Existing System	11
	3.3 Proposed System	11
	3.4 Architecture Diagram	12
	3.5 Advantages of Proposed System	12
	3.6 System Requirement	13
	3.7 Software Description	13
4.	SYSTEM IMPLEMENTATION	14
	4.1 List of Modules	14
	4.1.1 Module 1 : Data Evaluation	14
	4.1.2 Module 2 : n-grams and correlations	15
	4.1.3 Module 3 : Topic Modelling	15
	4.1.4 Module 4 : Prediction	16

E	5.	RESULT AND DISCUSSION	17
		5.1 Result	18
		5.2 Discussion	18
	6.	SOFTWARE TESTING	19
		6.1 General	20
		6.2 Test Driven Development	20
		6.3 Uint Testing	21
		6.4 BlackBox Testing	22
		6.5 Intergration Testing	22
		6.6 System Testing	23
		6.7 Sanity Testing	24
		6.8 Regression Testing	25
	7.	CONCLUSION AND FEATURE ENHANCEMENT	27
		REFERENCES	28
		APPENDIX	30

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
1.1	Supervised Architecture	3
1.2	Block Diagram	4
3.1	System Architecture	12
5.1	Precision of topic classification	17
5.2	The error recognition cost of topic classification.	17
5.3	F1 of topic classification.	18
5.4	Recall of topic classification.	18
7.1	Integration Testing	23
7.2	Module Testing	23

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

LDA(Latent Dirichlet Allocation) is a probabilistic model for modeling text data, which can model the topic information of text data. The LDAtopic model can realize the dimensionality reduction representation of text in the semantic space, and it models the text with the probability of vocabulary, which alleviates the problem of data sparsity to some extent. Therefore, in this paper, the LDA topic model is used to extract the topic of the text.

1.2 SCOPE OF THE PROJECT

The main contributions of this project therefore are:

- Data Analysis
- Dataset Preprocessing
- Training the Model
- Testing of Dataset

1.3 DOMAIN OVERVIEW

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future. Data is the core backbone of machine learning algorithms. With the help of the historical data, we are able to create more data by training these machine learning algorithms. For example, Generative Adversarial Networks are an advanced concept of Machine Learning that learns from the historical images through which they are capable of generating more images. This is also applied towards speech and text synthesis. Therefore, Machine Learning has opened up a vast potential for data science applications.

1.3.1 MACHINE LEARNING

Machine Learning combines computer science, mathematics, and statistics. Statistics is essential for drawing inferences from the data. Mathematics is useful for developing machine learning models and finally, computer science is used for

implementing algorithms. However, simply building models is not enough. You must also optimize and tune the model appropriately so that it provides you with accurate

results. Optimization techniques involve tuning the hyperparameters to reach an optimum result.

The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways. Machine Learning has revolutionized industries like medicine, healthcare, manufacturing, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry. Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms.

With an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it. These machine learning algorithms use the patterns contained in the training data to perform classification and future predictions.

Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions. Based on the final accuracy, one can optimize their models using various standardized approaches. In this way, Machine Learning model learns to adapt to new examples and produce better results.

Types of Machine Learning

Machine Learning Algorithms can be classified into 3 types as follows –

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

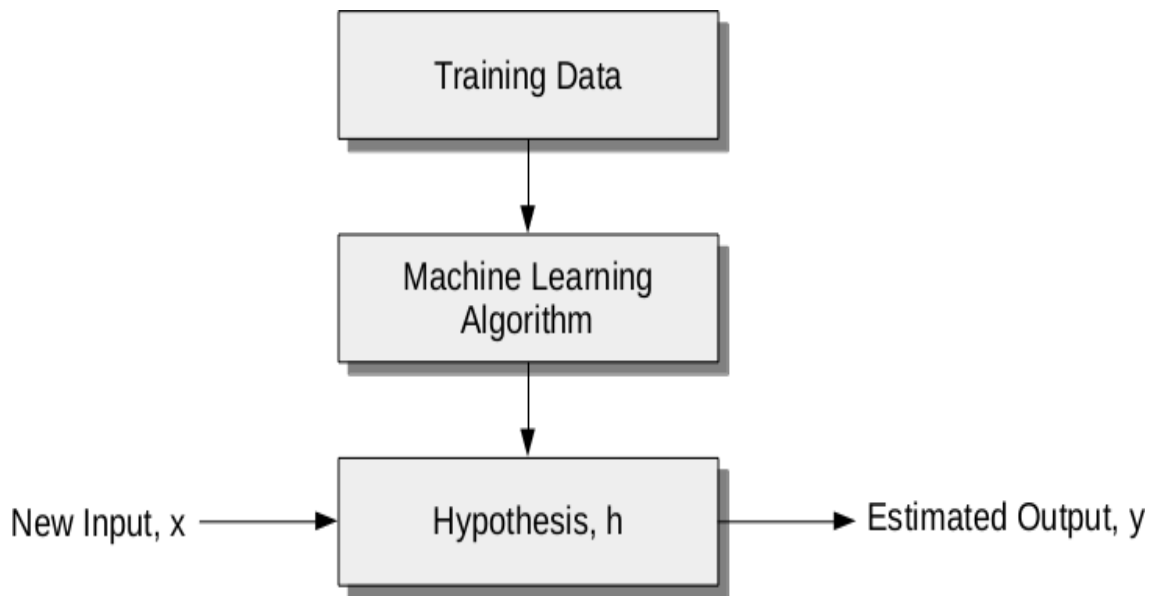


Fig 1.1 Supervised Architecture

SUPERVISED LEARNING

In the majority of supervised learning applications, the ultimate goal is to develop a finely tuned predictor function $h(x)$ (sometimes called the “hypothesis”). “Learning” consists of using sophisticated mathematical algorithms to optimize this function so that, given input data x about a certain domain (say, square footage of a house), it will accurately predict some interesting value $h(x)$ (say, market price for said house).

$$h(x_1, x_2, x_3, x_4) = \theta_0 + \theta_1 x_1 + \theta_2 x_3^2 + \theta_3 x_3 x_4 + \theta_4 x_1^3 x_2^2 + \theta_5 x_2 x_3^4 x_4^2$$

This function takes input in four dimensions and has a variety of polynomial terms. Deriving a normal equation for this function is a significant challenge. Many modern machine learning problems take thousands or even millions of dimensions of data to build predictions using hundreds of coefficients. Predicting how an organism’s genome will be expressed, or what the climate will be like in fifty years, are examples of such complex problems.

Under supervised ML, two major subcategories are:

- Regression machine learning systems: Systems where the value being predicted falls somewhere on a continuous spectrum. These systems help us with questions of “How much?” or “How many?”.
- Classification machine learning systems: Systems where we seek a yes-or-no prediction, such as “Is this tumor cancerous?”, “Does this cookie meet our quality standards?”, and so on.

In practice, x almost always represents multiple data points. So, for example, a housing price predictor might take not only square-footage (x_1) but also number of bedrooms (x_2), number of bathrooms (x_3), number of floors (x_4), year built (x_5), zip code (x_6), and so forth. Determining which inputs to use is an important part of ML design. However, for the sake of explanation, it is easiest to assume a single input value is used.

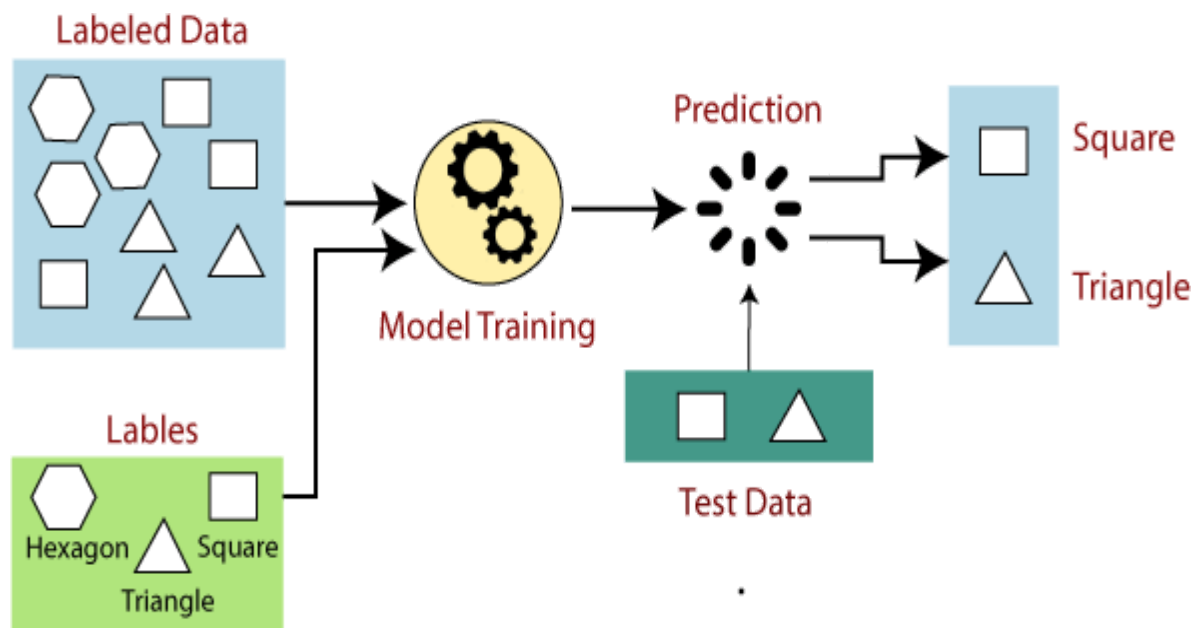


Fig 1.2 Block Diagram

Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set.

REGRESSION

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc.

1. Linear Regression
2. Regression Trees
3. Non-Linear Regression
4. Bayesian Linear Regression
5. Polynomial Regression

CLASSIFICATION

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

Spam Filtering,

- Random Forest
- Decision Tree
- Logistic Regression
- Support vector Machines

CHAPTER 2

LITERATURE SURVEY

2.1

Literature Survey 1

Title	When to Make a Topic Popular Again? A Temporal Model for Topic Re-hotting Prediction in Online Social Networks
Authors	Chaokun Wang, Member, IEEE, Xin Xin, and Jingwen Shang
Published Year	2018
Efficiency	✓ Topic re-hotting prediction ✓ Continuous modelling strategy ✓ Predict their popularity simultaneously
Drawbacks	✗ Very difficult to precisely obtain opportune time ✗ Cannot reduce the influence of topic fluctuation ✗ Cannot predicts the time points
Description	In E-Step, the maximum likelihood can be computed by the estimated value of latent parameters. In M-Step, the parameters are then re-estimated by the maximum likelihood which is got in the E- Step. The algorithm iteratively proceeds E-Step and M-Step until convergence. The EM algorithm is first applied in statistical areas and then broadly used in almost all fields where statistical techniques have been applied. The users in the same friend-circle tend to have similar interests while the users in different friend circles

might have different interests. For example, a user may be interested in a friend-circle in a certain period of time, and this friend-circle is always involved in some topics, like “Grammy Award”, “Country music”, and “Billboard Music Awards”. It means this user likes the songs and her/his friend-circle is the circle of “American songs” in this period of time. In our model, the friend-circle is denoted by another latent variable which represents hidden subsets of the user’s followers.

