IRE MAJOR PROJECTS – 2016 – SPRING

1. Text Processing Framework for Bahasa

Introduction

"Bahasa Indonesia" is the fourth most widely spoken language in the World and it is the official language of Indonesia. The framework that the team develops should include all the basic text processing algorithms for Bahasa such as, Stop word detection, Tokenization, Sentence Breaker, POS Tagging, Key Concepts Identification, Entity Recognition and Categorization.

Project Description

The framework should be for any one can to do some basic language processing in "Bahasa". It should be very similar to "Apache OpenNLP" . The team can also use tools that are already available in Bahasa and have all or some of those modules, and work on them improving the state-of-the-art.

The team should ideally do a thorough literature survey and understand the nuances of "Bahasa" language before they start implementing the project.

The modules in the framework should be

- **Stop Word detection**
- **Tokenization**
- **Sentence Breaker**
- **POS Tagging**
    - Tag a continuous text very similar to English POS tagging
- **Concept/Keyword Identification**
    - Use POS tagging or some other approaches to identify key concepts
- **Entity Recognition**
    - Identify people, locations, products, organizations, brands, money, health industry terminology (Zika virus, Pregnancy, Autism) etc.
- **Categorization**
    - Categorize an article into one of the following categories Politics, Crime, Entertainment, Sports, Business, Technology, Science, Health, Foods, Travel, Auto and Fashion. Politics can be treated as default category.

Dataset and Evaluation

Team will be provided with Bahasa news corpus. They can also make use of Wikipedia. Test data will be provided for Categorization, Entity Recognition. Other modules can use standard evaluation procedure.

2. Text Processing Framework for Indian Languages

(Potentially 9 teams)

Introduction

The goal of this project is to develop a Text Processing framework for nine Indian Languages (Hindi, Tamil, Telugu, Bengali, Malayalam, Kannada, Marathi, Gujarathi, Punjabi). The framework that the team develops should include all the basic text processing algorithms for Indian Languages such as, Stop word detection, Tokenization, Sentence Breaker, POS Tagging, Key Concepts Identification, Entity Recognition and Categorization.

Project Description

Indian languages are morphologically very rich. The challenge for any working in Indian Languages is to identify these subtle variations in writing. Hence, it becomes a very important task to identify these variations in a topic/entity and map it to single right one.

The team can also use tools that are already available and have all or some of those modules, and work on them improving the state-of-the-art.

The modules in the framework should be

- **Stop Word detection**
- **Tokenization**
- **Sentence Breaker**
- **Identify Variations (Highest Priority, Should not be confused with stemming)**
- **POS Tagging(Highest Priority)**
    - Tag a continuous text very similar to English POS tagging
- **Concept/Keyword Identification**
    - Use POS tagging or some other approaches to identify key concepts
- **Entity Recognition(Highest Priority)**
    - Identify people, locations, products, organizations, brands, money, health industry terminology (Zika virus, Pregnancy, Autism) etc.
- **Categorization(Highest Priority)**
    - Categorize an article into one of the following categories Politics, Crime, Entertainment, Sports, Business, Technology, Science, Health, Foods, Travel, Auto and Fashion. Politics can be treated as default category.

Dataset and Evaluation

Team will be provided with news corpus for Indian Languages. They can also make use of Wikipedia. Test data will be provided for Categorization, Entity Recognition, Variations. Other modules can use standard evaluation procedure.

3. ReadabilityMeasure for a given document in 10 languages

(Potentially 2 to 10 teams)

Introduction

The goal of this project is to develop a tool for one of the ten languages (Hindi, Tamil, Telugu, Bengali, Malayalam, Kannada, Marathi, Gujarathi, Punjabi and Bahasa) to measure the readability of a given document.

Project Description

Find out the Readability Level (School, Grad, Post Grad etc) and calculating the reading time metrics for a given document.

This is the description given for the terms on wordcounttools.com

What is the Readability level?
An estimated readability level measured by the Dale-Chall Score Formula.
What is the Estimated Reading Time?
Estimated based on an average reading time of 200 words per minute.

