International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-1, May 2020

Differential Hiring using a Combination of NER and Word Embedding

Suhas H E, Manjunath A E

Abstract*: In this data-driven world, AI is being used in almost all the tasks to automate processes and make human life more comfortable. One such industry where Artificial Intelligence (AI) importance is growing is the recruiting industry. This paper aims to propose a new and a better method to match the most suitable talent to jobs, which has been incorporated using two methods – suggesting top resumes to a job opening from a talent pool to a recruiter, recommending top jobs which match to a candidate based on the candidate's resume. Natural Language Processing techniques have been used in implementing this approach – Named Entity Recognition (NER), Word embedding model, and Cosine similarity using which a resume and job will be matched. The NER model is used to extract useful entities from documents, which is enhanced by the word2vec model by making the system more generic and the similarity is calculated using the cosine similarity algorithm.

Keywords: Cosine Similarity, Differential Hiring, Natural language Processing, Word Embedding.

I. INTRODUCTION

The importance of the internet is increasing every day and being used for many activities like online shopping, social media. Searching for jobs on the internet is quite popular and applying to them is a much easier process online. Companies can hire sufficiently qualified employees with the help of online recruitment websites and applications. Going through thousands of resumes and storing them in an unstructured format is highly time-consuming and expensive.

Natural Language Processing (NLP) is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. Natural Language processing is considered a challenging problem in computer science. It's the nature of the human language that makes NLP difficult. The goal is for computers to process or "understand" natural language to perform tasks.

AI for recruiting is an emerging technique of HR technology designed to reduce — or even remove — time-consuming activities like manually screening resumes.

The biggest obstacle in recruiting talent acquisition is still screening resumes: 52% of talent acquisition leaders say the hardest part of recruitment is identifying the right candidates from a large applicant pool.[1]

Manuscript received on April 02, 2020.
Revised Manuscript received on April 15, 2020.
Manuscript published on May 30, 2020.
* Correspondence Author
Suhas H E*, Computer Science, R V College of Engineering, Bengaluru, India. Email: suhas.2ab@gmail.com
Manjunath A E, Computer Science, R V College of Engineering, Bengaluru, India. Email: aemanju@gmail.com

According to a survey of talent acquisition leaders, 56% say their hiring volume increase this year, but 66% of recruiting teams will either stay the same size or contract [1]. The position of AI has been improving in the recruitment space for quite a few years now. The recent past already showed us that, in one form or another, AI is becoming a necessary tool in the recruiter space.

The paper is organized as follows. Section II discusses the literature Survey. Section III discusses the natural language processing concepts required to understand the remainder of the paper. Section IV discusses the flow of the data that happens for the analysis of data. Section V presents the results of this paper. Section VI discusses the conclusion and prospects of the paper.

II. LITERATURE SURVEY

In [2], this paper resume analysis was done using deep learning techniques such as the Convolutional Neural Network, Conditional Random Field (CRF), and LSTM (Long Short-Term Memory). Different parts of the resume are classified using a CNN model. This is one of the most recent papers which uses deep learning for resume parsing and identifying different identities. Word embedding is achieved by the usage of a Pretrained Glove model. The resume is segmented into 3 parts, and the extraction of several entities was done for further analysis.

In [14], the resumes undergo a series of analysis steps such as syntactic analysis, semantic analysis, and Lexical analysis. In each step of the analysis, attributes are extracted from which inferences are drawn, or the data extracted is cleaned to make it less noisy. The crux of the algorithm lies in the number of matched items from the existing candidate skill set database, which is continuously updated.

Ontology-based models are proposed by the following papers by Celik D et al. [4], and [5], a cascaded hybrid model has been shown as another approach by Yu K et al. [6]. The ontology-based model involves the aggregation of skills in a resume pool through its semantic approach. Data preprocessing steps are applied to remove HTML tags, and websites are parsed for information corpus. The paper was confined to segmentation, extraction of personal data, and obtaining educational information. Chuang et al. [7] another segmentation approach for resume parsing, based on pattern matching, split the resume into different blocks and then further steps of analysis applied on each block. The resume is initially classified into three types - structured document, freestyle document, and semi-structured document. The noteworthy point in this paper is the introduction of a feedback control algorithm that aims to improve the performance of the model based on the data received by the users. Named Entity Recognition (NER) is one of the toughest problems in NLP.

