IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 6, June 2020

"Sentence Similarity"

Mrs. Chethana C[1]

Assistant Professor, Department of CSE, BMSIT&M, Bangalore, India[1]

Abstract: Sentiment Analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral. In this paper by using Python to find the similarity of sentences in Hindi and English using word- embeddings is discussed.

Keywords: Artificial Intelligence, Machine learning, Sentiment Analysis, Similarity.

I. INTRODUCTION

1.1 Machine learning

Machine learning is functionality that helps software perform a task without explicit programming or rules. It is considered as a subcategory of artificial intelligence. Machine learning involves statistical techniques, such as deep learning. Machine Learning is the idea that the computer does not just use a pre-written algorithm, but learns how to solve the problem itself. Or, to explain it in other words, there is an excellent definition by Arthur Samuel – "Machine Learning is a field of study that gives computers the ability to learn without being explicitly programmed". Machine Learning teaches a machine to solve various complex tasks that are difficult to be solved algorithmically. For example, it can be used for face recognition on our phone or voice understanding, driving a car (Google), diagnose diseases by symptoms (Watson), recommend – books (Amazon), movies (Netflix), music (Spotify) etc. In traditional programming hard coding of the behaviour of the program is done [7].

1.2 Natural Language Processing

Natural Language Processing is the driving force behind the following common applications:

1. Language translation applications such as Google Translate
2. Word Processors such as Microsoft Word and grammarly that employ NLP to check grammatical accuracy of texts.
3. Interactive Voice Response (IVR) applications used in call centers to respond to certain users' requests.
4. Personal assistant applications such as OK Google, Siri, Cortana, and Alexa.

1.3 Artificial Intelligence

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving. The ideal characteristic of artificial intelligence is its ability to rationalize and take actions that have the best chance of achieving a specific goal. Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software. The study of natural language processing has been around for more than 50 years and grew out of the field of linguistics with the rise of computers.

II. LITERATURE SURVEY

LIBLINEAR is an open source library for large scale linear classification. It supports logistic regression and linear support vector machines. Easy-To-Use-Command –Line Tools and Library calls for users and developers are provided [1]. M. Faruqui, et.al. "Community Evaluation and Exchange of Word Vectors", this paper presents a demo system that supports rapid and consistant evaluation of word vector representation on a variety of task, visualisation and comparison of different word vector representation [2]. L.Finkelstein, et.al,"Placing search in context: The concept revisited", this paper describes a novel algorithm and system for processing queries in their context [3]. Bin Gao, Jiang Bian, and Tie-Yan Liu, "WordRep: A benchmark for research on learning word representations", describes the details of the WordRep collection and shows how to use it in different types of machine learning research related to word embedding [4]. Remi Lebret and Ronan Collobert, "Word embeddings through Hellinger PCA", proposes to drastically simplify the word embeddings computation through a Hellinger PCA of the word co-occurence matrix [5]. Omer Levy, Yoav Goldberg, and Ido Dagan, "Improving distributional similarity with lessons learned from word embeddings", observes mostly local or insignificant performance differences between the methods, with no global advantage to any single approach over the others.

III. TOOLS & TECHNOLOGIES

- **Gensim**: Gensim is a Python library for topic modelling, document indexing and similarity retrieval with large corpora. Target audience is the Natural Language Processing (NLP) and Information Retrieval (IR) community.
- **Scikit-learn**: **Scikit-learn** (formerly **scikits.learn** and also known as **sklearn**) is a free software machine learning library for the Python programming language.[3] It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
- **NLTK**: NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.
- **Flask**: Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions
- **Pandas**: Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.
- **Jupyter**: The Jupyter Notebook is an open-source web application that allows us to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

IV. PROJECT REQUIREMENTS

Software Requirements

- Editor/IDE: Jupyter-Lab, Visual studio code, Pycharm
- Programming Language: Python
- Framework: Docker, Flask
- API: Concerto API
- Packages: Gensim, Numpy, Flask, scikit-learn, nltk, Pandas, Jupyter

Hardware Requirements

- Processor: Intel Pentium Core 2 Duo (or greater)
- RAM: 2GB
- Operating System: Windows, Linux, MacOS
- Platform: Browser