2.2

Literature Survey 2

Title	A Deep and Autoregressive Approach for Topic Modeling of Multimodal Data
Authors	Yin Zheng, Yu-Jin Zhang, Senior Member, and Hugo Larochelle
Published Year	2015
Efficiency	↓ Cannot deal with the inbalance between the number of visual words and annotation words ↓ High Computational cost ↓ Cannot helps fighting against overfitting
Drawbacks	✓ Reduces its capability of modelling multimodal data ✓ Compute the probability of paths that reach a leaf corresponding to an annotation word ✓ Joint indexing of all visual and annotation words and use a larger binary word tree
Description	Task-driven comparison must address the wide variety of tasks for which topic models are used. These range from full-corpus summary, to comparisons between documents or groups of documents, to investigations of evolution over time. The first term is a purely discriminative term, while the second is unsupervised and can be understood as a regularizer, that encourages a solution which also explains the unsupervised

statistical structure within the visual words. In practice, this regularizer can bias the solution too strongly away from a more discriminative solution that generalizes well. Hence, As in regular DocNADE, computation of the training objective and its gradient requires that we define an ordering of the visual words. Though we could have defined an arbitrary path across the image to order the words (e.g. from left to right, top to bottom in the image), we follow Larochelle and Lauly and randomly permute the words before every stochastic gradient update.

2.3

Literature Survey 3

Title	Progressive Learning of Topic Modeling Parameters: A Visual Analytics Framework
Authors	Mennatallah El-Assady, Rita Sevastjanova, Fabian Sperrle, Daniel Keim, and Christopher Collins
Published Year	2018
Efficiency	↓ Cosine similarity and inverse document frequency are not considered ↓ Not well supported by topic modelling toolkits ↓ Cannot boost the impact of all proper nouns
Drawbacks	✓ Better understanding of the topic descriptors ✓ Adapts the topic models through document relevance feedback ✓ Independent of specific topic modelling algorithms

Description

The core component of our visual analytics technique is the visual analysis workspace. This is the interface in which users interact with the processed data and topic modeling output. We designed the visual interface as a dynamic workspace with consistent visual encoding to facilitate performing the mentally challenging exercise of comparing the different models and their document distributions. One central design consideration for both the usability and aesthetic appeal of the workspace is to use a visual linkage between the different shown components. For example, we always place the two topic models on the two sides of the screen, referring to them as the left and right model. This visualization relies on the output of an automatic topic matching approach that identifies three levels of topic matches: complete matches, similarity-only matches, and mismatches.

the underlying document overlap. Hereby the descriptor similarity between two topic descriptor vectors is computed using the Ranked and Weighted Penalty Distance function (RWPD) that introduces a weighted penalty for every keyword that is present in only one of the two descriptor vectors.

To activate the functionalities for the relevance feedback, the minimized close-reading panel is transitioned from the bottom of the workspace to the center, preserving the visual linkage of the anchored topics. When extended, this view is accompanied by a decision-slider between the two anchored topic-label boxes, along with a horizontal document bar chart which is sorted and colored according to the selected global sorting measure. In the center of this visualization is the currently selected document, with its keywords highlighted.

2.4

Literature Survey 4

Title	Learning Methods for Dynamic Topic Modelling in Automated Behavior Analysis
Authors	Qiqi Zhu, Yanfei Zhong, Siqi Wu, Liangpei Zhang, Deren Li
Published Year	2017
Efficiency	♫ High Implementation cost ♫ Cannot be scalable ♫ Impossible to distinguish two topics or two
Drawbacks	✓ Each topic is assumed to be a distribution over words ✓ Provide more accurate results than the scheme ✓ A document is fully described by a corresponding behaviour
Description	This paper proposes new learning algorithms for activity analysis in video. Model. Two novel learning algorithms based on the expectation maximization approach and variational Bayes inference are proposed. Theoretical derivations of the posterior estimates of model parameters are given. The designed learning algorithms are compared with the Gibbs sampling inference scheme introduced earlier in the literature. A detailed comparison of the learning algorithms is presented on real video data. We also propose an anomaly localization procedure, elegantly embedded in the topic modeling framework. It is shown that the developed learning algorithms can achieve 95% success rate. The proposed framework can be applied to a number of areas, including transportation systems, security, and surveillance. it is essential to detect anomalies as soon as possible. Hence

an approximation of the integral in (34) is used for efficient computation. The first approximation is based on the assumption that the training data set is representative for parameter learning, which means that the posterior probability of the parameters would not change if there is more observed.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Topic tracking can help people to explore the process of topic development from the huge and complex network texts information. By effectively organizing large-scale news documents, a method for the evolution of news topics over time is proposed in this paper to realize the tracking and evolution of topics in the news text set. First, the LDA (latent Dirichlet allocation) model is used to extract topics from news texts and the Gibbs Sampling method is used to speculate parameters. The topic mining using the K-means method is compared to highlight the advantages of using LDA for topic discovery. Second, the improved single-pass algorithm is used to track news topics. The JS (Jensen Shannon) divergence is used to measure the topic similarity, and the time decay function is introduced to improve the similarity

between topics with the similar time. Finally, the strength of the news topic and the content change of the topic in different time windows are analyzed.

In the existing system, the topic tracking method based on discrete time interval is proposed, and the time decay function is introduced to calculate the similarity between the documents. At the same time, the influence of the input order on the tracking results is also solved. The topic tracking is carried out by using the improved Single-Pass clustering method.

3.2 DRAWBACKS OF EXISTING SYSTEM

- ✓ Takes extremely longer time to reach the stationary distribution
- ✓ It does not produce an exact topic.
- ✓ Difficulties to obtain better performance
- ✓ Inaccurate estimations of the missing pixels
- ✓ High prediction complexity for large datasets

3.3 PROPOSED SYSTEM

In the proposed work, our predominant concern when considering topics across two models is topic alignment: which topics best match with one another, and how. Matching can mean either sharing the same documents or the same words. This comparison is important for tracking consistency and correspondence across models. In two closely aligned models, we would expect to see a one-to-one matching, but this is not always the case, either because of parameter differences or random variations.

The discrete modeling strategy divides the whole time domain into contiguous non- overlapping time windows, and then uses the trained data (depicted as blue broken lines) to predict whether the topic will re-hot in the next time window.

The data preprocessing filters out the tweets that do not conform to our requirements as follows. Firstly, the tweets which are irrelevant to any topics, such as tweets to express personal emotions, are removed. Besides, the tweets which are not related to certain topics that can be hot at least once are filtered out. Finally, we choose the tweets corresponding to the topics that can be re-hot at least once above-mentioned data preprocessing, the tweets that meet our requirements are sparser than the initial data sets.

3.4 ARCHITECTURE DIAGRAM

There are many kinds of architecture diagrams, like a software architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram, etc.

For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support.

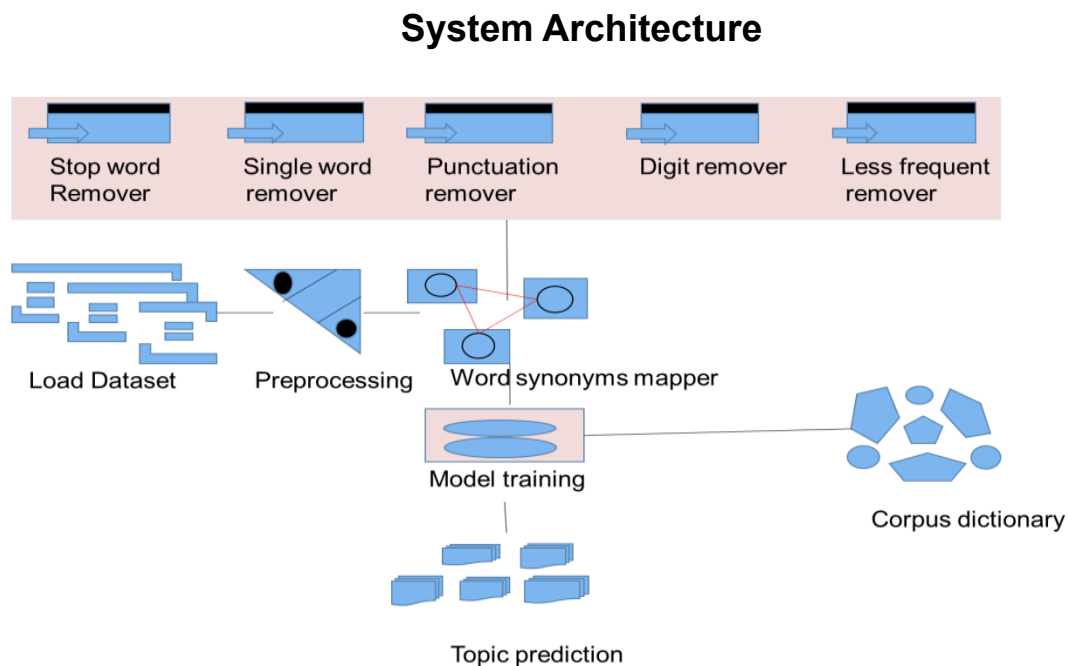


Fig 3.1 Architecture

3.5 ADVANTAGES OF PROPOSED SYSTEM

- ✓ Enable Topic Modelling naturally
- ✓ Reduce Costs and Inefficiencies
- ✓ Benefit from Market Research and Analysis
- ✓ Ability to Deliver High Quality Results
- ✓ Automatically detects the important features
- ✓ Constantly evolves with new information

3.6 SYSTEM REQUIREMENT

HARDWARE SPECIFICATION

- i. Hardware Requirement
- ii. Hardware Minimum Requirement
- iii. Disk Space 32 GB or more, 10 GB

iv. Memory 512 MB

SOFTWARE SPECIFICATION

- ✓ Python
- ✓ Numpy
- ✓ Sci-learn
- ✓ Eclipse IDE

3.7 SOFTWARE DESCRIPTION

Python is a free, open-source programming language. Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these operating systems. Python is also a great visualization tool. It provides libraries such as Matplotlib, seaborn and bokeh to create stunning visualizations.

In addition, Python is the most popular language for machine learning and deep learning. As a matter of fact, today, all top organizations are investing in Python to implement machine learning in the back-end.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 LIST OF MODULES

Module 1 : Data Evaluation

Module 2 : n-grams and correlations

Module 3 : Topic Modelling

4.1.1.Module 1 : Data Evaluation

Exploratory data analysis relies heavily on visualizations and graphical interpretations of data. While statistical modeling provides a "simple" low-dimensional representation of relationships between variables, they generally require advanced knowledge of statistical techniques and mathematical principles. Visualizations and graphs are typically much more interpretable and easy to generate, so you can rapidly explore many different aspects of a dataset. The ultimate goal is to generate simple summaries of the data that inform your question(s). It is not the final stop in the data science pipeline, but still an important one.

Characteristics of exploratory graphs

Graphs generated through EDA are distinct from final graphs. You will typically generate dozens, if not hundreds, of exploratory graphs in the course of analyzing a dataset. Of these graphs, you may end up publishing one or two in a final format. One purpose of EDA is to develop a personal understanding of the data, so all your code and graphs should be geared towards that purpose. Important details that you might add if you were to publish a graph² are not necessary in an exploratory graph.

