International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 885-889 www.ijaem.net ISSN: 2395-5252

Natural Language Processing Based Text Emotion Recognition Emotion Analysis for Cyber Bullying

K. Ramesh, M.E.,[1] Asst Professor, B.Vidhya,[2] K.Tamizh Selvi,[3] D.Sharmila,[4] A.Santhiya[6]
Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur.

Submitted: 05-05-2021 Revised: 17-05-2021 Accepted: 20-05-2021

ABSTRACT: Textual emotion detection has a high impact on business, society, politics or education with applications such as, detecting depression or personality traits, suicide prevention or identifying cases of cyber-bulling. Given this context, the objective of our research is to contribute to the improvement of emotion recognition task through an automatic technique focused on reducing both the time and cost needed to develop emotion corpora. Our proposal is to exploit a bootstrapping approach based on intentional learning for automatic annotations with two main steps: An initial similarity-based categorization where a set of seed sentences is created and extended by distributional semantic similarity (word vectors or word embedding). Train a supervised classifier on the initially categorized set. The technique proposed allows us an efficient annotation of a large amount of emotion data with standards of reliability according to the evaluation results. The social networking sites dispense their data conveniently and freely on the web. This availability of data entices the interest of young researchers to plunge them in the field of sentiment analysis. People express their emotions and perspectives on the social media discussion forums. The business organizations employ researchers to investigate the unrevealed facts about their products and services. Spontaneous and automatic determination of sentiments from reviews is the main concern of multinational organizations the machine learning techniques have improved accuracy of sentiment analysis and expedite automatic evaluation of data these days. This work attempted to utilize four machine learning techniques for the task of sentiment analysis.

I. INTRODUCTION

BACKGROUND

AUTOMATIC detection of affective states in text is becoming more and more important due to the fact that it has the potential of bringing substantial benefits for different sectors. Example of this can be for instance the applications in e-learning environment; suicide prevention depression detection identification cases of cyberbullying or tracking well-being This paper is focused on textual emotion detection because it is one of the main media employed to interact with humans through chats room, public reviews, emails, social networks or web blogs. This produces an abundance volume of information and hence requires a computer processing to classify automatically the text in accordance with its emotional degree and orientation.

AFFECTIVE COMPUTING

Affective Computing is also the title of a textbook on the subject by Rosalind Picard. Affective computing is the study and development of systems and devices that can recognize, interpret, process, and simulate human affects. It is an interdisciplinary field spanning computer science, psychology, and cognitive science. While some core ideas in the field may be traced as far back as to early philosophical inquiries into emotion, the more modern branch of computer science originated with Rosalind Picard 1995 paperon affective computing and her book Affective Computingpublished by MIT Press. One of the motivations for the research is the ability to give machines emotional intelligence, including to simulate empathy. The machine should interpret the emotional state of humans and adapt its behavior to them, giving an appropriate response to those emotions.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 885-889 www.ijaem.net ISSN: 2395-5252

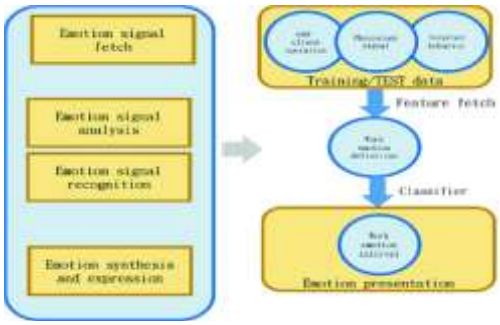


Fig: 1.1 Affective computing

II. LITERATURE SURVEY

2.1 TITLE: LSHWA: IMPROVING SIMILARITY-BASED WORD EMBEDDING WITH LOCALITY SENSITIVE HASHING FOR CYBERBULLYING DETECTION, AUTHOR: ZEHUA ZHAO, YEAR: 2020

Description:

Word embedding methods use low-dimensional vectors to represent words in the corpus. Such low-dimensional vectors can capture lexical semantics and greatly improve the cyberbullying detection performance. However, existing word embedding methods have a major limitation in cyberbullying detection task: they cannot represent well on "deliberately obfuscated words", which are used by users to replace bullying words in order to evade detection. These deliberately obfuscated words are often regarded as "rare words" with a little contextual information and are removed during preprocessing. In this paper, we propose a word embedding method called LSHWE to solve this limitation, which is based on an idea that deliberately obfuscated words have a high context similarity with their corresponding bullying words. LSHWE has two steps: firstly, it generates the nearest neighbor matrix according to the cooccurrence matrix and the nearest neighbor list obtained by Locality Sensitive Hashing (LSH); secondly, it uses an LSH-based autoencoder to learn word representations based on these two matrices. Especially, the reconstructed nearest neighbor matrix generated by the LSH-based autoencoder is used to make the representations of deliberately obfuscated words close to their corresponding bullying words. In order to improve the algorithm efficiency, LSHWE uses LSH to generate the nearest neighbor list and the reconstructed nearest neighbor list. Empirical experiments prove the effectiveness of LSHWE in cyberbullying detection, particularly on the "deliberately obfuscated words" problem. Moreover, LSHWE is highly efficient, it can represent tens of thousands of words in a few minutes on a typical single machine.

2.2 Title: Knowledge-Based Biomedical Wordsense Disambiguation With Neural Concept Embeddings, Author: Akm Sabbir, Year : 2017

Description:

Biomedical word sense disambiguation (WSD) is an important intermediate task in many natural language processing applications such as named entity recognition, syntactic parsing, and relation extraction. In this paper, we employ knowledgebased approaches that also exploit recent advances in neural word/concept embeddings to improve over the state-of-the-art in biomedical WSD using the public MSH WSD dataset as the test set. Our methods involve weak supervision – we do not use any hand-labeled examples for WSD to build our prediction models; however, we employ an existing concept mapping program, MetaMap, to obtain our concept vectors. Over the MSH WSD dataset, our linear time (in terms of numbers of senses and words in the test instance) method achieves an accuracy of 92.24% which is a 3% improvement over the best known results obtained via unsupervised means. A more expensive approach that we developed relies on a nearest neighbor framework and achieves accuracy of 94.34%, essentially cutting the error rate in half. Employing dense vector representations learned from unlabeled free text has been shown to benefit many language processing tasks recently and our efforts show that biomedical WSD is no exception to this trend. For a complex and rapidly evolving domain such as biomedicine, building labeled datasets for larger sets of ambiguous terms may be impractical. Here, we show that weak supervision that leverages recent advances in representation learning can rival supervised approaches in biomedical WSD. However, external knowledge bases (here sense inventories) play a key role in the improvements achieved.

III. SYSTEM ANALYSIS

EXISTING SYSTEM

- In this project the state of the art of different aspects related to our approach.
- On the one hand, bootstrapping technique and semantic similarity metrics are analyzed since both are the pillars of our approach.
- On the other hand, an exhaustive review of emotion lexicons and corpora is carried out with the aim of obtaining conclusions and determining the pending issues.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 885-889 www.ijaem.net ISSN: 2395-5252

- We adding and updating some processes to give the output as high accuracy.

DISADVANTAGE

- High Cost
- Less Efficiency
- Data Loss
- Privacy Issues

PROPOSED SYSTEM

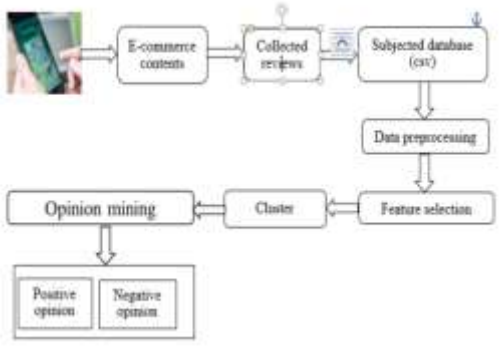
- Identifies the features of a cybercrime incident and their potential elements and provides a two-level offence classification system based on specific criteria.
- The proposed schema can be extended with a list of recommended actions, corresponding measures and effective policies that counteract the offence type and subsequently the particular incident.
- In proposed system sentiment similarity analysis has been implemented using machine learning un supervised and supervised algorithm.
- In this method modeled the sentiment classification problems as learning sentiment specific word embedded issue and designed three neural network to effectively incorporate the super vision from text data with sentiment labels.

