Fuzzy Systems and Soft Computing
ISSN : 1819-4362

SENTIMENT ANALYSIS OF INTERNET MEMES

Mr. GOBI. I Assistant Professor, Department of Information Technology, Sri Krishna Adithya College of Arts and Science, Coimbatore

Mr. SYED AADIL AHAMED. S I III B.Sc.IT, Department of Information Technology, Sri Krishna Adithya College of Arts and Science, Coimbatore

ABSTRACT:

Nowadays, the spread of Internet memes on online social media platforms such as Instagram, Facebook, Reddit, and Twitter is very fast. Analyzing the sentiment of memes can provide various useful insights. Meme sentiment classification is a new area of research that is not explored yet. Recently SemEval provides a dataset for meme sentiment classification. As this dataset is highly imbalanced, we extend this dataset by annotating new instances and use a sampling strategy to build a meme sentiment classifier. We propose a multi-modal framework for meme sentiment classification by utilizing textual and visual features of the meme. We found that for meme sentiment classification, only textual or only visual features are not sufficient. Our proposed framework utilizes textual as well as visual features together. We propose to use the attention mechanism to improve meme classification performance. Our proposed framework achieves macro F1 and accuracy of 34.23% and 50.02%, respectively. It increases the accuracy by 6.77% and 7.86% compared to only textual and visual features, respectively.

1. INTRODUCTION

The rapid growth of users on social media platforms leads to new ways of spreading information. Meme nowadays has become one of the most popular words for social media. A meme is an idea, the way in which a person behaves in response to a particular situation or a manner that spreads from one person to another within a culture. Spreading of memes on social media platforms such as Facebook, Instagram, Reddit, and Twitter is very fast.

Sentiment analysis is a growing field of Natural Language Processing (NLP), aiming to identify the polarity of opinion. Sentiment can be positive, negative or neutral. Sentiment analysis has a vast number of applications in real life, including the product's recommendation to a user based on opinions provided by other users, in political uses, etc.

Memes play an important role in handling various political battles or public relations on social media platforms. The most common practice in sentiment analysis is finding the sentiment of textual content crawled from Twitter, product reviews, hotel reviews, etc. Existing literature has mostly addressed the problem of sentiment analysis primarily using textual contents. But with the growing social media, users are expressing their opinions through text and the image.

Hence, researchers nowadays are also giving attention to sentiment analysis in multi-modal content. Spreading of memes is also very fast, but meme analysis is yet to be explored. Recently, SemEval-2020 proposed a task to detect the meme's polarity, which can fall into three predefined classes: positive, negative, or neutral. This is the very first attempt towards the meme sentiment analysis.

To analyze the sentiment of memes, the text-only approaches may not be sufficient. For example, consider the meme given in Figure 1, if the only textual content is considered, then the sentence 'FINALLY GETS JOB INTERVIEW' seems to have a neutral sentiment (no explicit positive words are used). However, if we also consider the visual information of meme, as shown in Figure 1, then we can say overall sentiment is positive. Hence, to analyze memes, both text and visual features have their own importance.