Exploratory data analysis is a data analysis methods to analysis data and find the inherent law based on the actual distribution of data. Exploratory Data Analysis(EDA) using visual methods to discover the structure contained in the data. Visual data analysis methods in use in a wide range can be traced back to many centuries ago, it is because that human eyes and brains possess strong structural ability to detect that occupy such important position in data exploring. And visual analysis is to play a variety of human models in the processing capacity of the special way to display data.

Analysts always do Exploratory Data Analysis for data first, then are certain to selected the mode of structure quantity or stochastic quantity; Exploratory Data Analysis also can show the unexpected deviation which the common model can not. The key point of Exploratory Data Analysis is not only flexible apply to the data structure but also flexible reaction to the revealed mode of the later analysis- step.

4.1.2 MODULE 2 N-GRAMS AND CORRELATIONS

Data correlation is the way in which one set of data may correspond to another set. In ML, think of how your features correspond with your output.

For example, the image below visualizes a dataset of brain size versus body size. Notice that as the body size increases, so does the brain size. This is known as a linear correlation. In a simple definition of linear correlation, the data follows a straight line. Not all data is linearly correlated. The following image shows a curve of ice cream sales versus temperature. It has an inverted U shaped graph. Why?

It could mean that if it's hot enough, people might not want to leave their homes to go buy ice cream. Or there could be another reason. So using Linear Regression doesn't make sense for this dataset.

Pearson's Correlation Coefficient helps you find out the relationship between two quantities. It gives you the measure of the strength of association between two variables. The value of Pearson's Correlation Coefficient can be between -1 to +1.

1 means that they are highly correlated and 0 means no correlation. -1 means that there is a negative correlation. Think of it as an inverse proportion.

4.1.3 MODULE 3 : TOPIC MODELLING

Topic modeling is an unsupervised machine learning technique that's capable of scanning a set of documents, detecting word and phrase patterns within them, and automatically clustering word groups and similar expressions that best characterize a set of documents.

Since topic modeling doesn't require training, it's a quick and easy way to start analyzing your data. However, you can't guarantee you'll receive accurate results, which is why many businesses opt to invest time training a topic classification model. Since topic classification models require training, they're known as 'supervised' machine learning techniques. What does that mean? Well, as opposed to text modeling, topic classification needs to know the topics of a set of texts before analyzing them. Using these topics, data is tagged manually so that a topic classifier can learn and later make predictions by itself.

For example, we could imagine a two-topic model of American news, with one topic for "politics" and one for "entertainment." The most common words in the politics topic might be "President", "Congress", and "government", while the entertainment topic may be made up of words such as "movies", "television", and

“actor”. Importantly, words can be shared between topics; a word like “budget” might appear in both equally.

4.1.4 MODULE 4 : PREDICTION

Once training is complete, it's time to see if the model is any good, using Evaluation. This is where that dataset that we set aside earlier comes into play. Evaluation allows us to test our model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen. This is meant to be representative of how the model might perform in the real world. A good rule of thumb I use for a training-evaluation split somewhere on the order of 80/20 or 70/30. Much of this depends on the size of the original source dataset. If you have a lot of data, perhaps you don't need as big of a fraction for the evaluation dataset.

Once you've done evaluation, it's possible that you want to see if you can further improve your training in any way. We can do this by tuning our parameters. There were a few parameters we implicitly assumed when we did our training, and now is a good time to go back and test those assumptions and try other values. A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively. Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The feature importance is clear and relations can be viewed easily. This methodology is more commonly known as learning decision tree from data and above tree is called Classification tree as the target is to classify passenger as survived or died.

CHAPTER 5

RESULT AND DISCUSSION

5.1 EXPERIMENTAL RESULTS

In the experiments, the collected network news data is utilized for the analysis of topic tracking. The news corpus used in the experiments comes from Sina News Network, Tencent News Network, Netease News Network, People's Network, Xinhua Net, China News Network. From the captured news corpus, the news of February, 2018 is selected as experimental data. The number of news is 123,569. In the following experiments, the LDA model is adopted to extract news text topics and the best number of the topics per day is verified and determined by multiple experiments. Then the improved Single-pass algorithm is used for topic tracking. Finally, the strength and the content evolution of the topic tracking are analyzed.

Out[28]:

	headlines	text
0	upGrad learner switches to career in ML & AI w...	Saurav Kant, an alumnus of upGrad and IIIT-B's...
1	Delhi techie wins free food from Swiggy for on...	Kunal Shah's credit card bill payment platform...
2	New Zealand end Rohit Sharma-led India's 12-ma...	New Zealand defeated India by 8 wickets in the...
3	Aegon life iTerm insurance plan helps customer...	With Aegon Life iTerm Insurance plan, customer...
4	Have known Hirani for yrs, what if MeToo claim...	Speaking about the sexual harassment allegatio...
...	...	...
2995	RCom deposits ₹131 crore as part payment ...	Anil Ambani-led Reliance Communications (RCom)...
2996	It's hypocrisy: Angad on social media criticis...	Actor Angad Bedi, while speaking about people ...
2997	Was offered Sanju, didn't do it as couldn't pa...	Akshaye Khanna, who was offered the role of Su...
2998	Salman Khan passed exams with help of leaked p...	Actor Salman Khan's father Salim Khan revealed...
2999	Congress rejected surgical strike idea after 2...	Actor and BJP leader Paresh Rawal has said, "T...

Dataset	Method	Precision	recall	F1	Pmiss	pta	Cdet
D1	Baseline	0.8590	0.6867	0.7632	0.3133	0.0783	0.144
	LDA	0.9824	0.8060	0.8190	0.1940	0.0485	0.0881
D2	Baseline	0.6685	0.6013	0.6332	0.3987	0.0997	0.1811
	LDA	0.7537	0.7060	0.7291	0.2940	0.0735	0.1335
D3	Baseline	0.7648	0.6313	0.6917	0.3687	0.0922	0.1674
	LDA	0.8282	0.8053	0.8166	0.1947	0.0487	0.0884
D4	Baseline	0.6678	0.5567	0.6072	0.4433	0.1108	0.2013
	LDA	0.8221	0.7713	0.7959	0.2287	0.0572	0.1039
D5	Baseline	0.8138	0.6893	0.7464	0.3107	0.0777	0.1411

	LDA	0.8337	0.8133	0.8234	0.1867	0.0467	0.0848
--	-----	--------	--------	--------	--------	--------	--------

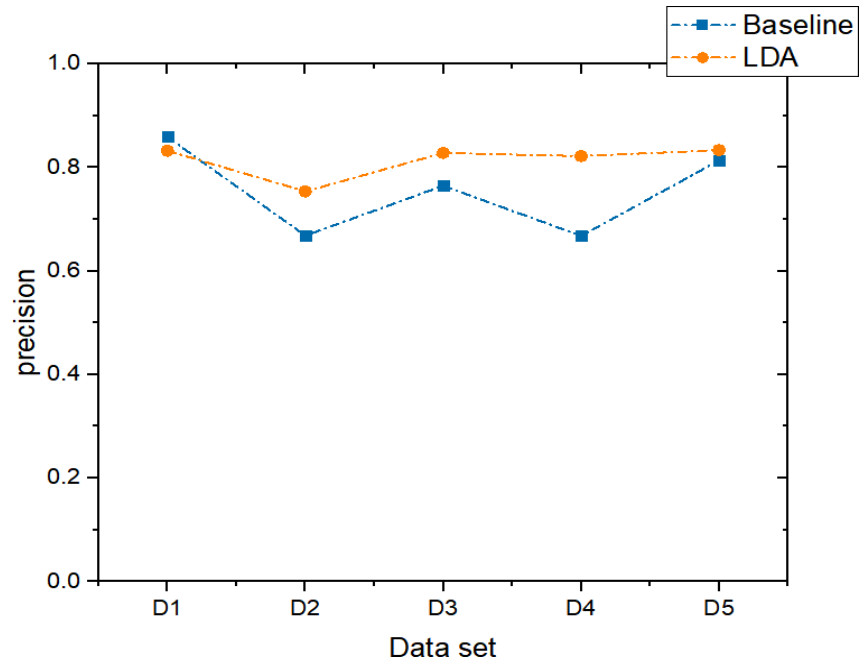


Fig 5.1. Precision of topic classification

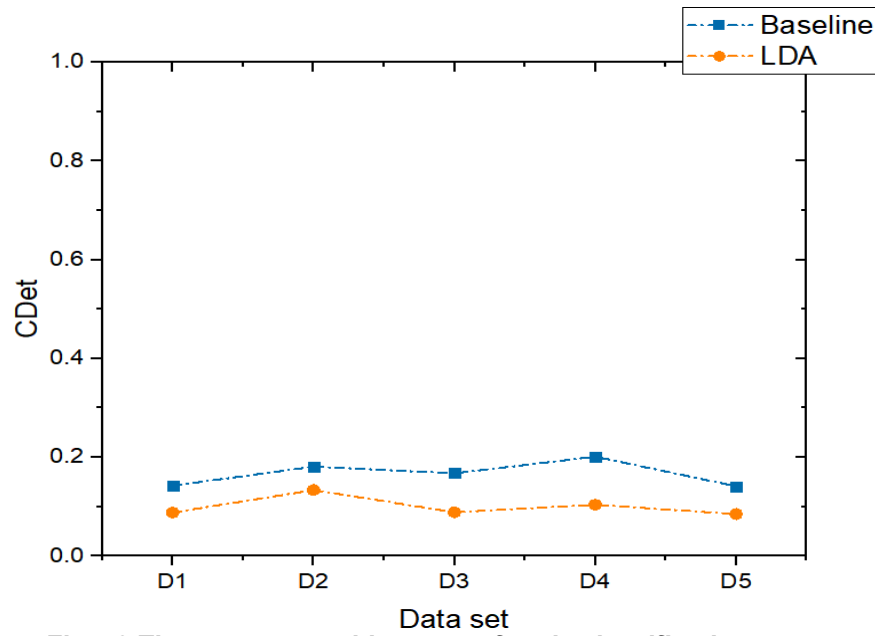


Fig 5.2 The error recognition cost of topic classification

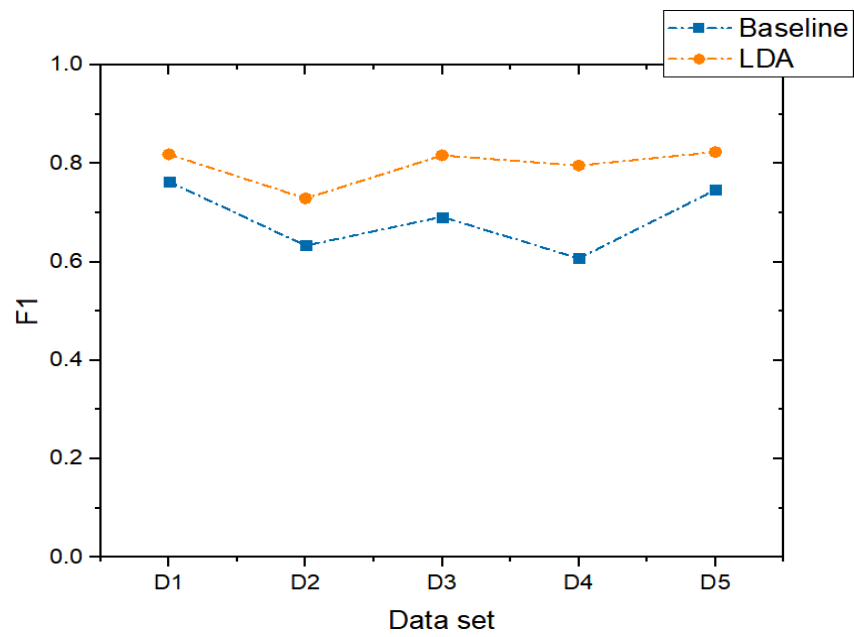


Fig 5.3 F1 of topic classification.