Heres the formula for the Reading Level

https://en.wikipedia.org/wiki/Dale%E2%80%93Chall_readability_formula

4. YouTube Video Extraction

Introduction

Today many news channels are in YouTube publishing videos every day covering all the important events of the day. Many times Videos present more interesting information and highly engaging for news readers rather than an article. The project is about extracting videos related to a news article from YouTube. The videos can be directly uploaded by the news publisher or a user sharing the video about that event. It should be done for English as well as Indian Languages news articles.

Project Description

The two key challenges

1. How to get videos around the article published time and rank them in terms of relevancy
2. How to fit it in Indian Language case, the common case is article will be in Indian Language, whereas video might be in English.

The team should work on building a query model from the article or a group of articles to query against YouTube API and use some metrics to rank the videos based on relevancy.

Dataset and Evaluation

Team will be provided with news corpus for English and Indian Languages which will have potential video links in YouTube. Optionally, we can provide the articles related to it. Evaluation can be done by fuzzy matching.

5. Web Page Parsing

Introduction

Web Pages contain a lot of elements, but there are only a few core elements that of importance to search engines/news app. In addition to the core elements, web pages contain a lot of navigational elements, templates and advertisements termed as boiler plate. The goal of this project is it identify the elements that aren't related to the main content blocks and need to be detected precisely. While it's easy for a human to easily distinguish the relevant content from the irrelevant page elements, automatic detection is always challenging problem, particularly at scale. The goal here is to develop an algorithm/features which can detect the main content elements with high precision.

Project Description

The algorithm should be developed for both English and Indian Languages, if needed team can also develop algorithms for each language separately. Compared to English, the challenges here are slightly different and complexity of the problem is less as Indian Local websites are few and have less boilerplate.

The major elements of importance and need to be extracted include the below.

- Title : Extract the correct title. (High Priority)
- Text : The main article content. (High Priority)
- Image : Most relevant image of the article. (High Priority)
- Video : Most relevant video of the article that belong to public domain like YouTube, Vimeo. (High Priority)
- Meta description : Meta description of the article.
- Meta keywords : Meta keywords of the article.
- Timestamp : The time when the article was published.
- Author : The author of the article.
- Credits : Any credits given in the article.

The algorithm should be web scalable i.e. should be very fast (milliseconds) and applicable to popular sources for each language. It should auto detect whether parse was successful or not with high precision.

Dataset and Evaluation

Team will be provided with news corpus for Indian Languages and English to work on. The articles will be from **popular sources** in each language. A separate test data set will be provided to test the algorithm with judgements.

6. Product Cataloging & Intelligence

Introduction:

This project is related to the area of information extraction .
Product comparison sites like http://www.compareraja.in/ scrape data from various e-commerce websites and provide user with a one-stop pricing details page . The goal is to similarly build a product database , extract vendor pricing information and perform analytics over the data .

Project Description:

1) Scrape product data from various product websites and store it in a particular schema
2) Extract vendor and pricing information of every product and store it in a particular schema
    a) also get non-portal information of vendor through google search etc .
3) Devise a mechanism to validate product information by extracting from multiple sources.
4) Perform analytics over product and vendor information.
    a) vendor location based analytics(Ex: who is the popular vendor for cricket bats in Hyderabad)
    b) vendor pricing analysis(Ex:average , median price of a product sold by various vendors ; what is the price offered by my competitor)etc..

Challenges:

1) How to escape “block scraping bot” mechanisms many popular product website

2) How to come up with a consistent format for storing information

Evaluation:

The project will be evaluated based on effectiveness of data extraction process and the kind of analytics performed over this data that would be useful for the consumer,vendor,manufacturer etc .

7) Dialog Engine for Product Information

Introduction :

A customer service provider answers questions posed users by going through product,pricing and personal information of which they have access to . Can this process be automated by a chat bot that extracts and stores information in a format useful for this task ?

Build a chat bot that would answer products-related questions of the users. Assume that we have product catalogue and user information(like history of purchases etc.) , then the bot should use the information from these sources and answer questions posed by users

Project Description :

1) Consider that we have a predefined set of general queries that can be posed by a user . Now given a query identify which category the query falls into . (Ex : Is it a comparison based query? Is it related to my past orders ?)

2) In case the user's query is incomplete come up with ways to elicit required information from the user .