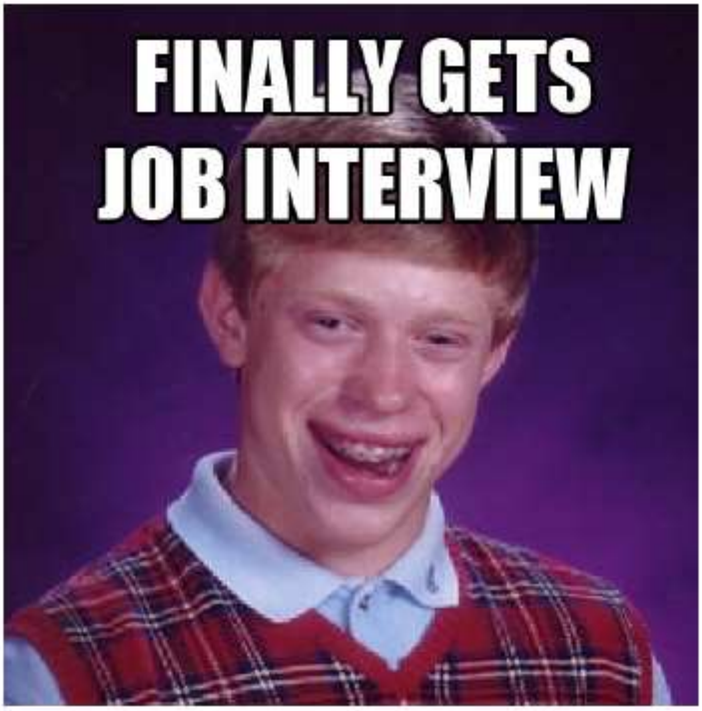


Figure 1: Meme example

2. SYSTEM STUDY

System analysis will be performed to determine if it is flexible to design information based on policies and plans of organization and on user requirements and to eliminate the weakness of present system. This chapter discusses the existing system, proposed system and highlights of the system requirements.

2.1 EXISTING SYSTEM

According to our survey the works that has been done in the area of memes and memetic images have been niche and only concentrated to few certain approaches like clustering the memes for detection and retrieving social impact, event-specific analysis, escalation and growth analysis. This is maybe due to the challenges that are faced while dealing with memes due to its wider context, diversity, and types (text, image, and video). French et al. proposed a system for using image-based memes as agents for predicting sentiments using the metadata like comments and caption on image. This survey depicts the trends in which the work on memes is progressing.

DISADVANTAGES OF EXISTING SYSTEM

- Above mentioned works are either for text or multi-modal content. Meme classification has not been explored much in detail.
- Computations are complex and time-consuming.
- Work on memes has been done on text-based content with two broad spectrums left relatively unexplored, viz., image- and video-based memes. This stands for our identified gap area to explore further.

2.2 PROPOSED SYSTEM

In this project, we work on the SemEval-2020 Task-8 dataset to detect the sentiment of memes. But this dataset is imbalanced. Hence we extend this dataset by adding more training instances for balancing purposes and then propose a multi-modal framework based on deep neural networks to classify the sentiment of the meme into one of the predefined classes, namely positive, negative, and neutral. We use a multi-modal framework with attention applied to both image and text to find out important regions and important words. Thereafter, to combine the image and textual modality, we use a fully connected layer that tries to find the relation between textual and visual features and finally produces a combined feature vector. We evaluate the proposed approach using accuracy and Macro F1 score on the test set of the SemEval-2020 dataset. We get the macro F1 of 34.23%, and accuracy of 50.02%, respectively, which is higher than the SemEval baseline, i.e., Macro F1 of 0.21%.

Contributions:

- We develop a multi-modal neural network that learns from the two modalities, viz. Textual and visual. For text modality, our model takes as input the embedding representation of each word present in the OCR extracted text.
- Further, we use Convolutional Neural Network (CNN) to learn textual features, and then we apply attention to the output of CNN to extract the most relevant features for classification.
- We use the pretrained model VGGNet to extract the visual features for image modality, and then we apply attention to the extracted features to detect important visual features for classification. Finally, both the features are fused with the help of a fully connected layer. The overall architecture of our proposed model is shown in Figure 2.

Advantages:

- Our designed neural network is simpler than pre-trained networks, and it is possible to run it on conventional modern personal computers. This is possible because the algorithm requires many less resources for both training and implementation.
- The importance of developing smaller networks is also linked to the possibility of deploying the algorithm on mobile platforms, which is significant for diagnostics in developing countries.
- In addition, the network has a very good execution speed of 15 ms per image.
- Our proposed system outperforms the other baselines, which indicates that multi-modal information actually helps to improve the effectiveness of the system.