State of the art NERs requires a vast amount of manually annotated corpora and domain-specific knowledge to achieve high precision. In [8], using a neural architecture based on bidirectional LSTM and conditional random fields, performance has been increased without using any language-specific or domain-specific knowledge. Anjuna and Ushadevi present a pattern matching model that applies to the documents which are in an unstructured format.[11]. Using regular expressions, they convert structured data to unstructured data. Mainly, the system extracts useful entities such as email id, name, phone number, and date of birth. Initially, techniques such as stop word removal and Stemming are applied to clean the data. Concluding this section, there has been research conducted on different parts of resume parsing on the segmentation of different blocks in a resume, extracting entities from these blocks to gain insights into the resume. Since the methods used in current research do not have a generic way of matching talent and it provides only 1-way matching (resumes to job openings), we are proposing a new system which continually evolves by learning and provides 2-way matching (resumes to jobs and jobs to candidates).

III. NATURAL LANGUAGE PROCESSING TECHNIQUES

A. Named Entity Recognition

Named Entity Recognition (NER) is the process in which required nouns are identified in the text. As shown in the above diagram, NER will be applied to both job openings and candidate's resumes to extract important attributes from them, such as technical skills, soft skills, experience required/experience of the candidate extracted from the resume, previous companies worked at, educational institution, location. These attributes play an important role in ranking the document against the input text.

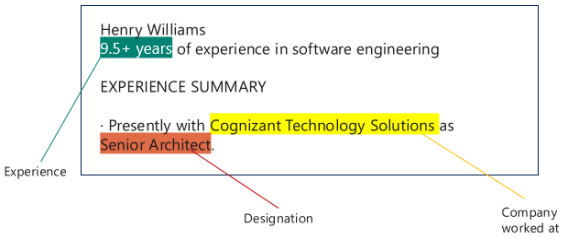

Fig. 1.Example of NER applied on a Resume.

Training a NER

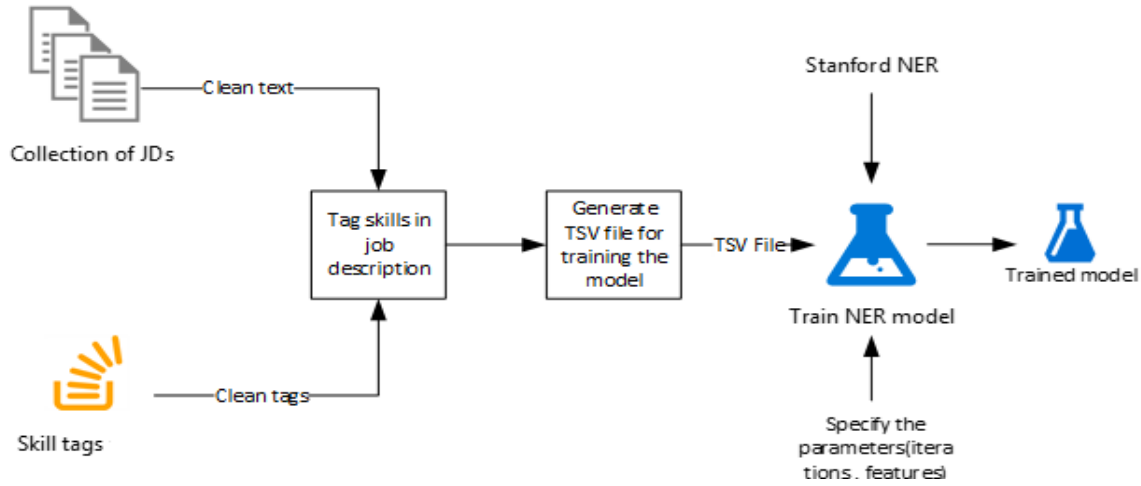

Fig. 2.Training a custom NER model.