Ex : User asks the questions : "What is the warranty period for a Videocon TV" . It will now prompt for the missing information, which is the Model Number / Code of the Videocon TV . Once the model number is inputted by the user, it will display the answer in the form of a canned reply "Warranty period for a Videocon TV Model XYZ is 2 years".

3) For each question type devise mechanisms to answer them . (Ex : if the query is about price about an product , look through the price property of a product in product catalogue)

Evaluation :

How effectively is the system answering auto generated property based questions(like screen size of a TV) or pose some FAQ questions in product websites .

8. USER PROFILING ENGINE

Introduction:

The goal is to predict whether the user (a session) is going to buy something or not, and if he is buying, what would be the items he is going to buy. Such an information is of high value to an e-business as it can indicate not only what items to suggest to the user but also how it can encourage the user to become a buyer. For instance to provide the user some dedicated promotions, discounts etc' .

Project Description :

Given a sequence of click events performed by some user during a typical session in an e-commerce website, the goal is to predict whether the user is going to buy something or not, and if he is buying, what would be the items he is going to buy.

Input :

The input that we would give to the problem is - A sequence of click events performed by some user during a typical session in an e-commerce website.

Output :

Is the user going to buy items in this session?
If yes, what are the items that are going to be bought?

Challenges :

1. No textual information is available. One needs to use the anonymized click through data in order to make predictions
2. Cold start problem
3. Sparse training data
4. Ongoing competition

9) Customer Review Analytics

Introduction :

Understand the sentiment of user reviews and provide useful information for the end-user as well as the product manufacturer regarding public opinion of the product .

Project Description :

1) Map a sentiment to Hierarchy of product-related objects(a review on samsung s3 mobile affects both “samsung s3” as well as its parent “samsung”)
2) Identifying review parameters . (Ex.a review such as “battery life is excellent” is related to the parameter “battery life”).
3) Contribution of review to overall sentiment based on parameter weights . (if a review parameter occurs in 100 out of 1000 reviews for a product then its weight would be 0.1. Now if a review has this parameter with negative sentiment of -1 then -0.1 is added to overall sentiment)
4) Extract the sentiment for each review parameter mentioned in the user review .

Such parameter-based sentiment for each product would help the manufacturer to understand if general public is unhappy with certain aspect of their product and hence can help to modify it accordingly (Ex: users may be unhappy with the battery life in iPhone 5s mobile).

It will also be useful information during product recommendation for a user looking to meet certain requirements . (Ex: user might be looking for mobiles with excellent battery life and we might accordingly provide appropriate products with good sentiment on battery life)

Evaluation :

Effectiveness of the system in predicting the parameter-based sentiments of various products .

10) Reverse Auctioning Engine :

Introduction :

Based On Human or Robot? competition on Kaggle . The goal of the competition is to identify auctioning agents that are software robots and thus eliminate them from the auction. Such robots have an undue advantage over human agents and their inability to win against robots may lead to plummeting core customer crowd .

Attempts at building a model to identify these bids using behavioral data, including bid frequency over short periods of time, has proven insufficient. Therefore come up with better ways to solve this problem .

Project Description :

There are two datasets in this competition. One is a bidder dataset that includes a list of bidder information, including their id, payment account, and address. The other is a bid dataset that includes 7.6 million bids on different auctions.

The online auction platform has a fixed increment of dollar amount for each bid, so it doesn't include an amount for each bid. You need to learn the bidding behavior from the time of the bids, the auction, or the device.

A training set of bidders is provided along with the respective outcome of whether they are robots or not and we need to model to differentiate between robot bidders and human bidders front this dataset .

Evaluation :

Model is used to predict the robot/human status of all bidders in the test dataset and the effectiveness of the model is evaluated using a measure such as Area Under the Curve .

11. Semantic Job Candidate Recommendation Engine

Problem Statement :
Consider the scenario where there is a database of resumes of various candidates and recruiter wants to gather the resumes that best suit his job requirement . It is obviously unfeasible to go through each resume and find out . A better choice would be to first filter these resumes based on some keywords and go through manually these filtered out resumes . There are many drawbacks in this approach . Firstly , people tend to use acronyms and words that are different but mean the same . Also , the skillset of the candidate is somewhat hidden in the kind of projects he has done in the past . For example , if a candidate has implemented some algorithm or worked in certain area of research that is related to natural language processing , one has to extract these underlying skill concepts so as to get good results upon searching for natural language processing as a keyword . In other words , one should semantically match the words mentioned by candidate in the resume to those mentioned in the job description and get the best candidates suitable for the job .