3. SYSTEM SPECIFICATION

3.1 HARDWARE SPECIFICATION

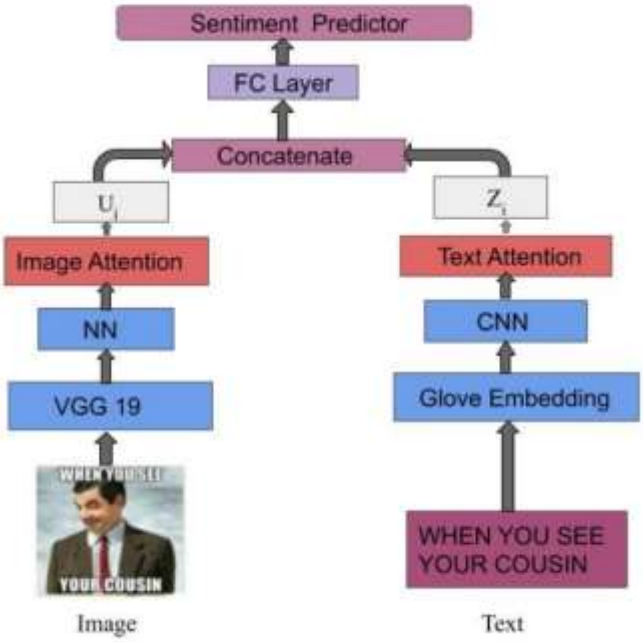
Processor	:	i3 and above
RAM	:	2 GB and above
HDD	:	500GB and Above

3.2 SOFTWARE SPECIFICATION

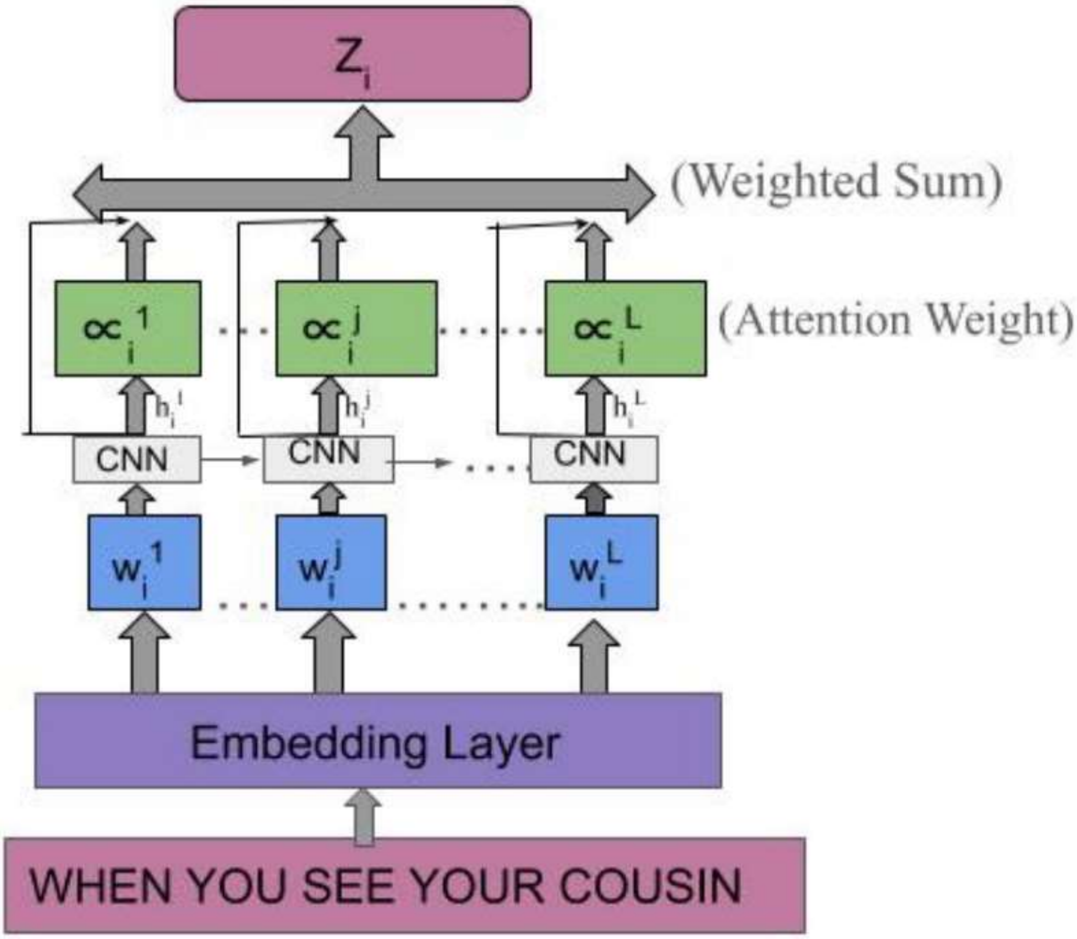
Operating System	: Windows XP/8/10
Programming Language (Front End)	: Python
Back End	: Data Set (SemEval)
IDE	: Anaconda 2020