The accuracy of the NER model depends on the data that is used to train the model and the pre-processing steps that have been applied to reduce the noise in the data and the accuracy of annotations in the labeled data. There are several steps which were followed to train a NER which should recognize technical skills from the given input text:

1. A collection of documents which is related to the problem at the handset of Job openings, set of resumes with the approval of candidates were taken.
2. Series of pre-processing steps were applied to these documents such as stop word removal, stemming, punctuation removal, which yielded only words that are crucial to NER training.
3. A data dump of technical skills [13] used to tag the technical skills in each of these input documents.
4. A Tab-separated value (TSV) file is generated for each of these annotated documents and provided as input to the Stanford NER package
5. After 20 iterations, the accuracy of the NER is optimized, and the model is obtained.

A conditional random field (CRF) based network provided by the Stanford NER package has been used to build the custom entity recognizer. As explained in [12], a Gibbs sampling method is used, which forms long-distance relationships with distant words, which enhances the performance of the model.

B. Word Embedding model (Word2vec)

Word2vec is a word embedding model that processes text. It is a simpler and computationally efficient way of learning the word embeddings.

As shown in the above diagram, Word2vec is a 2-layer shallow neural network that takes adjacent input words and tries to predict the current word. 2 types of models can be used – either a Skip-gram model or CBOW (Continuous bag of words) model.

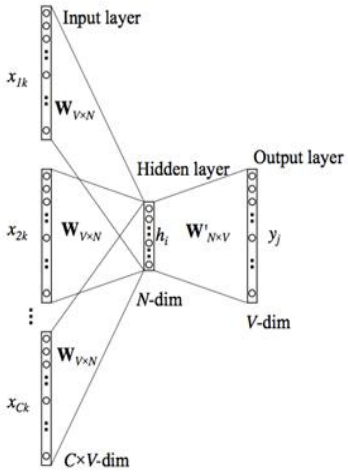

Fig. 3.Word2Vec architecture
Source: Adapted from [21]

With sufficient data and context, precise guesses about a word's meaning based on past appearances can be made by the Word2Vec model. The guesses made by the model are used to form an association between words (e.g., "woman" is to "girl" what "man" is to "boy"), or cluster documents and classify them by topic. CBOW model has been chosen as it has been proven to be a faster model than skip-gram.

Training a Word2Vec model

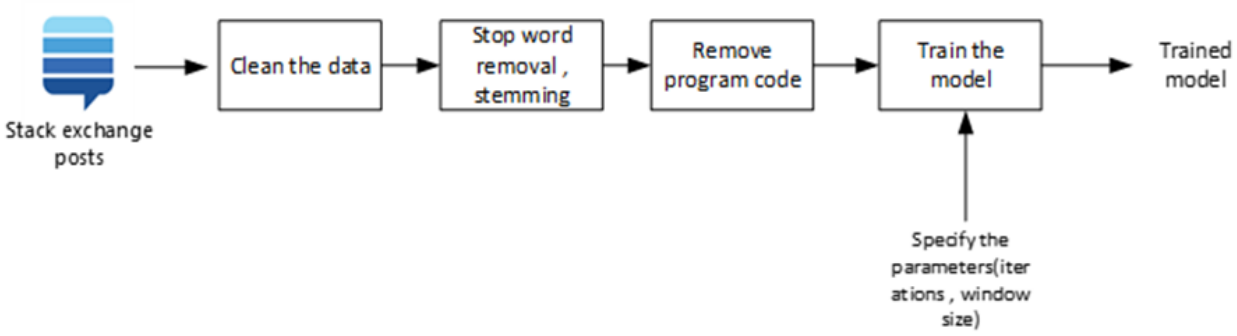

Fig. 4.Training a Word2vec model

Word2vec training begins with a large corpus of data related to the domain of the problem statement; in this scenario, data has been taken from the stock exchange, which is a website where users ask questions based on programming, computer science concepts. The steps involved in training a wod2vec model are

1. Clean the dataset by removing punctuation, appending different posts to form a vast corpus.
2. Apply techniques such as stop word removal, stemming from cleaning the data further
3. Since a lot of these questions have program code in them, which is not useful in making predictions, this is removed using regular expressions.
4. After the pre-processing steps have been applied, the model is trained for several iterations by specifying the parameters such as window size, model type – Skip-gram, or CBOW.
5. The completion of training yields a model which, when takes input, a word and outputs similar words as output, the results of which are shown below.