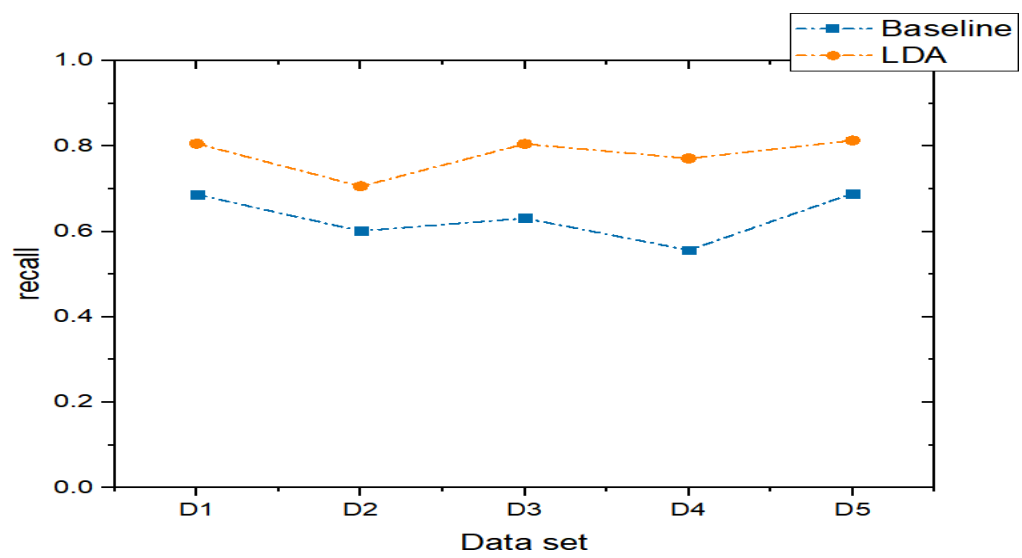


Fig 5.4 Recall of topic classification.

CHAPTER 6

SOFTWARE TESTING

The test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. The liner statement is considered as a scenario. The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases. In the test scenario, there is a detailed testing process due to many associated test cases. Before performing the test scenario, the tester has to consider the test cases for each scenario.

Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed. The most popular testing documentation files are test reports, plans, and checklists. These documents are used to outline the team's workload and keep track of the process. Let's take a look at the key requirements for these files and see how they contribute to the process. Test strategy. An outline of the full approach to product testing. As the project moves along, developers, designers, product owners can come back to the document and see if the actual performance corresponds to the planned activities.

Test data. The data that testers enter into the software to verify certain features and their outputs. Examples of such data can be fake user profiles, statistics, media content, similar to files that would be uploaded by an end-user in a ready solution.

Test plans. A file that describes the strategy, resources, environment, limitations, and schedule of the testing process. It's the fullest testing document, essential for informed planning. Such a document is distributed between team members and shared with all stakeholders.

Test scenarios. In scenarios, testers break down the product's functionality and interface by modules and provide real-time status updates at all testing stages. A module can be described by a single statement, or require hundreds of statuses, depending on its size and scope.

- Testing can be done in the early phases of the software development lifecycle when other modules may not be available for integration
- Fixing an issue in Unit Testing can fix many other issues occurring in later

development and testing stages

- Cost of fixing a defect found in Unit Testing is very less than the one found in the system or acceptance testing
- Code coverage can be measured
- Fewer bugs in the System and Acceptance testing
- Code completeness can be demonstrated using unit tests. This is more useful in the agile process. Testers don't get the functional builds to test until integration is completed.
- Code completion cannot be justified by showing that you have written and checked in the code. But running Unit tests can demonstrate code completeness.
- Expect robust design and development as developers write test cases by understanding the specifications first.
- Easily identify who broke the build
- Saves development time: Code completion may take more time but due to decreased defect count overall development time can be saved.

6.1 GENERAL

Unit Testing frameworks are mostly used to help write unit tests quickly and easily. Most of the programming languages do not support unit testing with the inbuilt compiler. Third-party open source and commercial tools can be used to make unit testing even more fun.

List of popular Unit Testing tools for different programming languages:

Java framework – JUnit

PHP framework – PHPUnit

C++ frameworks – UnitTest++ and Google C++

.NET framework – NUnit

Python framework – py.test

Functional Testing is a type of black box testing whereby each part of the system is tested against functional specification/requirements. For instance, seek answers to the following questions,

Are you able to login to a system after entering correct credentials?

Does your payment gateway prompt an error message when you enter incorrect card number? Does your “add a customer” screen adds a customer to your records successfully?

Well, the above questions are mere samples to perform full-fledged functional testing of a system.

6.2 Test Driven Development

Test Driven Development, or TDD, is a code design technique where the programmer writes a test before any production code, and then writes the code that will make that test pass. The idea is that with a tiny bit of assurance from that initial test, the programmer can feel free to refactor and refactor some more to get the cleanest code they know how to write. The idea is simple, but like most simple things, the execution is hard. TDD requires a completely different mind set from what most people are used to and the tenacity to deal with a learning curve that may slow you down at first.

6.3 UNIT TESTING

UNIT TESTING is a level of software testing where individual units/components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/fake objects are used to assist in unit testing.

A unit can be almost anything you want it to be -- a line of code, a method, or a class. Generally though, smaller is better. Smaller tests give you a much more granular view of how your code is performing. There is also the practical aspect that when you test very small units, your tests can be run fast; like a thousand tests in a second fast.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details. Unit Testing is not a new concept. It's been there since the early days of

programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test driven development TDD or test-first development). Most of us might know the classic definition of Unit Testing – “Unit Testing is the method of verifying the smallest piece of testable code against its purpose.” If the purpose or requirement failed then the unit test has failed.

In simple words, Unit Testing means – writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.

6.4 BLACKBOX TESTING

During functional testing, testers verify the app features against the user specifications. This is completely different from testing done by developers which is unit testing. It checks whether the code works as expected. Because unit testing focuses on the internal structure of the code, it is called the white box testing. On the other hand, functional testing checks app’s functionalities without looking at the internal structure of the code, hence it is called black box testing. Despite how flawless the various individual code components may be, it is essential to check that the app is functioning as expected, when all components are combined. Here you can find a detailed comparison between functional testing vs unit testing.

6.5 INTEGRATION TESTING: Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

INTEGRATION TESTING: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

COMPONENT INTEGRATION TESTING: Testing performed to expose defects in the interfaces and interaction between integrated components. System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Integration tests determine if independently developed units of software work correctly when they are connected to each other. The term has become blurred even by the diffuse standards of the software industry, so I've been wary of using it in my writing. In particular, many people assume integration tests are necessarily broad in scope, while they can be more effectively done with a narrower scope.

As often with these things, it's best to start with a bit of history. When I first learned about integration testing, it was in the 1980's and the waterfall was the dominant influence of software development thinking. In a larger project, we would have a design phase that would specify the interface and behavior of the various modules in the system. Modules would then be assigned to developers to program. It was not unusual for one programmer to be responsible for a single module, but this would be big enough that it could take months to build it. All this work was done in isolation, and when the programmer believed it was finished they would hand it over to QA for testing.

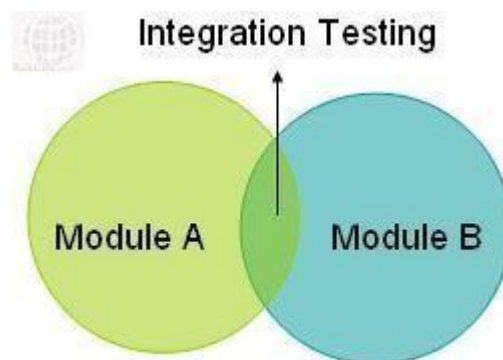


Fig 6.1 Integration Testing

6.6 SYSTEM TESTING

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not. System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product.

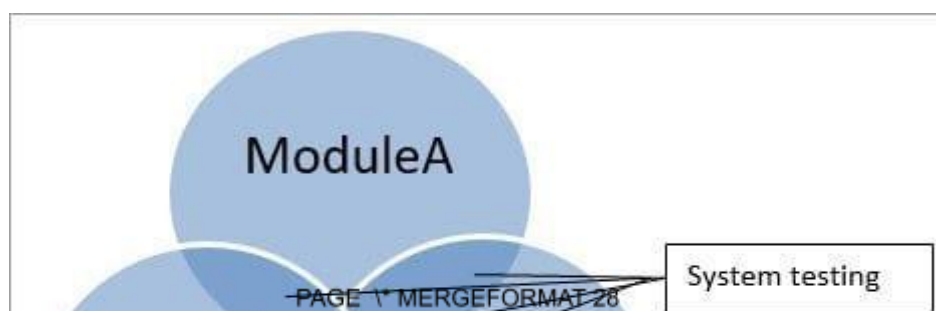


Fig 6.2 Module Testing

System testing is a method of monitoring and assessing the behaviour of the complete and fully-integrated software product or system, on the basis of pre-decided specifications and functional requirements. It is a solution to the question "whether the complete system functions in accordance to its pre-defined requirements?"

It's comes under black box testing i.e. only external working features of the software are evaluated during this testing. It does not requires any internal knowledge of the coding, programming, design, etc., and is completely based on users-perspective.

A black box testing type, system testing is the first testing technique that carries out the ta sk of testing a software product as a whole. This System testing tests the integrated system and validates whether it meets the specified requirements of the client.

System testing is a process of testing the entire system that is fully functional, in order to ensure the system is bound to all the requirements provided by the client in the form of the functional specification or system specification documentation. In most cases, it is done next to the Integration testing, as this testing should be covering the end-to-end system's actual routine. This type of testing requires a dedicated Test Plan and other test documentation derived from the system specification document that should cover both software and hardware requirements. By this test, we uncover the errors. It ensures that all the system works as expected. We check System performance and functionality to get a quality product. System testing is nothing but testing the system as a whole. This testing checks complete end-to-end scenario as per the customer's point of view. Functional and Non-Functional tests also done by System testing. All things are done to maintain trust within the development that the system is defect-free and bug-free. System testing is also intended to test hardware/software requirements specifications. System testing is more of a limited type of testing; it seeks to detect both defects within the "inter-assemblages".