4. SYSTEM ANALYSIS AND DESIGN

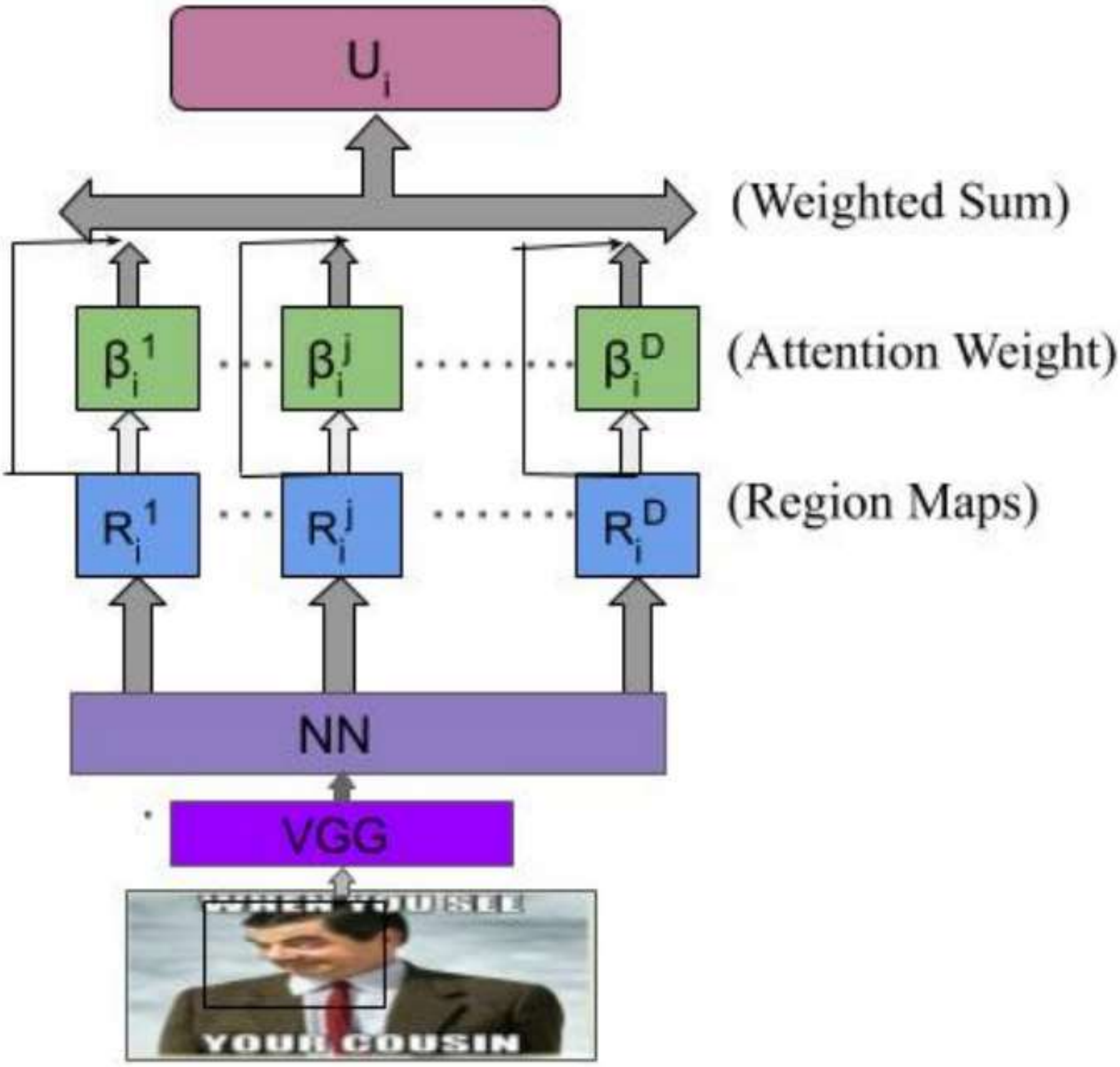
Architecture diagram


Sentiment Predictor
FC Layer
Concatenate
U_i
Z_i
Image Attention
Text Attention
NN
CNN
VGG 19
Glove Embedding
WHEN YOU SEE
YOUR COUSIN
WHEN YOU SEE YOUR COUSIN
Image
Text

DFD: For Text


Z_i
(Weighted Sum)
∝_i^1
∝_i^j
∝_i^L
(Attention Weight)
h_i^l
h_i^j
h_i^L
CNN
CNN
CNN
w_i^1
w_i^j
w_i^L
Embedding Layer
WHEN YOU SEE YOUR COUSIN

DFD: For Image



4.2 MODULE DESIGN

4.2.1 Textual features

In this section, we discuss the textual features, how they are given as input to our system, and how to apply attention to the features extracted.