IV. DESIGN

The overall flow of the process is depicted as in Figure 4 below. It depicts how the end to end process occurs given a Job opening description and a resume. It has 3 essential steps Named Entity Recognition, Word2vec, and Cosine similarity.

Named Entity Recognition

As explained above, a NER model is trained, and the model is prepared. The NER model takes to input the text of the document, which can be a JD/Resume text and recognizes entities in them.

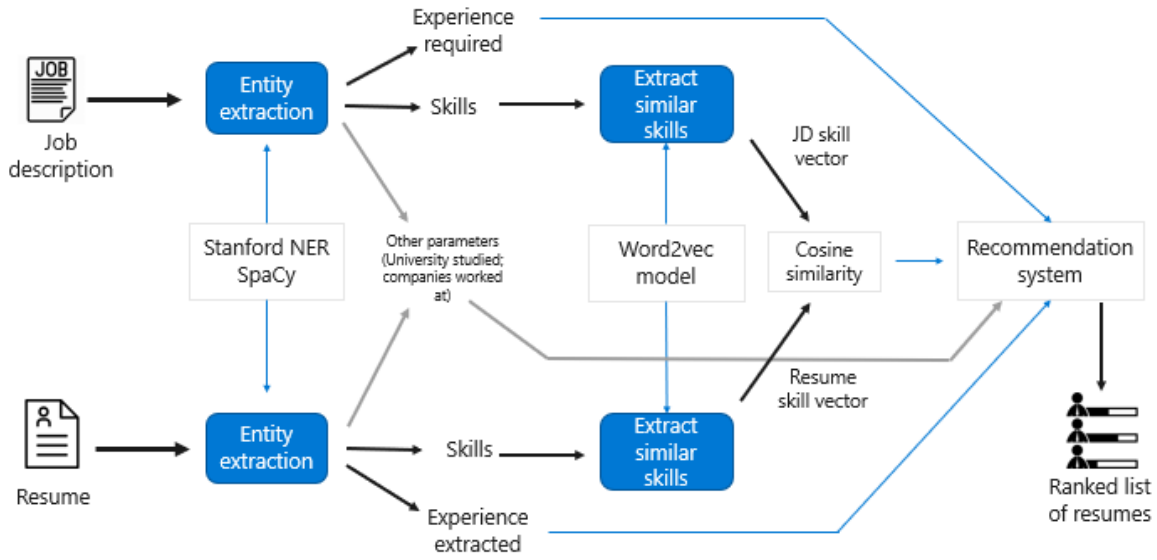

Fig. 5.The flow of the process

Word2vec model

The skills extracted from the NER are provided as input to the word2vec model yields similar skills to the skills extracted by the NER. The most important part played by the word2vec is that it gives weightage to a resume even if that candidate has similar skills to the ones mentioned in the job opening description.

Input text	Output from word2vec		
Azure	ec2	Heroku	Cloud
HTML	CSS	XHTML	JavaScript
C	C++	Pascal	Swift

Fig. 6.Outputs of the trained word2vec model

As depicted in the above table, if a job opening has a skill required such as Azure, when this word is given as input to the word2vec model words such as ec2, Heroku, and cloud have been obtained as output. In reality, each of these output words is accompanied by a floating number in the range of 0 to 1, indicating the closeness to the input word, which plays an essential part in the next step of the process.

Cosine Similarity

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Fig. 7.Cosine similarity expression

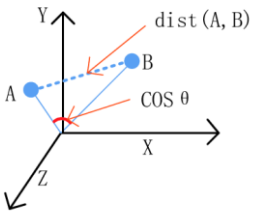

Fig. 8.Cosine similarity between 2 vectors

Cosine similarity is the final step of the process, which determines how much a given resume and job opening matches. Cosine similarity literally translates to the angle between the 2 vectors representing a JD (Job Description) and a candidate's resume. Based on the research done in [19][18], the best way to apply cosine similarity between the resume and JD was ideated and summarized below.