ADVANTAGE

- High accurate
- More efficiency
- Reliability

SYSTEM ARCHITECTURE

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system**.**



MODULE IMPLEMENTATION

1. REGISTRATION

- The registration module allow the user and data owner to create login username and the password by submitting their information like mail id, phone number, name, etc.
- By registering the network or cloud the user can gain access to the resources stored in the cloud.

2. LOGIN

- In this module the user can login by using their unique username and password.
- The login module verify the user given username and password with the stored username and password in the cloud.
- If the username and password is matched the user can access the resources.
- If it does not match the user does not allowed to access the resource.

3. ADMIN

- Admin module allows system administrator to set up back-end of the system and perform basic system configuration, mainly definition of predefined drop-down fields, definition of classes time schedule, etc.
- All the new packages and promo bundles as well as new prices and price types for classes, new subjects offered, etc.

IV. CLASSIFICATION

- Classification is a data mining function that assigns items in a collection to target categories or classes.
- The goal of classification is to accurately predict the target class for each case in the data.
- For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

V. FEATURE EXTRACTION

- Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.
- A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process.

SOFTWARE ENVIRONMENT

THE JAVA PROGRAMMING LANGUAGE

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 885-889 www.ijaem.net ISSN: 2395-5252

- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

VI. SYSTEM DESIGN

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Goals:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

VII. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

1. Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system

configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

2. Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

VIII. CONCLUSION

Machine learning approaches have been so far good in delivering accurate results. Depending upon the application, the success of any approach will vary. Hybrid approach has so far displayed positive sentiment as far as performance is concerned. Though they have been deployed using unigrams and diagrams, their performance is worse on trigrams. This definitely leaves researchers to explore the terrain.

IX. FUTURE WORK

Our future research will deal with 1) exploiting larger emotion lexicons than Emolex for creating the seed such as EmoSenticNet or Depeche Mood since the core of IL is the first step; 2) analyzing in depth the use of DSMs and testing the approach with domain specific embedding; 3) testing the technique in more corpora with another group of emotions, since the adaptation of the process is really simple provided that the emotion lexicon was annotated with the desired emotions; and 4) analyzing the usability of other resources to extend the seed in the enriched approaches.

REFERENCE

[1]. F. Vaassen, "Measuring emotion," Ph.D. dissertation, Universiteit Antwerpen, 2014. [5] C. Cherry, S. M. Mohammad, and B. De Bruijn, "Binary Classifiers and Latent Sequence Models for Emotion Detection in Suicide Notes," Biomedical informatics insights, vol. 5(Suppl 1), pp. 147–154, 2012.

[2]. M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, "Improving cyberbullying detection with user context," in Advances in Information Retrieval, 2013, pp. 693–696.

[3]. H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, R. E. Lucas, M. Agrawal, G. J. Park, S. K. Lakshmikanth, S. Jha, M. E. P. Seligman, and L. Ungar, "Characterizing Geographic Variation in Well-Being Using Tweets," in Proceedings of the International AAAI Conference on Weblogs and Social Media, 2013.

[4]. S. M. Mohammad, "Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text," in Emotion Measurement, H. Meiselman, Ed. Elsevier, 2016.

[5]. S. Aman and S. Szpakowicz, "Identifying Expressions of Emotion in Text," in Text, Speech and Dialogue, 2007, pp. 196–205.