6.7 SANITY TESTING

Sanity Testing is done when as a QA we do not have sufficient time to run all the test cases, be it Functional Testing, UI, OS or Browser Testing. Sanity testing is a subset of regression testing. After receiving the software build, sanity testing is performed to

ensure that the code changes introduced are working as expected. This testing is a checkpoint to determine if testing for the build can proceed or not. Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as “Day 0” check. It is done at the “build level”. It helps not to waste the testing time to simply testing the whole application when the key features don’t work or the key bugs have not been fixed yet. Here our focus will be on primary and core application work flow. To conduct smoke testing, we don’t write test cases. We just pick the necessary test cases from already written test cases. Do we really write test cases for all testing types? Here in this article, we have given clear idea on choosing testing types to write test cases.

As mentioned earlier, here in Smoke Testing, our main focus will be on core application work flow. So we pick the test cases from our test suite which cover major functionality of the application. In general, we pick minimal number of test cases that won’t take more than half an hour to execute. After receiving a Software build with the minor issues fixes in code or functionality, Sanity testing is carry out to check whether the bugs reported in previous build are fixed & there is regression introduced due to these fixes i.e. not breaking any previously working functionality. The main aim of Sanity testing to check the planned functionality is working as expected. Instead of doing whole regression testing the Sanity testing is perform.

6.8 REGRESSION TEESTING

Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

Regression Testing is a Software Testing type in which test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs. This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix. For regression testing to be effective, it needs to be seen as one part of a comprehensive testing methodology that is cost-effective and efficient while still incorporating enough variety—such as well-designed frontend UI automated tests alongside targeted unit testing, based on smart risk prioritization—to prevent any aspects of your software applications from going unchecked. These days, many Agile work environments employing workflow practices such as XP (Extreme Programming), RUP (Rational Unified Process), or

Scrum appreciate regression testing as an essential aspect of a dynamic, iterative development and deployment schedule.

But no matter what software development and quality-assurance process your organization uses, if you take the time to put in enough careful planning up front, crafting a clear and diverse testing strategy with automated regression testing at its core, you can help prevent projects from going over budget, keep your team on track, and, most importantly, prevent unexpected bugs from damaging your products and your company's bottom line. Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size. The process incorporates "performance" indicators such as:

- Browser, page, and network response times
- Server request processing times
- Acceptable concurrent user volumes

Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value. Here Increasing load means increasing number of concurrent users, transactions & check the behavior of application under test. It is normally carried out underneath controlled environment in order to distinguish between two different systems. It is also called as "Endurance testing" and "Volume testing". The main purpose of load testing is to monitor the response time and staying power of application when system is performing well under heavy load. Load testing comes under the Non Functional Testing & it is designed to test the non-functional requirements of a software application.

Load testing is perform to make sure that what amount of load can be withstand the application under test. The successfully executed load testing is only if the specified test cases are executed without any error in allocated time.

Simple examples of load testing:

- Testing printer by sending large job.
- Editing a very large document for testing of word processor
- Continuously reading and writing data into hard disk.
- Running multiple applications simultaneously on server.

- Testing of mail server by accessing thousands of mailboxes
- In case of zero-volume testing & system fed with zero load..

Chapter 7

Conclusion And Feature Enhancement

7.1 Conclusion

We have offered visual techniques for making comparisons associated with the tasks of using topic models to understand topics, similarity, and change within text corpora. Among these techniques, we have introduced buddy plots, a novel visualization for viewing changes across two sets of distances by restricting our vantage point to that of a single reference document. Finally, we have described a number of use cases that exhibit the variety of comparisons that can be made with these methods.

7.2 Future Enhancement

In the future, for predicting the re-hotting time points more accurately, some data pre- processing methods can be used to reduce the noise in OSN data.

REFERENCES

- 1) K. Liu, J. Xu, L. Zhang, Z. Ding, and M. Li, "Discovering Hot Topics from Geo-tagged Video," *Neurocomputing*, vol. 105, pp. 90–99, 2013.
- 2) X. Wang, L. Qi, C. Chen, J. Tang, and M. Jiang, "Grey System Theory based Prediction for Topic Trend on Internet," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 191–200, 2014.
- 3) P. Velardi, G. Stilo, A. E. Tozzi, and F. Gesualdo, "Twitter Mining for Fine-grained Syndromic Surveillance," *Artificial Intelligence in Medicine*, vol. 61, pp. 153–163, 2014.
- 4) X. Zhang, X. Chen, Y. Chen, S. Wang, Z. Li, and J. Xia, "Event Detection and Popularity Prediction in Microblogging," *Neurocomputing*, vol. 149, pp. 1469–1480, 2015.
- 5) K. Liu, J. Xu, L. Zhang, Z. Ding, and M. Li, "Discovering Hot Topics from Geo-tagged Video," *Neurocomputing*, vol. 105, pp. 90–99, 2013.
- 6) X. Wang, L. Qi, C. Chen, J. Tang, and M. Jiang, "Grey System Theory based Prediction for Topic Trend on Internet," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 191–200, 2014.
- 7) P. Velardi, G. Stilo, A. E. Tozzi, and F. Gesualdo, "Twitter Mining for Fine-grained Syndromic Surveillance," *Artificial Intelligence in Medicine*, vol. 61, pp. 153–163, 2014.
- 8) T. Muhlbacher, H. Piringer, S. Gratzl, M. Sedlmair, and M. Streit. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *Visualization and Computer Graphics*, IEEE Transactions on, 20(12):1643–1652, 2014.
- 9) Y. Zhang and A. Song, "Application of improved algorithm based on K-means in microblog topic discovery," *Comput. Syst. Appl.*, vol. 25, no. 10, pp. 308–311, Dec. 2016.
- 10) L. Che and X. Yang, "News topic discovery model of multi feature fusion text clustering," *J. Nat. Univ. Defense Technol.*, vol. 39, no. 3, pp. 85–90, Jan. 2017.

APPENDICES

Screenshots

Out[26]:

	headlines	text
0	upGrad learner switches to career in ML & AI w...	Saurav Kant, an alumnus of upGrad and IIIT-B's...
1	Delhi techie wins free food from Swiggy for on...	Kunal Shah's credit card bill payment platform...
2	New Zealand end Rohit Sharma-led India's 12-ma...	New Zealand defeated India by 8 wickets in the...
3	Aegon life iTerm insurance plan helps customer...	With Aegon Life iTerm Insurance plan, customer...
4	Have known Hirani for yrs, what if MeToo claim...	Speaking about the sexual harassment allegatio...

Fig 1: Summary Outlook

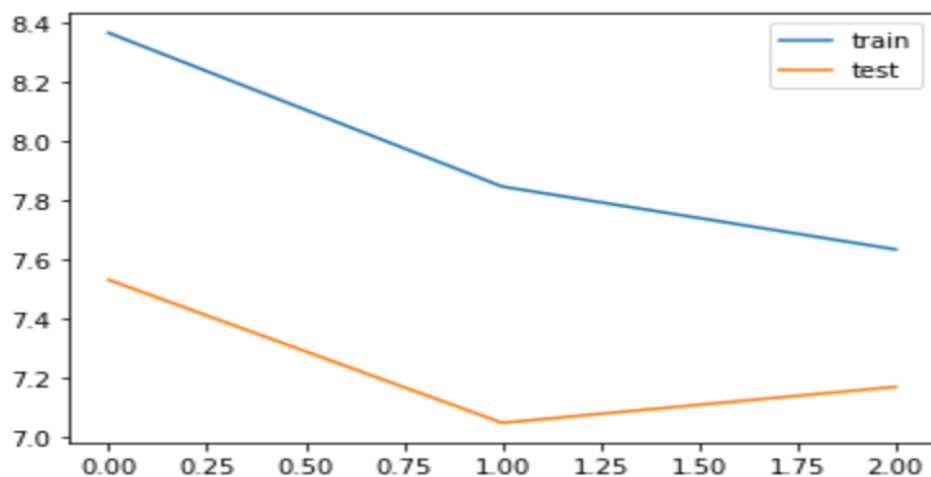


Fig 2: Topic Figstrength change chart.

Out[28]:

	headlines	text
0	upGrad learner switches to career in ML & AI w...	Saurav Kant, an alumnus of upGrad and IIIT-B's...
1	Delhi techie wins free food from Swiggy for on...	Kunal Shah's credit card bill payment platform...
2	New Zealand end Rohit Sharma-led India's 12-ma...	New Zealand defeated India by 8 wickets in the...
3	Aegon life iTerm insurance plan helps customer...	With Aegon Life iTerm Insurance plan, customer...
4	Have known Hirani for yrs, what if MeToo claim...	Speaking about the sexual harassment allegatio...
...	...	...
2995	RCom deposits ₹131 crore as part payment ...	Anil Ambani-led Reliance Communications (RCom)...

Fig 3: Summary Outlook

SAMPLE CODING

```
import warningswarnings.filterwarnings("ignore")

import pandas as pd

import tensorflow.keras

from tensorflow.keras.models import Model,load_model

from tensorflow.keras.layers import Input, GRU, Dense, Embedding,
TimeDistributed, Concatenate

from tensorflow.keras import backend

from tensorflow.keras import backend

from tensorflow.keras.callbacks import EarlyStopping

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad_sequences

from sklearn import model_selection

import numpy as np

import nltk

from nltk.corpus import stopwords

import re

import tensorflow as tf

import os

from tensorflow.keras.layers import Layer

from tensorflow.keras import backend as K

from nltk.translate.bleu_score import corpus_bleu

import nltk.translate.gleu_score as gleu
```

```

class AttentionLayer(Layer):

    def __init__(self, **kwargs):
        super(AttentionLayer, self).__init__(**kwargs)

    def build(self, input_shape):

        assert isinstance(input_shape, list)

        self.W_a = self.add_weight(name='W_a',
                                    shape=tf.TensorShape((input_shape[0][2],
input_shape[0][2])),
                                    initializer='uniform',
                                    trainable=True)

        self.U_a = self.add_weight(name='U_a',
                                    shape=tf.TensorShape((input_shape[1][2],
input_shape[0][2])),
                                    initializer='uniform',
                                    trainable=True)

        self.V_a = self.add_weight(name='V_a',
                                    shape=tf.TensorShape((input_shape[0][2], 1)),
                                    initializer='uniform',
                                    trainable=True)

        super(AttentionLayer, self).build(input_shape) # Be sure to call this at the end

    def call(self, inputs, verbose=False):
        assert type(inputs) == list
        encoder_out_seq, decoder_out_seq = inputs

        def energy_step(inputs, states):