The steps followed to obtain a list of ranked candidates is as follows:

1. The skills extracted from each of the JD and resume were obtained along with similar skills obtained from the word embedding model.
2. The set of union of all the skills was taken, and the vector length was set to the length of this set.
3. 2 vectors are created for a JD and resume, respectively. For each skill in the union of skills, the score of each skill is taken as 1 if it is present directly in the document. Otherwise, the score is taken as per the output of the word2vec model.
4. Using the above formula, as in figure 6, the cosine similarity is calculated against each JD and sorted according to the score. The top 5 to 10 candidates are recommended to the recruiter.

V. METRICS AND RESULTS

Named Entity Recognizer

NER models are usually measured for their accuracy by comparing the outputs of the NER model and human-labeled counterparts.

Its particularly difficult to measure the accuracy in this case because we are identifying words, so what happens in the case the output words overlap with the correct output either partially or subsumes it. Exact-match Evaluation [16] mainly involves two subtasks: boundary detection and type identification. More specifically, the number of False positives (FP), False negatives (FN), and True positives (TP) are used to compute Recall, Precision, and F-score.

• False Positive (FP): an entity that is the output of the NER model but does not appear in the ground truth.

• False Negative (FN): an entity that is the output of the NER model but appears in the ground truth.

• True Positive (TP): an entity that is returned by a NER system and appears in the ground truth.

Precision indicates the percentage of our results, which are correctly recognized. Recall indicates the percentage of total entities correctly recognized by the NER model, as shown below.

$$\text{Precision} = \frac{\#TP}{\#(TP + FP)} \qquad \text{Recall} = \frac{\#TP}{\#(TP + FN)}$$

F-score takes both Precision and Recall and computes the harmonic mean of the two. It provides a more realistic measure of the performance of the model.

$$\text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Fig. 9.Expressions for different metrics

The optimization of models went through 4 main iterations which are explained in detail below:

• Iteration 1

Trained the NER with minimal preprocessing steps such as removal and punctuation initially with around 100 manually annotated resumes.

• Iteration 2

Text preprocessing supplemented with techniques like stop word removal and Stemming.

• Iteration 3

The dataset was increased from 100 to 200 manually annotated resumes increasing the training dataset.

• Iteration 4

Several optimizations are done. A dictionary of skills was maintained to consider skills with symbols in them, such as C#; the window size was increased to take into consideration more words on either side of the current word during the training of the NER.