Approach :
Basically one has to know how job description and resume words are related by not just wordmatch but semantically . For this , one has to basically build a graph of words with weights indicating how well the words are related . One way to build such a graph is to extract paragraphs of text which point to only one/two underlying skill concepts and connect all the words that we come across in such paragraphs . Having built such a graph one has to devise ways to score the relatedness of words connected through edges in the graph .

12. Set expansion

Set expansion refers to expanding a partial set of “seed terms” into a more complete set. Given only a few seed entities, the goal is to discover other entities that belong to the same concept set.

Example: input seed terms: C++, java
Output: c++, java, perl, python, javascript, ..

Challenges:

1. How to handle word-based seed entities
2. How to handle phrase-based seed entities ?
3. Extend to a particular domain like education domain.
4. Develop your own algorithm for seed entities using available tools.
5. Develop two different algorithms for set expansion and compare and contrast both the algorithms.

References:

1. Entity List Completion Using Set Expansion Techniques
2. Identifying Sets of Related words from World Wide web.
3. A Cross-Lingual Dictionary for English Wikipedia Concepts

13. Community detection from research articles

Community detection is an important aspect in discovering the complex structure of social networks. The community detection involves grouping of similar users into clusters, where users in a group are strongly bonded with each other than the other members in the network. In this project, you need to work on DBLP or ACL dataset. There are different techniques for community detection. You need to compare and contrast several techniques in the due course.

Challenges:

1. Learning and applying different algorithms.
2. Identifying author similarity based on various factors like paper title, publication year.
3. Identify author similarity based on paper content, journal name, references.
4. Identify various features to detect user similarity.

References:

http://esatjournals.net/ijret/2013v02/i14/IJRET20130214017.pdf
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7022653

14. Gender Detection in Blogs

The goal of this project is, given a blog, you need to analyze the specific features in the text differentiating whether it is written by a male or a female. The features can be anything, for example, If a blog is about dresses, or cats then it may be written by a female, and if a blog is about sports, suits, etc then it would be written by a male. But in this project, you should also analyze the salient features which differentiate the text content and not merely on the topic of the text.

Reference:
http://www.ccse.kfupm.edu.sa/~ahmadsm/coe589-121/cheng2011-gender-identification.pdf

15. Sentiment analysis on Twitter data.

Input: Textual content of a tweet.
Output: Label signifying whether the tweet is positive, negative or neutral
Challenges:
- Noisy text - Misspellings, lack of grammar
"@user9 wassup wid u??"
- Tweets are short - lack of context
- Use of acronyms
"brb, afaik, etc."
- Open domain
"I had to wait for a long time" versus "Laptop X has a long battery life"
- Sarcasm
"Breaking News: Game of Thrones episode forces plane to make emergency landing #RedWedding"
- Negation handling
Dataset:
SemEval 2015 Task 10 subtask B provides both train and test sets of tweets along with labels here: http://alt.qcri.org/semeval2015/task10/index.php?id=data-and-tools

16. Named Entity Recognition on Twitter data.

Task: Named entity recognition is one of the first steps in most IE pipelines. The diverse and noisy style of user-generated social media text presents serious challenges, however. Performance still lags far behind than on formal text genres such as newswire. The goal of this shared evaluation is to promote research on NER in noisy text.

Dataset:

Register, downlad the dataset and build system for the shared task “Named Entity Recognition in Twitter" at the WNUT http://noisy-text.github.io/index.html#

Participant teams are provided with training and dev data in addition to a baseline system.

Challenges:

- You'll provided with dataset (Training and Testing) and a baseline system with baseline P, R and F.

So, evaluation will be based how much you beat the baseline.

- Create/annotate a 500 tweet test dataset to prove your results.