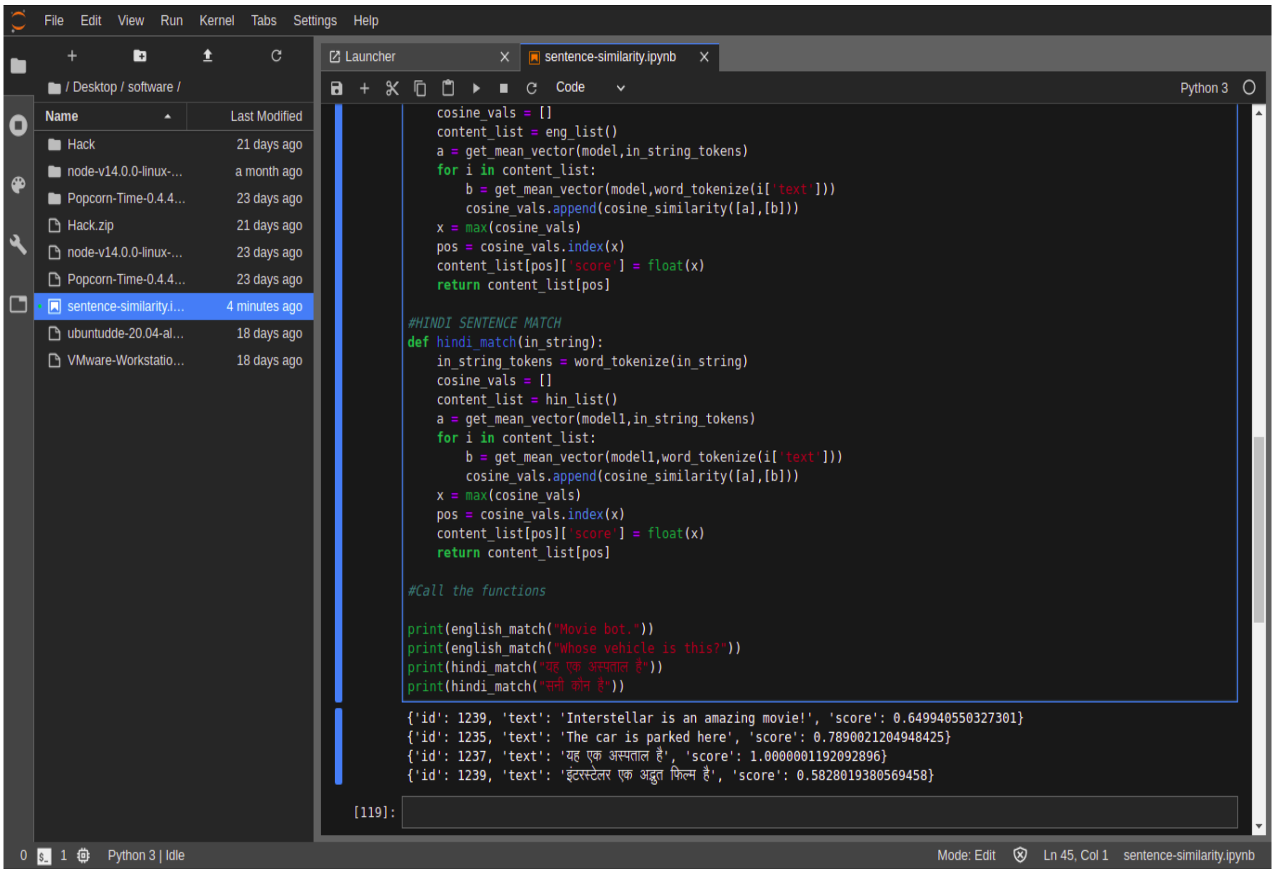
V. SYSTEM DESIGN

The system design is as follows:

1. Load all the necessary libraries: gensim, nltk, numpy, scikit-learn.
2. Load the fast word vectors for English 'wiki-news-300d-1M.vec' to gensim function load_word2vec_format('wiki-news-300d-1M.vec', binary=False)
3. Load the fast word vectors for Hindi '/home/sunny/FTM/Hindi/cc.hi.300.vec' to gensim function load_word2vec_format('/home/sunny/FTM/Hindi/cc.hi.300.vec', binary=False)
4. Save both the word-embedding models to temporary files.
5. Create two lists of dictionaries with ID and Text as keys.
6. Define a function get_mean_vector: (Input: word2vec_model, words, Output: [])
7. Define two functions: english_list(Input: None , Output: content_list_hindi) and hindi_list(Input: None , Output: content_list_hindi)
8. Define function english_match(in_string) -> content_list[pos]
9. Define function hindi_match(in_string) -> content_list[pos]