```

```

        en_seq_len, en_hidden = encoder_out_seq.shape[1],
encoder_out_seq.shape[2]

        de_hidden = inputs.shape[-1]

        reshaped_enc_outputs = K.reshape(encoder_out_seq, (-1, en_hidden))

        W_a_dot_s = K.reshape(K.dot(reshaped_enc_outputs, self.W_a), (-1,
en_seq_len, en_hidden))

        if verbose:

            print('wa.s>', W_a_dot_s.shape)

        U_a_dot_h = K.expand_dims(K.dot(inputs, self.U_a), 1) # <= batch_size,
1, latent_dim

        if verbose:

            print('Ua.h>', U_a_dot_h.shape)

        reshaped_Ws_plus_Uh = K.tanh(K.reshape(W_a_dot_s + U_a_dot_h, (-1,
en_hidden)))

        if verbose:

            print('Ws+Uh>', reshaped_Ws_plus_Uh.shape)

        e_i = K.reshape(K.dot(reshaped_Ws_plus_Uh, self.V_a), (-1, en_seq_len))

        e_i = K.softmax(e_i)

        if verbose:

            print('ei>', e_i.shape)

        return e_i, [e_i]

def context_step(inputs, states):

    c_i = K.sum(encoder_out_seq * K.expand_dims(inputs, -1), axis=1)

    if verbose:

        print('ci>', c_i.shape)

    return c_i, [c_i]

def create_initial_state(inputs, hidden_size):

```

```

fake_state = K.zeros_like(inputs)

fake_state = K.sum(fake_state, axis=[1, 2])

fake_state = K.expand_dims(fake_state)

fake_state = K.tile(fake_state, [1, hidden_size])

return fake_state


fake_state_c = create_initial_state(encoder_out_seq,
encoder_out_seq.shape[-1])

fake_state_e = create_initial_state(encoder_out_seq,
encoder_out_seq.shape[1])

last_out, e_outputs, _ = K.rnn(
    energy_step, decoder_out_seq, [fake_state_e],
)

last_out, c_outputs, _ = K.rnn(
    context_step, e_outputs, [fake_state_c],
)

return c_outputs, e_output

def compute_output_shape(self, input_shape):
    return [
        tf.TensorShape((input_shape[1][0], input_shape[1][1], input_shape[1][2])),
        tf.TensorShape((input_shape[1][0], input_shape[1][1], input_shape[0][1]))
    ]

df = pd.read_csv('news_summary_more.csv', encoding='iso-8859-1', nrows=3000)

df.shape

(3000, 2)

df.info()

<class 'pandas.core.frame.DataFrame'>

```

RangeIndex: 3000 entries, 0 to 2999

Data columns (total 2 columns):

#	Column	Non-Null Count	Dtype
---	--------	----------------	-------

---	-----	-----	-----
-----	-------	-------	-------

0	headlines	3000 non-null	object
---	-----------	---------------	--------

1	text	3000 non-null	object
---	------	---------------	--------

dtypes: object(2)

memory usage: 47.0+ KB

df.head()

df.info()df.isnull().sum()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 3000 entries, 0 to 2999

Data columns (total 2 columns):

#	Column	Non-Null Count	Dtype
---	--------	----------------	-------

---	-----	-----	-----
-----	-------	-------	-------

0	headlines	3000 non-null	object
---	-----------	---------------	--------

1	text	3000 non-null	object
---	------	---------------	--------

dtypes: object(2)

memory usage: 47.0+ KB

headlines 0

text 0

dtype: int64

df.dropna(inplace=True,axis=0)df.drop_duplicates(subset='text')

3000 rows × 2 columns

df.isnull().sum()

headlines 0

```

text      0
dtype: int64

stop_words=[sw for sw in all_stop_words if not sw in wanted_words]

def cleaning_data(textdata):

    textdatanew = textdata.lower()

    textdatanew=[contractions[token] if token in contractions else token for token in
textdatanew.split()]

    textdatanew=" ".join(textdatanew)

    textdatanew=[s for s in textdatanew.split() if not s in stop_words]

    textdatanew=" ".join(textdatanew)

    textdatanew = re.sub(r'^a-zA-Z', ' ', textdatanew)

    return textdatanew

new_text = []new_summary=[]

for text in df.text:

    new_text.append(cleaning_data(text))

for summary in df.headlines:

    new_summary.append(cleaning_data(summary))

df['New_Text']=new_textdf['New_Summary']=new_summary

df['New_Summary'] = df['New_Summary'].apply(lambda x : '<START>'+ x +
'<EOS>')

for i in range(7):

    print("Text ",i+1," : ")

    print(df['New_Text'][i])

    print("Summary: ")

    print(df['New_Summary'][i])

    print('-----')

print(df.shape)

```

(3000, 4)

```
x_train, x_test, y_train, y_test= model_selection.train_test_split(df['New_Text'],  
df['New_Summary'], test_size=0.20,random_state=30, shuffle=False)
```

```
print('x_train: ',x_train.shape)print('y_train: ', y_train.shape)print('x_test: ',  
x_test.shape)print('y_test: ', y_test.shape)
```

x_train: (2400,)

y_train: (2400,)

x_test: (600,)

y_test: (600,)

```
text_lengths=[]summary_lengths=[]
```

```
for text in df['New_Text']:
```

```
    text_lengths.append(len(text.split()))
```

```
for summary in df.New_Summary:
```

```
    summary_lengths.append(len(summary.split()))
```

```
print(text_lengths[:10])print(summary_lengths[:10])
```

[43, 47, 41, 35, 40, 39, 39, 38, 39, 35]

[8, 9, 11, 9, 8, 9, 9, 8, 9, 8]

```
print('Text Lengths:')print(pd.DataFrame(text_lengths).describe())
```

Text Lengths:

0

count 3000.000000

mean 39.033667

std 4.126040

min 1.000000

25% 36.000000

50% 39.000000

75% 42.000000

```

max    55.000000

print("85% of text lengths is: ",np.percentile(text_lengths,85))print("90% of text
lengths is: ",np.percentile(text_lengths,90))

85% of text lengths is:  43.0
90% of text lengths is:  44.0

print('Summary Lengths:')print(pd.DataFrame(summary_lengths).describe())

Summary Lengths:
      0
count 3000.000000
mean    8.236333
std     1.371535
min     1.000000
25%     7.000000
50%     8.000000
75%     9.000000
max    13.000000

print("85% of summary lengths is: ",np.percentile(summary_lengths,85))print("90%
of summary lengths is: ",np.percentile(summary_lengths,90))

85% of summary lengths is:  10.0
90% of summary lengths is:  10.0

max_text_length=95max_summary_length=8

text_tokenizer= Tokenizer()

text_tokenizer.fit_on_texts(x_train)

text_vocab_size=len(text_tokenizer.word_counts)+1print('Text Vocabulary Size=
',text_vocab_size)

Text Vocabulary Size=  14309

```

```

x_train=text_tokenizer.texts_to_sequences(x_train)x_test=text_tokenizer.texts_to_
sequences(x_test)

x_train= pad_sequences(x_train, maxlen= max_text_length, padding='post',
truncating='post', value=0)x_test= pad_sequences(x_test, maxlen=
max_text_length, padding='post', truncating='post', value=0)

summary_tokenizer= Tokenizer()

summary_tokenizer.fit_on_texts(y_train)

summary_vocab_size=len(summary_tokenizer.word_counts)+1print('Summary
Vocabulary Size= ',summary_vocab_size)

Summary Vocabulary Size= 6196

gru_3[0][0]

concat_layer (Concatenate)      (None, None, 1024) 0      gru_3[0][0]

                                attention_layer[0][0]

time_distributed (TimeDistribut (None, None, 6196) 6350900      concat_layer[0][0]

Total params: 23,678,004

Trainable params: 23,678,004

Non-trainable params: 0

model.compile(optimizer='rmsprop', loss='sparse_categorical_crossentropy')

early_stop= EarlyStopping(monitor='val_loss', mode='min', verbose=1)

history=model.fit([x_train,y_train[:, :-1]],
y_train.reshape(y_train.shape[0],y_train.shape[1], 1)[:,1:],
,epochs=15,callbacks=[early_stop],batch_size=128,
validation_data=([x_test,y_test[:, :-1]],
y_test.reshape(y_test.shape[0],y_test.shape[1], 1)[:,1:]))

def decode_sequence(input_seq):

    pred_output, pred_h = encoder_inf_model.predict(input_seq)

    target_seq = np.zeros((1,1))

    target_seq[0, 0] = summary_tokenizer.word_index['start']

```

```

stop_condition = False

decoded_sentence = ""

while not stop_condition:

    output_tokens, h = decoder_inf_model.predict([target_seq] + [pred_output,
pred_h])

    sampled_token_index = np.argmax(output_tokens[0, -1, :])

    if(sampled_token_index not in summary_tokenizer.index_word):

        sampled_token='eos'

    else:

        sampled_token = summary_tokenizer.index_word[sampled_token_index]

    if(sampled_token!='eos'):

        decoded_sentence += ' '+sampled_token

    if(sampled_token == 'eos' or len(decoded_sentence.split()) >=
(max_summary_length-1)):

        stop_condition = True

        target_seq = np.zeros((1,1))

        target_seq[0, 0] = sampled_token_index

        pred_h = h

    return decoded_sentence

def seq2summary(input_seq):

    newString=""

    for i in input_seq:

        if((i!=0 and i!=summary_tokenizer.word_index['start']) and
i!=summary_tokenizer.word_index['eos']):

            newString=newString+summary_tokenizer.index_word[i]+' '

    return newString

```

```

def seq2text(input_seq):
    newString=""
    for i in input_seq:
        if(i!=0):
            newString=newString+text_tokenizer.index_word[i]+' '
    return newString

for i in range(230, 250):
    print("Review:",seq2text(x_test[i]))
    print("Original summary:",seq2summary(y_test[i]))
    print("Predicted
summary:",decode_sequence(x_test[i].reshape(1,max_text_length)))
    print("\n")
    score = corpus_bleu([seq2summary(y_test[i]).split()],
[decode_sequence(x_test[i].reshape(1,max_text_length)).split()])
    print("BLEU Score : ",score)
    gelu_score = gleu.sentence_gleu([seq2summary(y_test[i]).split()],
decode_sequence(x_test[i].reshape(1,max_text_length)).split())
    print("GELU Score : ", gelu_score)

```

C. PUBLICATION WITH PLAGIARISM REPORT:

A Knowledgeable and Smart semantic Methodology for Topic Modelling

Ch Teja Reddy[1], Ch Trinadh Sai Bhargav[2], Jeberson retnaraj[3] [1][2]

Abstract— Text preprocessing is a fundamental piece of any NLP framework, since the characters, words, and sentences recognized at this stage are the central units passed to all further preparing stages, from analysis and labeling segments, like morphological analyzers and grammatical form taggers, through applications, like data recovery and machine interpretation frameworks. As more data is accessible, it turns out to be extremely hard to get to the information from enormous documents. So we need a few strategies to coordinate, look and understand huge amounts of data. Topic Modeling fills in as an effective technique for getting to the information. In this paper, Document Topic Modeling approach has been proposed to create topics and word cloud from the enormous assortment of printed data.

Keywords— Text Processing, Natural Language Processing, Text Modelling, Text Summarization, Latent Dirichlet Allocation

I. A GENERAL SYSTEM

Natural language text is by and large not comprised of the short, perfect, all around framed, and very much delimited sentences we find in reading material; and for languages like Chinese, Japanese, or Thai, which don't share the obviously simple space-delimited tokenization we may accept to be a property of languages like English, the capacity to address issues of tokenization is fundamental to making headway by any means. We additionally treat lexical analysis as a different advance all the while. Somewhat this better grained deterioration mirrors our present status of information about language handling:

we know a considerable amount about broad methods for tokenization, lexical analysis, and syntactic analysis, however significantly less about semantics and talk level preparing.