(i) Embedding Layer

The embedding layer takes the input as a sequence of words present in the sentence. For each word w present in the sentence, a lookup matrix is created to obtain its embedding representation. Lookup matrix can be initialized using pretrained word embedding vectors. In our work, the pre-trained vector representations provided by Glove are used. It captures syntactic and semantic relations among the words. The embedding of each word w is then given as an input to the CNN to learn the text representation.

(ii) Convolutional Neural Network (CNN)

The CNN automatically learns the features with the help of convolutional filters. Convolutional filters capture the semantic and syntactic features of a given sentence. CNN has been used in a wide variety of tasks. The CNN consists of convolutional layers. Convolutional layers are followed by non-linear layers that contain the Relu activation function, followed by the pooling layers. For our task, we use 3 convolutional layers. The three convolutional layers contain 128 filters of sizes 2, 3, and 4 each. Word embedding vectors of a sentence are given as input to CNN to learn the n-gram features.

(iii) Attention for text

In NLP related tasks, some words in the sentence are more important for the task compared to the other words in the same sentence. To capture this phenomenon, attention model for the text has been proven beneficial for many NLP related tasks i.e., text summarization, machine translation, textual sentiment analysis, etc. Attention models calculate the attention score which lies in the range of 0 and 1.

4.2.2 Visual Features

The image with size 224*224 is used as the input to the pre-trained model VGG-19 to extract features of the image. We use the output of conv5*4 layer of VGG-19 as the region features which consist of 196 regions, and each region is represented in 512 dimensions. Thus, region features are having dimensions of (196*512). The output of VGG-19 is further passed to a dense layer that has 250 hidden neurons. The output of this dense layer is passed to the attention layer to find out the important regions for classification.

(i) Attention for Image

Image attention has been proven to be beneficial for many vision-related tasks. We apply the attention over the image regions (output of dense layer) to find out the most important regions. Attention score is calculated for each region, signifying the region importance. It lies in the range between 0 and 1.

(ii) Fusion of Text and Image features

Finally, the attended image features vector and text features vector are passed to a fully connected layer containing hidden neurons. This layer tries to find out the relation between image and text features and finally combines both.

4.2.3 Output Layer

The output of dense layer, i.e., the combined feature vector of image and text is finally passed to the output layer, which contains softmax as an activation function. The output layer maps the combined feature vector to a probability score. This probability score helps to classify the tweet into one of its predefined categories.

4.3 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:’

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

4.4 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

4.5 DATABASE DESIGN

Databases are normally implemented by using a package called a Data Base Management System (DBMS). Each particular DBMS has somewhat unique characteristics, and so such, general techniques for the design of database are limited. One of the most useful methods of analyzing the data required by the system for the data dictionary has developed from research into relational database, particularly the work of E.F.Codd. This method of analyzing data is called “Normalization”. Unnormalized data are converted into normalized data by three stages. Each stage has a procedure to follow.

NORMALIZATION:

The first stage is normalization is to reduce the data to its first normal form, by removing repeating items showing them as separate records but including in them the key fields of the original record.

The next stage of reduction to the second normal form is to check that the record, which one is first normal form, all the items in each record are entirely dependent on the key of the record. If a data item is not dependent on the key of the record, but on the other data item, then it is removed with its key to form another record. This is done until each record contains data items, which are entirely dependent on the key of their record.

The final stage of the analysis, the reduction of third normal form involves examining each record, which one is in second normal form to see whether any items are mutually dependent. If there are any item there are removed to a separate record leaving one of the items behind in the original record and using that as the key in the newly created record.

BUSINESS MODELING:

The information flow among business function is modeled in a way that answers the following questions: what information drives the business process? What information is generated? What generate it? Where does the information go? Who process it?