References:

- Task description - http://noisytext.github.io/index.html#
- Alan Ritter et al. Open Domain Event Extraction from Twitter KDD'12
- K. Gimpel et al . Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments

17. Aspect Based sentiment analysis

Description

Sentiment analysis is increasingly viewed as a vital task both from an academic and a commercial standpoint. The majority of current approaches, however, attempt to detect the overall polarity of a sentence, paragraph, or text span, regardless of the entities mentioned (e.g., laptops, restaurants) and their aspects (e.g., battery, screen; food, service). By contrast, this task is concerned with aspect based sentiment analysis (ABSA), where the goal is to identify the aspects of given target entities and the sentiment expressed towards each aspect. Datasets consisting of customer reviews with human authored annotations identifying the mentioned aspects of the target entities and the sentiment polarity of each aspect will be provided.

In particular, the task consists of the following subtasks:
Subtask 1: Aspect term extraction
Subtask 2: Aspect term polarity
Subtask 3: Aspect category detection
Subtask 4: Aspect category polarity

Datasets:

Two domain specific datasets for laptops and restaurants, consisting of over 6K sentences with fine grained Aspect level human annotations have been provided for training.

References:

- G. Ganu, N. Elhadad, and A. Marian, “Beyond the stars: Improving rating predictions using review text content”. Proceedings of the 12th International Workshop on the Web and Databases, Providence, Rhode Island, 2009.
- M. Hu and B. Liu, “Mining and summarizing customer reviews”. Proceedings of the 10th KDD, pp. 168–177, Seattle, WA, 2004.
- S.M. Kim and E. Hovy, “Extracting opinions, opinion holders, and topics expressed in online news media text”. Proceedings of the Workshop on Sentiment and Subjectivity in Text, pp. 1– 8, Sydney, Australia, 2006.
- B. Liu, Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. Morgan & Claypool, 2012.
- S. Moghaddam and M. Ester, “Opinion digger: an unsupervised opinion miner from unstructured product reviews”. Proceedings of the 19th CIKM, pp. 1825–1828, Toronto, ON, 2010.
- M. Tsytsarau and T. Palpanas. “Survey on mining subjective data on the web”. Data Mining and Knowledge Discovery, 24(3):478–514, 2012.
- Z. Zhai, B. Liu, H. Xu, and P. Jia. “Clustering product features for opinion mining”. Proceedings of the 4th International Conference of WSDM, pp. 347–354, Hong Kong, 2011.
- S. Brody and N. Elhadad. “An unsupervised aspectsentiment model for online reviews”. Proceedings of NAACL, pages 804–812, Los Angeles, CA, 2010.

18. Semantic Annotation of documents

Dataset : CS_network dataset
Challenge : Implement neural network to improve efficiency beyond what is achieved by baseline [2]

Project Description : Semantic Annoation of documents - To annotate a document with a Wikipedia article that matches its contents most closely.
Application : Use the results to find the category of a document, tag of document and similar applications.
Method : Use paragraph Vectors[1], Implement neural network to do topic modeling.
Evaluation : Report NDCG / MAP on dataset of your choice with topic modeling approaches like LDA, LSI as baseline.
Mentor office timings : Thursday 3.00 to 4.30pm
Code : Python only
[1] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In ICML, 2014.
[2] Cao et al. Neural Topic Models.

19. Medical Named Entity Recognition in Twitter

Description: The task of a Medical Name Entity Recognizer is to identify medical entities in text. Medical entities can be diseases, drugs, symptoms, etc. Previously, researchers in the field have used hand crafted features to identify medical entities in medical literature. In this work, we wish to extend medical entity recognition on tweets. Students enrolled in this project will be expected to use NLP toolkits designed for processing tweets along with other medical ontologies (or databases) to exploit a lot of semantic features for this task.

20. Formulate a scheme to measure the diversity of a summary through dense clustering algorithm

A summary should cover the relevant topics in the original corpus and
diverse enough.

Use different clustering algorithms to measure the diversity of a candidate summary and report the accuracies in terms of ROUGE score.

The scoring function of summarization should of the form

$$F(S) = \lambda * Coverage(S) - (1 - \lambda) * Redundancy(S)$$

where redundancy is penalized

Or

$$F(S) = \lambda * Coverage(S) + (1 - \lambda) * Diversity(S)$$

where diversity is rewarded.