Life Cycle of a Corpus is illustrated as shown in below

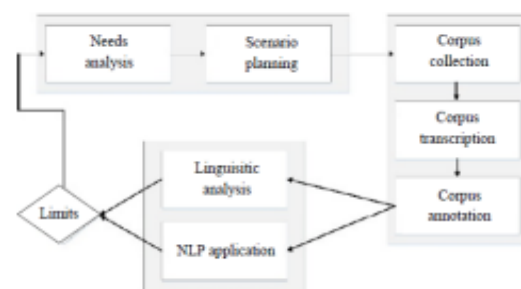


Fig 1: Text Processing

A. Text Classification

Text classification, the classification of text documents as per classes or topics, is a significant part of any text handling framework. There is an enormous group of work which utilizes content – the words showing up in the documents, the design of the documents – and outer sources to construct exact document classifiers. What's more, there is a developing assortment of writing on methods which endeavour to utilize the connection structure among the documents to improve document classification execution. Text documents can be associated together in an assortment of ways. The most widely recognized connection structure is the reference chart: e.g., papers refer to different papers and website pages connect to different pages. However, joins among papers can be developed from different relationships like

co-creator, co-reference, appearance at a meeting scene, and others. These can be consolidated together to make an interlinked assortment of text documents.

B. TOPIC TRACKING

Topic tracking is based on the definition of a topic with one or more stories and then uses this topic to find all relevant news stories in the stories stream. There are two main research directions in traditional topic tracking research. They are based on rules and statistics. The key technology of rule-based research is to analyze the association and inheritance relationship between text content and use the related domain knowledge to summarize the related news texts. The statistical method is mainly based on the probability distribution of text features, using mathematical statistics to determine the relevance of text to topic models. In the topic tracking research, researchers have tried a variety of methods.

Allan et al. Used the Rocchio algorithm to perform topic tracking. By adding weight to specific features in the topic model, the topic model was continuously updated during the tracking of the topic. Ren et al. Used K-Modes clustering to conduct adaptive topic tracking research. First, the texts were clustered. The topic model was represented by the center of each cluster, and the named entity vector was used as the cluster center. The cluster center was iterated continuously during the tracking process until the topic cluster center was stable. Bai et al.

[18] proposed a method of tracking user relationship, which first mapped the blog to the feature space, and then used the improved K-Means algorithm to perform binary clustering to track related topics.

C. TOPIC TRACKING

Topic tracking is based on the definition of a topic with one or more stories and then uses

this topic to find all relevant news stories in the stories stream. There are two main research directions in traditional topic tracking research. They are based on rules and statistics. The key technology of rule-based research is to analyze the association and inheritance relationship between text content and use the related domain knowledge to summarize the related news texts.

The statistical method is mainly based on the probability distribution of text features, using mathematical statistics to determine the relevance of text to topic models. In the topic tracking research, researchers have tried a variety of methods. Allan et al. [16] used the Rocchio algorithm to perform topic tracking. By adding weight to specific features in the topic model, the topic model was continuously updated during the tracking of the topic. Ren et al. [17] used K-Modes clustering to conduct adaptive topic tracking research. First, the texts were clustered. The topic model was represented by the center of each cluster, and the named entity vector was used as the cluster center. The cluster center was iterated continuously during the tracking process until the topic cluster center was stable. Bai et al. [18] proposed a method of tracking user relationship, which first mapped the blog to the feature space, and then used the improved K-Means algorithm to perform binary clustering to track related topics.

D. Topic Modelling Problem

The focal issue is that this design—the list of thoughts contained in the articles and which different articles are about similar sorts of thoughts—isn't promptly accessible in most present day assortments, and the size and development pace of these assortments block us from building it by hand. To build up the important devices for investigating and perusing present day computerized libraries, we require mechanized methods of getting sorted out, overseeing, and conveying their substance.

E. Latent Dirichlet Allocation LDA

The Latent Dirichlet Allocation is a speculation of the beta appropriation, yet expanded to handle multivariate information. The beta dispersion is a circulation of probabilities.

The probability distribution can be written as below equation

$$p(\theta|\alpha) = \left(\frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \right) \theta_1^{\alpha_1-1}, \dots, \theta_k^{\alpha_k-1}$$

$$p(\theta, \mathbf{z}, \mathbf{w} | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) p(w_n | z_n, \beta)$$

The Joint Distribution can be written as the below equation
this model, it assumes that words appearing in the text are independent and irrelevant, and it considers that each document is composed of several implicit topics. And these hidden topics are made up of some specific words in the text. LDA is a typical probability model, determined by the parameters (α , β). α represents the relative strength between the implicit topics in the text set, and β represents the probability.

F. Histogram

A histogram is a classed portrayal of cardinal variables. What recognizes the histogram from other realistic portrayals is that it communicates relative class recurrence not by stature but rather by region (height * width). The stature of the bars addresses recurrence thickness. As the recurrence thickness for a class increments, so too does its territory (height * width). The histogram submits to the rule that the stretches in a graph ought to be chosen so the information are not contorted. In the histogram, the portion of region for a particular class comparative with the whole zone of all classes

risers to the overall recurrence of the particular class.

G. Scatterplots

SCATTERPLOTS CAN BE UTILIZED TO DISTINGUISH WHETHER ANY RELATIONSHIP EXISTS BETWEEN TWO PERSISTENT VARIABLES DEPENDENT ON THE PROPORTION OR STRETCH SCALES. THE TWO VARIABLES ARE PLOTTED ON THE X-AND Y-TOMAHAWKS. EACH POINT SHOWED ON THE SCATTERPLOT IS A SOLITARY PERCEPTION. THE SITUATION OF THE FACT IS CONTROLLED BY THE ESTIMATION OF THE TWO VARIABLES.

II DETAILS OF PROPOSED OPERATIONS

Topic models are a technique for removing data from collections of text to perceive what "topics" happen across every one of the documents. The instinct is that we anticipate that certain topics should show up additional in significant documents and not as much in unimportant documents. This may be helpful when utilizing the topics we partner with a document as keywords for better and more instinctive hunt, or when utilizing it for shorthand outline.

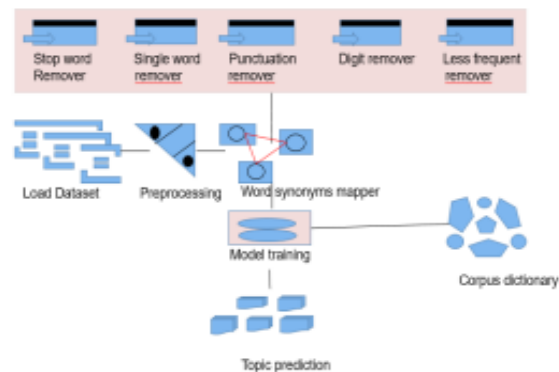


Fig 3: System Architecture

H. Tokenization

co-creator, co-reference, appearance at a meeting scene, and others. These can be consolidated together to make an interlinked assortment of text documents.

B. TOPIC TRACKING

Topic tracking is based on the definition of a topic with one or more stories and then uses this topic to find relevant news stories in the stories stream. There are two main research directions in traditional topic tracking research. They are based on

this topic to find all relevant news stories in the stories stream. There are two main research directions in traditional topic tracking research. They are based on rules and statistics. The key technology of rule-based research is to analyze the association and inheritance relationship between text content and use the related domain knowledge to summarize the related news texts.

The statistical method is mainly based on the probability distribution of text features, using

E. Latent Dirichlet Allocation LDA

The Latent Dirichlet Allocation is a speculation of the beta appropriation, yet expanded to handle multivariate information. The beta dispersion is a circulation of probabilities.

The probability distribution can be written as below equation

$$p(\theta|\alpha) = \left(\frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \right) \theta_1^{\alpha_1-1}, \dots, \theta_k^{\alpha_k-1}$$

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^N p(z_n|\theta) p(w_n|z_n, \beta)$$

The Joint Distribution can be written as the below equation

this model, it assumes that words appearing in the text are independent and irrelevant, and it considers that each document is composed of several implicit topics. And these hidden topics are made up of some specific words in the text. LDA is a typical probability model, determined by the parameters (α , β). α represents the relative strength between the implicit topics in the text set, and β represents the probability.

F. Histogram

A histogram is a classed portrayal of cardinal variables. What recognizes the histogram from other realistic portrayals is that it communicates relative class recurrence not by stature but rather by region (height * width). The stature of the bars addresses recurrence thickness. As the recurrence thickness for a class increments, so too does its territory (height * width). The histogram submits to the rule that the stretches in a graph ought to be chosen so the information are not contorted. In the histogram, the portion of region for a particular class comparative with the whole zone of all classes

risers to the overall recurrence of the particular class.

G. Scatterplots

SCATTERPLOTS CAN BE UTILIZED TO DISTINGUISH WHETHER ANY RELATIONSHIP EXISTS BETWEEN TWO PERSISTENT VARIABLES DEPENDENT ON THE PROPORTION OR STRETCH SCALES. THE TWO VARIABLES ARE PLOTTED ON THE X-AND Y-TOMAHAWKS. EACH POINT SHOWED ON THE SCATTERPLOT IS A SOLITARY PERCEPTION. THE SITUATION OF THE FACT IS CONTROLLED BY THE ESTIMATION OF THE TWO VARIABLES.

II DETAILS OF PROPOSED OPERATIONS

Topic models are a technique for removing data from collections of text to perceive what "topics" happen across every one of the documents. The instinct is that we anticipate that certain topics should show up additional in significant documents and not as much in unimportant documents. This may be helpful when utilizing the topics we partner with a document as keywords for better and more instinctive hunt, or when utilizing it for shorthand outline.

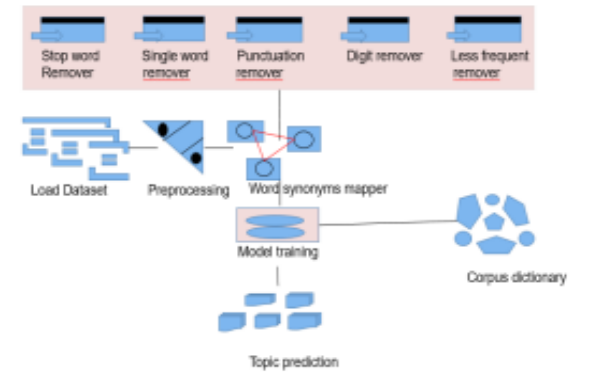


Fig 3: System Architecture

H. Tokenization

A word (Token) is the insignificant unit that a machine can understand and measure. So any content string can't be additionally prepared without going through tokenization. Tokenization is the way toward parting the crude string into significant tokens.

I. Stemming

Stemming, in strict terms, is the way toward chopping down the parts of a tree to its stem. So adequately, with the utilization of some fundamental guidelines, any token can be chopped down to its stem. Stemming is all the more a rough principle based interaction by which we need to club together various varieties of the token. Stemming is utilized to club both linguistic changes to the base of the word. While stemming is utilized more often than not for its effortlessness, there are instances of complex language or complex NLP errands where it's important to utilize lemmatization all things being equal. Lemmatization is a more hearty and precise method of consolidating syntactic varieties to the foundation of a word.