DATA MODELING:

The information flow defined as a process of the business modeling is refined into a set of data objects that are needed to support the business. The characteristics (called attributes) of each object are identified and relationships between these objects are defined.

PROCESS MODELING:

The data objects defined in the data-modeling phase are transformed to achieve the information flow necessary to implement a business function. Processing description is created for addition, modifying, deleting, or retrieving a data object.

5. SYSTEM TESTING

Software testing is a critical element if software quality assurance represents the ultimate reviews of specification, design and coding. Testing is vital of the system.

Errors can be injected at any stage during development. During testing, the program is executed with correctness. A series of testing are performed for the proposed systems before the system is delivered to the user.

5.1 Unit Testing

In the unit testing the testing is performed on each module and this module is known as module testing. This testing was carried out during programming state itself. In this testing all the modules working satisfactorily as regard to the expected output from the module.Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual function or procedure. Unit tests are created by programmers or occasionally by white box testers.

Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development. Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit API.

5.2 Acceptance Testing

Acceptance testing is black-box testing performed on a system (e.g. software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery. It is also known as functional testing, black-box testing, release acceptance, QA testing, application testing, confidence testing, final testing, validation testing, or factory acceptance testing.

Acceptance testing generally involves running a suite of tests on the completed system. Each individual test, known as a case, exercises a particular operating condition of the user's environment or feature of the system, and will result in a pass or fail, or Boolean, outcome. There is generally no degree of success or failure. The test environment is usually designed to be identical, or as close as possible, to the anticipated user's environment, including extremes of such. These test cases must each be accompanied by test case input data or a formal description of the operational activities (or both) to be performed—intended to thoroughly exercise the specific case—and a formal description of the expected results.

5.2.1 Types of Acceptance Testing

Typical types of acceptance testing include the following

User acceptance testing

This may include factory acceptance testing, i.e. the testing done by factory users before the factory is moved to its own site, after which site acceptance testing may be performed by the users at the site.

Operational acceptance testing

Also known as operational readiness testing, this refers to the checking done to a system to ensure that processes and procedures are in place to allow the system to be used and maintained.

Contract and regulation acceptance testing

In contract acceptance testing, a system is tested against acceptance criteria as documented in a contract, before the system is accepted. In regulation acceptance testing, a system is tested to ensure it meets governmental, legal and safety standards.

Alpha and beta testing

Alpha testing takes place at developers' sites, and involves testing of the operational system by internal staff, before it is released to external customers. Beta testing takes place at customers' sites, and involves testing by a group of customers who use the system at their own locations and provide feedback, before the system is released to other customers. The latter is often called “field testing”.

5.3 Integration Testing

One module can have adverse effect on another such functions when combined may not produce the desired results. Integration testing is a systematic technique for constructing the program structure and conducting test to uncover errors associated with interface. All the modules are combined in this testing step. The entire program is tested as the whole. The errors uncovered are corrected for the next testing step.

5.4 Black Box Testing

The black box approach is attesting method in which test data are delivered from the functional requirement without regard to the final program structure. Because only functionality of the software is concerned.

In black box testing, only the functionality is determined by observing the outputs to the corresponding input. In this testing various input images are exercised and the output images are compared as required by the content retriever.

5.5 White Box Testing

White box testing are the software predicates on close examination of procedure details. It provides test cases that exercise specific test for conditions and loops. White box testing was carried out in the order to guarantee that

- All independent parts within a module exercised at least once.
- All logical decision on this true and false side was exercised

5.6 Validation Testing

Computer input procedures are designed to detect errors in the data at the lower level of detail which is beyond the capability of the control procedures. The validation succeeds when the software functions in the manner that can be reasonably expected by the customer.

6. IMPLEMENTATION

The implementation phase focuses how the engineer attempts to develop the system. It also deals with how data are to be structured, how procedural details are to be implemented, how interfaces are characterized, how the design will be translated into programming and hoe the testing will be performed. The methods applied during the development phase will vary but three specific technical tasks should always occur.

- The software design
- Code generation
- Software testing

The system group has changed with responsibility to develop a new system to meet requirements and design and development of new information system. The source of these study facts is variety of users at all level throughout the organization.