If F is monotone submodular in nature ,a greedy approach can approximate the optimum summary by a percentage of 0.67 and rewarding the diversity of candidate summary helps to frame monotone submodular scoring function for candidate summary. Performing a clustering in the original corpus can help in deriving a better diversity measure for a candidate summary. The project is intended to Derive such a diversity measure and with clustering and report the improvement of summary in terms of ROUGE score for different clustering algorithms.

DatasetsUC 2004,DUC 2003(http://duc.nist.gov/data.html)

21. Derive a method to measure the Contextual Independence of a sentence

Contextual independence of a sentence can play a vital role in deciding it's generality.

Derive a method to measure the contextual independence of sentence using annotated corpus of contextually independent and dependent sentences

A sentence is produced to complete a local discourse and a document can be viewed as a sequence of local discourse units than sentences.
Each local discourse unit is a sequence of Contextually independent(CI) sentence followed by contextually dependent(CD) sentences.

eg: Scientists A and B discovered twin stars near the black hole in Andromeda galaxy. Also the life span of the black hole is determined by these scientists.

An annotated dataset of around 100 documents in which each sentence is annotated as CI/CD will be given.Students has to use a sequence labeling scheme and create a model for identifying the best possible sequence of CI and CD sentences in a test document. The probability of each sentence ton CI can be a measure to represents it's contextual independence.Local Discourse Unit can be utilized as a better linguistic unit with more topic information than a sentence as far as automated text summarization is concerned.

22. Humor Detection in Yelp reviews using Deep Learning

Project Description: Humor Detection is a difficult problem even for a human being. It is needless to say, how challenging such a task would be for machines. It is interesting to see how deep learning performs in capturing the higher order structures of humor, keeping in mind the sequential nature of reviews.

23. Integrating Network and Community Discovery for Mining Constant and Outlier Nodes

In real world, we often don't have access to the entire network of nodes and edges. This motivated researchers to propose various link prediction and community detection algorithms. A lot of work has been done on network discovery(link prediction) and community detection individually. However, recently, there has been an attempt[1] to integrate these two problems hoping that each of these would help the other.

During the network discovery and community detection process, it is possible that certain nodes would rapidly change the communities as we discover the network while certain nodes would remain constant. The goal is to find the characteristics of these nodes and understand why are they constant (or rapidly changing)

The following are the different stages of the project:
a) Understanding [1] end to end, demonstrated through a viva or presentation (10%)
b) Implementation of the algorithm and reproduce the results mentioned in [1] (25%+25%)
c) Proposing methods to detect constant nodes/rapidly changing nodes while detecting network and communities (20%)
d) Studying the characteristics of these nodes (20%)

References
[1] http://hanj.cs.illinois.edu/pdf/wsdm15_jliu.pdf

24. Reasoning over Knowledge Base

Knowledge Bases (KB) such as Google Knowledge Graph, WordNet, Yago are extremely useful resources for query expansion, coreference resolution, question answering (Apple's Siri), information retrieval or providing structured knowledge to users. There has been less number of works that focuses on developing algorithms to reason over existing KB.

In this project, we work on the goal of predicting the likely truth of additional facts based on existing facts in the KB. Such factual, common sense reasoning is available and useful to people. For instance, when told that a new species of monkeys has been discovered, a person does not need to find textual evidence to know that this new monkey, too, will have legs.
You will build a KB model that should accurately predict additional true facts using only an existing database. Now you can answer questions such as "Does a Bengal tiger have a tail?"

Reference:

1. http://nlp.stanford.edu/~socherr/SocherChenManningNg_NIPS2013.pdf

25. Understanding large social network

Recently, there has been an increasing attention to use Deep Learning (DL) techniques to analyze social graphs such as Flickr, Youtube, Twitter and so on. The beauty of such solutions is that once DL is applied, several network mining tasks such as node classification, link prediction, node visualization, node recommendation can be solved by conventional machine learning algorithms.

In this project, you will build a model that can capture the network information of a node in an efficient and scalable manner. You will get an opportunity to work with real world information networks such as Flickr, Youtube, Blogcatalog.

Reference:

1. http://arxiv.org/pdf/1403.6652.pdf
2. http://www.www2015.it/documents/proceedings/proceedings/p1067.pdf

26. Viewing text as a heterogeneous graph

How about viewing a sentence as a path of a graph, with words as its vertices? What about the heterogeneity? Will you accept if I call word, document and labels as three different types of nodes in a graph?