Vol. 9, Issue 6, June 2020

VI. SNAPSHOT


File Edit View Run Kernel Tabs Settings Help
/ Desktop / software /
Name
Last Modified
Hack 21 days ago
node-v14.0.0-linux-... a month ago
Popcorn-Time-0.4.4... 23 days ago
Hack.zip 21 days ago
node-v14.0.0-linux-... 23 days ago
Popcorn-Time-0.4.4... 23 days ago
sentence-similarity.i... 4 minutes ago
ubuntudde-20.04-al... 18 days ago
VMware-Workstatio... 18 days ago
Launcher
sentence-similarity.ipynb
Code
Python 3
Python 3 | Idle
Mode: Edit
Ln 45, Col 1
sentence-similarity.ipynb

```
    cosine_vals = []
    content_list = eng_list()
    a = get_mean_vector(model,in_string_tokens)
    for i in content_list:
        b = get_mean_vector(model,word_tokenize(i['text']))
        cosine_vals.append(cosine_similarity([a],[b]))
    x = max(cosine_vals)
    pos = cosine_vals.index(x)
    content_list[pos]['score'] = float(x)
    return content_list[pos]

#HINDI SENTENCE MATCH
def hindi_match(in_string):
    in_string_tokens = word_tokenize(in_string)
    cosine_vals = []
    content_list = hin_list()
    a = get_mean_vector(model1,in_string_tokens)
    for i in content_list:
        b = get_mean_vector(model1,word_tokenize(i['text']))
        cosine_vals.append(cosine_similarity([a],[b]))
    x = max(cosine_vals)
    pos = cosine_vals.index(x)
    content_list[pos]['score'] = float(x)
    return content_list[pos]

#Call the functions

print(english_match("Movie bot."))
print(english_match("Whose vehicle is this?"))
print(hindi_match("यह एक अस्पताल है"))
print(hindi_match("सनी कौन है"))
```

```
{'id': 1239, 'text': 'Interstellar is an amazing movie!', 'score': 0.649940550327301}
{'id': 1235, 'text': 'The car is parked here', 'score': 0.7890021204948425}
{'id': 1237, 'text': 'यह एक अस्पताल है', 'score': 1.0000001192092896}
{'id': 1239, 'text': 'इंटरस्टेलर एक अद्भुत फिल्म है', 'score': 0.5828019380569458}
```

```
[119]:
```

Fig 1: Snapshot of the output.

The figure 1 shows the output of average of word vectors to get one equivalent word of vector for the whole sentence.

VII. CONCLUSION

Sentiment Analysis is a subfield of NLP used for contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations. The similarity of sentences constructed in Hindi and English is discussed.

REFERENCES

[1]. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, XiangRui Wang, and Chih-Jen Lin.. "LIBLINEAR: A library for large linear classification", JLMR, 9:1871–1874, 2008.
[2]. Manaal Faruqui and Chris Dyer "Community evaluation and exchange of word vectors", ACL: System Demonstrations, 2014.
[3]. Lev Finkelstein, Ehud Rivlin Zach Solan Gadi Wolfman Evgeniy Gabrilovich, Yossi Matias, and Eytan Ruppin, "Placing search in context: The concept revisited", TOIS, 20(1):116–131, January, 2002
[4]. Bin Gao, Jiang Bian, and Tie-Yan Liu ,"WordRep: A benchmark for research on learning word representations", ICML Workshop on Knowledge Powered Deep Learning for Text Mining 2014.
[5]. Remi Lebret and Ronan Collobert,"Word embeddings through Hellinger PCA", EACL, 482–490, 2014.
[6]. Omer Levy, Yoav Goldberg, and Ido Dagan," Improving distributional similarity with lessons learned from word embeddings" TACL, 2015
[7]. https://cloud.google.com/what-is-machine-learning, "Cloud Machine Learning Services", 2019