J. Lemmatization

Lemmatization is a more efficient method of changing over all the linguistic/bent types of the base of the word. Lemmatization utilizes setting and grammatical feature to decide the bent type of the word and applies diverse standardization rules for each grammatical feature to get the root word

K. Stop word removal

Stop word removal is quite possibly the most regularly utilized pre-processing ventures across various NLP applications. The thought is essentially eliminating the words that happen generally across every one of the documents in the corpus. Commonly, articles and pronouns are for the most part delegated stop words. These words have no importance in a portion of the NLP assignments like data recovery and characterization, which implies

these words are not discriminate. In actuality, in some Natural Language Processing applications stop word removal will have almost no effect. More often than not, the stop word list for the given language is a well hand-curated rundown of words that happen most normally across corpuses. While the stop word records for most languages are accessible on the web, these are additionally approaches to naturally produce the stop word list for the given corpus.

II. EXPERIMENT RESULTS

A. TOPIC DETECTION EVALUATION INDEX

In the topic identification process, the evaluation method in TDT2003 [29] is usually adopted. This method is to weigh and evaluate depending on the loss rate and false positive rate of the relevant reports in the topic detection and tracking results. Its calculation formula is as follows:

$$(C_{det})_{norm} = \frac{C_{miss} \cdot P_{miss} \cdot P_{target} + C_{fa} \cdot P_{fa} \cdot P_{nonn}}{C_{miss} \cdot P_{miss} \cdot P_{target} + C_{fa} \cdot P_{fa} \cdot P_{nonn}}$$

$$target / (min(C_{miss} \cdot P_{miss} \cdot P_{target}, C_{fa} \cdot P_{fa} \cdot P_{nonn})) \quad (8)$$

In the above formula, C_{det} represents the error recognition cost of the tracking system. The smaller the value is, the better the system tracking performance is. C_{miss} indicates the cost of tracking loss. C_{fa} indicates the cost of tracking the error report. P_{target} and P_{nonn} target are a priori probabilities of whether a report is related to a tracking topic. P_{miss} is the loss rate of the tracking report; P_{fa} is the false positive rate of the tracking report. In general, C_{miss} , C_{fa} , P_{nonn} , P_{target} are set to 1, 0.1, 0.98, and 0.02. F1 value are also used as evaluation indicators of the experiment. The specific formula is as follows: The precision P, the recall rate R and the F1 value are calculated as follows:

$$P = TP / (FP + TP) \quad (9)$$

$$R = TP / (FN + TP) \quad (10)$$

A word (Token) is the insignificant unit that a machine can understand and measure. So any content string can't be additionally prepared without going through tokenization. Tokenization is the way toward parting the crude string into significant tokens.

I. Stemming

Stemming, in strict terms, is the way toward chopping down the parts of a tree to its stem. So adequately, with the utilization of some fundamental guidelines, any token can be chopped down to its stem. Stemming is all the more a rough principle based interaction by which we need to club together various varieties of the token. Stemming is utilized to club both linguistic changes to the base of the word. While stemming is utilized more often than not for its effortlessness, there are instances of complex language or complex NLP errands where it's important to utilize lemmatization all things being equal. Lemmatization is a more hearty and precise method of consolidating syntactic varieties to the foundation of a word.

J. Lemmatization

Lemmatization is a more efficient method of changing over all the linguistic/bent types of the base of the word. Lemmatization utilizes setting and grammatical feature to decide the bent type of the word and applies diverse standardization rules for each grammatical feature to get the root word

K. Stop word removal

Stop word removal is quite possibly the most regularly utilized pre-processing ventures across various NLP applications. The thought is essentially eliminating the words that happen generally across every one of the documents in the corpus. Commonly, articles and pronouns are for the most part delegated stop words. These words have no importance in a portion of the NLP assignments like data recovery and characterization, which implies

these words are not discriminate. In actuality, in some Natural Language Processing applications stop word removal will have almost no effect. More often than not, the stop word list for the given language is a well hand-curated rundown of words that happen most normally across corpuses. While the stop word records for most languages are accessible on the web, these are additionally approaches to naturally produce the stop word list for the given corpus.

II. EXPERIMENT RESULTS

A. TOPIC DETECTION EVALUTION INDEX

In the topic identification process, the evaluation method in TDT2003 [29] is usually adopted. This method is to weigh and evaluate depending on the loss rate and false positive rate of the relevant reports in the topic detection and tracking results. Its calculation formula is as follows:

$$(C_{det})_{norm} = \frac{C_{miss} \cdot P_{miss} \cdot P_{target} + C_{fa} \cdot P_{fa} \cdot P_{nonn}}{target / (\min(C_{miss} \cdot P_{target}, C_{fa} \cdot P_{nonn}))}$$

(8)

In the above formula, C_{det} represents the error recognition cost of the tracking system. The smaller the value is, the better the system tracking performance is. C_{miss} indicates the cost of tracking loss. C_{fa} indicates the cost of tracking the error report. P_{target} and P_{nonn} target are a priori probabilities of whether a report is related to a tracking topic. P_{miss} is the loss rate of the tracking report; P_{fa} is the false positive rate of the tracking report. In general, C_{miss} , C_{fa} , P_{nonn} target, P_{target} are set to 1, 0.1, 0.98, and 0.02. F1 value are also used as evaluation indicators of the experiment. The specific formula is as follows: The precision P, the recall rate R and the F1 value are calculated as follows:

$$P = TP / (FP + TP) \quad (9)$$

$$R = TP / (FN + TP) \quad (10)$$

$$F1 = 2 * P * R / (P + R) \quad (11)$$

TP means the number of correctly identified news in the classification results of positive class. FP means the number of misidentified messages in the classification results of positive class. FN refers to the number of misidentified messages in the classification results of negative class. The F1 value is the harmonic mean of the precision and recall rate.

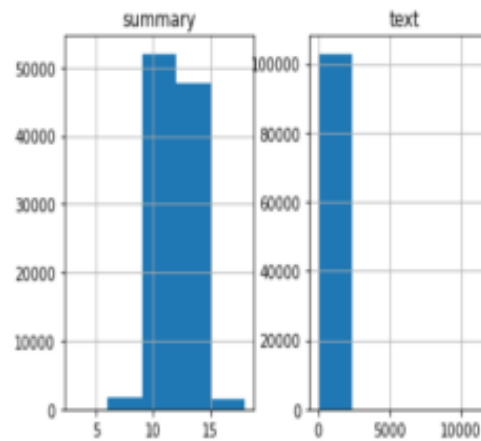


Fig 5: Topic strength change charts

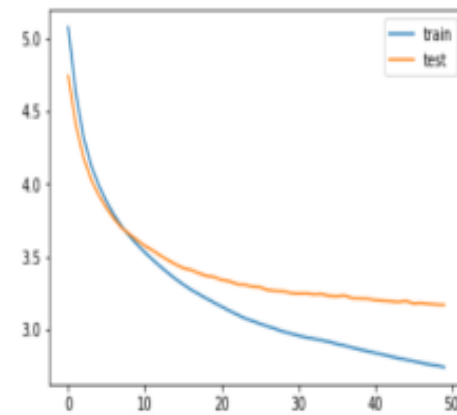
I. CONCLUSION

Therefore, Topic Models interfaces Words with comparable implications and recognizes words with various implications. Topic Models are successfully used to decipher the data from the whole assortment of documents. Latent Dirichlet Allocation Methodology is a graphical, probabilistic generative and measurable model for building a model of words appropriated inside topics and topics conveyed inside words.

II. REFERENCES

- [1] J. Yu and L. Qiu, "ULW-DMM: An effective topic modeling method for microblog short text," *IEEE Access*, vol. 7, pp. 884–893, Jan. 2019.
- [2] X. Zhang, X. Chen, Y. Chen, S. Wang, Z. Li, and J. Xia, "Topic Detection and Popularity Prediction in

Fig 5: Topic strength change charts



Microblogging," *Neurocomputing*, vol. 149, pp. 1469–1480, 2015.

- [3] Y. Zhang and A. Song, "Application of improved algorithm based on K-means in microblog topic discovery," *Comput. Syst. Appl.*, vol. 25, no. 10, pp. 3083–311, Dec. 2016.
- [4] L. Che and X. Yang, "News topic discovery model of multi feature fusion text clustering," *J. Nat. Univ. Defense Technol.*, vol. 39, no. 3, pp. 8590, Jan. 2017.
- [5] W.-Y. Bai, C. Zhang, K.-F. Xu, and Z.-M. Zhang, "A self-adaptive microblog topic tracking method by user relationship," *Acta Electronica Sinica*, vol. 45, no. 6, pp. 1375–1381, Jun. 2017.
- [6] S. Xu, "Topic model on image classification and their applications in high spatial resolution remote sensing image," Ph.D. dissertation, School Electron. Inf. Elect. Eng., Shanghai Jiao Tong Univ., Shanghai, China, Aug. 2012.
- [7] J. Liu, Y. Peng, L. Zhang, Y. Zhang, and J. Deng, "LDA-K-means algorithm of network food safety topic detection," *Eng. J. Wuhan Univ.*, vol. 50, no. 2, pp. 307–310, Apr. 2017.

- [8] X. Li, J. Zhang, and M. Yuan, "On topic evolution of a scientific journal based on LDA model," *J. Intell.*, vol. 33, no. 7, pp. 115–121, Jul. 2014.
- [9] X. Wang, L. Qi, C. Chen, J. Tang, and M. Jiang, "Grey System Theory based Prediction for Topic

- Trend on Internet," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 191–200, 2014.
- [10] T. He and H. Wang, "Evaluating perplexity of Chinese sentences based on grammar & semantics analysis," *Appl. Res. Comput.*, vol. 34, no. 12, pp. 3538–3542 and 3546, Jan. 2018.

Similarity Found: 4%

Date: Monday, April 12, 2021

Statistics: 110 words Plagiarized / 2495 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

INTERNET SOURCES:

- 1% - <https://www.sathyabama.ac.in/faculty/hod/drgarun-kumar-mech>
- < 1% - <https://www.ijarait.com/editions/volume-5-issue-3/>
- 1% - <https://silo.pub/handbook-of-natural-language-processing.html>
- < 1% - <https://www.coso.org/Pages/default.aspx>
- < 1% - https://www.researchgate.net/publication/2394181_Unsupervised_and_Supervised_Clustering_for_Topic_Tracking
- < 1% - https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation
- < 1% - <https://acp.copernicus.org/articles/18/15231/2018/>
- < 1% - <https://patents.google.com/patent/US6901399B1/en>
- < 1% - <https://academic.oup.com/sleep/article/42/12/zsz180/5549536>
- < 1% - <https://wenku.baidu.com/view/b468fadace2f0066f5332268.html>
- < 1% - <https://content.iopress.com/articles/journal-of-x-ray-science-and-technology/xst200715>
- 1% - <https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/>
- < 1% - https://www.researchgate.net/publication/346802781_Short-Text_Feature_Expansion_and_Classification_Based_on_Non-negative_Matrix_Factorization
- 1% - <https://link.springer.com/article/10.1007/s10479-020-03641-5>