In this project, you will build a simple model that learns good feature vectors to represent each node in the above-mentioned graph. You will compete against powerful Deep Learning solutions for several text classification tasks.

Reference:

1. http://research.microsoft.com/pubs/255567/fp292-Tang.pdf

27. Algorithm name detection in computer science papers

In this project, you have to extract unique algorithm names from a big corpus of published computer science research / journal / conference articles. Then try to locate or identify the original work where it was first published. Finally categorize or cluster the different algorithm names into various sub-domains of computer science. You will use context based learning and semi-supervised learning techniques.

Tasks:

- Identify algorithm names in individual research papers, using modified *Named Entity Recognition* techniques,
- Filter out noise like *university names*, *places*, *author names*, names of *datasets* etc,
- Create a seed list of true positives and false positives for algorithm names,
- **Use algorithm name co-occurrence context information and sentence level context, information to get representations for each algorithm name,**
- **Given a new named entity, use some metric to identify whether this is an algorithm or not - use information from previous step to determine,**

Classify / *cluster* algorithms into several categories.

28. Learning Scientific scholar representations using a combination of collaboration, citation graph and text data

Here your aim would be to find good representations for authors who publish scientific research. The representations have to capture both textual data and graph data. These representations would then help categorize or cluster authors into various categories or perhaps predict future collaboration based on past data. You will use neural network based representation learning techniques similar to in word2vec to obtain them.

Tasks:

- Gather / crawl correct research paper dataset (*CiteseerX, arXiv, ACL, DBLP* etc.) which would contain all the necessary information needed (text + citation network),
- Identify duplicates from different sources and resolve them,
- Get all necessary context information like *co-authorship, citation based* etc. ,
- **Train a neural network based on above information using negative sampling,**
- **Refine both author and paper representations over iterations,**
- Evaluate the newly obtained author representations with both qualitative and quantitative tasks.

29. Wikipedia Document Classification

This project requires a comparison study of the various ways for text classification. You will start with baseline methods like tf-idf vectors and Naive Bayes classification, then move on to topic modeling based classification (LDA), then word2vec, doc2vec feature based classification (f1-score) and finally state of the art methods like skip-thought- vectors. Through this project you will learn about how the field of text classification have evolved over the last 10 years.

Tasks:

- The given dataset contains multiple tags per wiki page, create meaningful label for each,
- Perform basic text cleaning, stemming, stop-word etc. on full text,
- Start with the *td-idf* based vector space classification with *Naive Bayes* and *SVM*,
- Using Latent Semantic techniques (*LSI, LDA*), try to improve on above scores,
- Use word2vec/glove vectors per word and use document average for classification**,**
- **Modify previous step to use better methods than just averaging like in *doc2vec*,**

Try to achieve / beat state of the art performance.

Project No's	Mentors
1 – 5	SIEL PhDs
6 – 7	Ankur
8 – 10	Anurag
11	Soumyajit
12 – 13	Mrugani
14	Lokesh
15	Priyanka, Lokesh
16	Priyanka
17	Ankur
18	Priya
19	Nikhil
20 – 21	Litton
22	Satarupa
23	Ayushi
24 – 26	Ganesh
27 – 28	Soumyajit
29	Lokesh

Mentor Name	Mentor Contact Id
Anurag	anurag.tyagi@research.iiit.ac.in
Mrugani	mrugani.prasant@students.iiit.ac.in
Lokesh	lokesh.sayaji@students.iiit.ac.in
Priyanka	priyanka.bajaj@students.iiit.ac.in
Ankur	ankur.shrivastava@students.iiit.ac.in
Priya	priya.r@research.iiit.ac.in
Nikhil	nikhil.pattisapu@research.iiit.ac.in
Litton	litton.jkurisinkel@research.iiit.ac.in
Satarupa	satarupa.guha@research.iiit.ac.in
Ayushi	ayushi.dalmia@research.iiit.ac.in
Ganesh	ganesh.j@research.iiit.ac.in
Soumyajit	soumyajit.ganguly@research.iiit.ac.in