6.1 Stage of Development of a System

- Feasibility assessment
- Requirement analysis
- External assessment
- Architectural design
- Detailed design

- Coding
- Debugging
- Maintenance

6.2 Feasibility Assessment

In Feasibility this stage problem was defined. Criteria for choosing solution were developed, proposed possible solution, estimated costs and benefits of the system and recommended the course of action to be taken.

6.3 Requirement Analysis

During requirement analysis high-level requirement like the capabilities of the system must provide in order to solve a problem. Function requirements, performance requirements for the hardware specified during the initial planning were elaborated and made more specific in order to characterize features and the proposed system will incorporate.

6.4 External Design

External design of any software development involves conceiving, planning out and specifying the externally observable characteristic of the software product. These characteristics include user displays, report formats, external data source and data links and the functional characteristics.

6.5 Internal Design Architectural and Detailed Design

Internal design involved conceiving, planning out and specifying the internal structure and processing details in order to record the design decisions and to be able to indicate why certain alternations were chosen in preference to others. These phases also include elaboration of the test plans and provide blue prints of implementation, testing and maintenance activities. The product of internal design is architectural structure specification.

The work products of internal design are architectural structure specification, the details of the algorithm, data structure and test plan. In architectural design the conceptual view is refined.

6.6 Detailed Design

Detailed design involved specifying the algorithmic details concerned with data representation, interconnections among data structures and packaging of the software product. This phase emphasizes more on semantic issues and less synthetic details.

6.7 Coding

This phase involves actual programming, i.e, transacting detailed design into source code using appropriate programming language.

6.8 Debugging

This stage was related with removing errors from programs and making them completely error free.

6.9 Maintenance

During this stage the systems are loaded and put into use. They also get modified accordingly to the requirements of the user. These modifications included making enhancements to system and removing problems.

7. CONCLUSION AND FUTURE ENHANCEMENT

In this project, we have proposed a multi-modal framework for meme sentiment classification by utilizing textual and visual information of memes. We use the SemEval-2020 task data and also annotated our own dataset to make this dataset balanced. We found that only textual information or only visual information is not sufficient to analyze a meme's sentiment. Our proposed framework utilizes textual and visual features and finally fuses both the information through a fully connected layer. Our proposed framework achieved the macro F1 and accuracy of 34.23% and 50.02%, respectively. Our proposed framework increases the accuracy by 6.77% and 7.86% compared to only textual and visual features, respectively.

SCOPE FOR FUTHURE ENHANCEMENT

In the future, we are planning to explore other fusion methods to incorporate textual and visual features. We would also explore contextual embeddings for the text part of meme classifications.

8. BIBLIOGRAPHY

1. Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca J Passonneau. 2011. Sentiment analysis of twitter data. In Proceedings of the workshop on language in social media (LSM 2011), pages 30–38.
2. Md Shad Akhtar, Asif Ekbal, and Erik Cambria. 2020. How intense are you? predicting intensities of emotions and sentiments using stacked ensemble. IEEE Computational Intelligence Magazine, 15(1):64–75.
3. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
4. Akshat Bakliwal, Jennifer Foster, Jennifer van der Puil, Ron O'Brien, Lamia Tounsi, and Mark Hughes. 2013. Sentiment analysis of political tweets: Towards an accurate classifier. Association for Computational Linguistics.
5. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:P10008.
6. Hongshen Chen, Yue Zhang, and Qun Liu. 2016. Neural network for heterogeneous annotations. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 731–741.
7. Very Large Corpora. 2000. Empirical methods in natural language processing.
8. Deepak Chowdary Edara, Lakshmi Prasanna Vanukuri, Venkatramaphanikumar Sistla, and Venkata Krishna Kishore Kolli. 2019. Sentiment analysis and text categorization of cancer medical records with lstm. Journal of Ambient Intelligence and Humanized Computing, pages 1–17.
9. Manoochehr Ghiassi and S Lee. 2018. A domain transferable lexicon set for twitter sentiment analysis using a supervised machine learning approach. Expert Systems with Applications, 106:197–216.
10. Douglas M Hawkins. 2004. The problem of overfitting. Journal of chemical information and computer sciences, 44(1):1–12.
11. Tao Jiang, Jiahai Wang, Zhiyue Liu, and Yingbiao Ling. 2020. Fusion-extraction network for multimodal sentiment analysis. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 785–797. Springer.
12. Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.