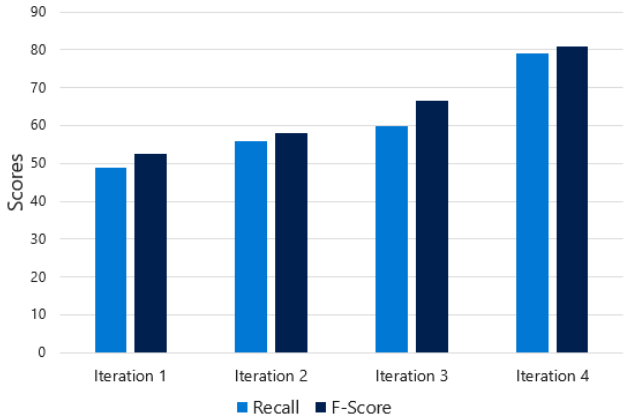

Fig. 10. Variation of Recall and F-score with each iteration

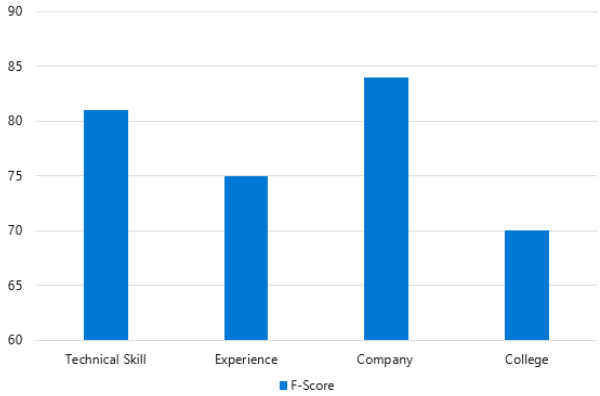

Fig. 11. Variation of Recall and F-score with each iteration

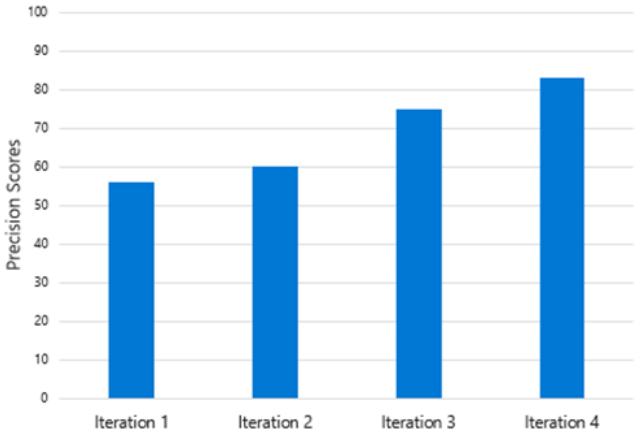

Fig. 12. Variation of Precision scores with each iteration

Shown in Figure 11 is the F-score with respect to each of the entities which the model has been trained, an average of F-score has been taken for the four iterations and plotted, it can be observed that experience has the lease F-score since the format of experience mentioned in resumes varies to a large extent, and the model is not able to identify it accurately.

VI. CONCLUSION AND FUTUREWORK

This paper presents a method to match talent using Natural Language Processing techniques such as NER and word embedding. The system built on the proposed method is successfully abele to match jobs and candidates in the 2 scenarios discussed.

The novel part of the method being extraction of skills that are similar to the ones present in the document with the help of a word embedding model which makes the model more inclusive. Based on the metrics of the model performance, it can be said that technical skills are extracted with high enough accuracy. Also, the system is built in such a way that there is an absence of absolutely any kind of bias in the model and treat all the applicants relatively [17]. Thus, this method of talent matching greatly enhances the work of a recruiter as well as the job application process, which alleviates the problem of going through thousands of job openings, making it a seamless process.

Future work can be directed towards developing techniques of a feedback loop to increase the performance of the model continually. Feedback buttons are provided at the user interface level of the application where the recruiter can provide the feedback, and accordingly, the parameters of NER and word2vec training are tuned to obtain the maximum match between resume and job opening.

REFERENCES

1. Article by Ideal: Effective recruiting using AI, Article link.
2. Ayishathahira C H, Sreejith C, Raseek C. Combination of Neural Networks and Conditional Random Fields for Efficient Resume Parsing. 18233630
3. Sayed Zainul Abideen Mohd Sadiq, Juneja Afzal Ayub, Intelligent hiring with resume parser and ranking using natural language processing and machine learning.
4. D. Çelik, A. Elçi, "An ontology-based information extraction approach for résumés," Joint International Conference on Pervasive Computing and the Networked World, pp. 165-179, 2012.
5. D. Çelik, A. Karakas, G. Bal, C. Gültunca, A. Elçi, B. Buluz, M. C. Alevli, "Towards an information extraction system based on ontology to match resumes and jobs," Computer Software and Applications Conference Workshops (COMPSACW) 2013 IEEE 37th Annual, pp. 333-338, 2013.
6. K. Yu, G. Guan, M. Zhou, "Resume information extraction with the cascaded hybrid model," Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL 05, 2005.
7. Z. Chuang, W. Ming, L. C. Guang, X. Bo, L. Zhi-qing, "Resume parser: Semi-structured Chinese document analysis," Computer Science and Information Engineering 2009 WRI World Congress on, vol. 5, pp. 12-16, 2009.
8. G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, "Neural architectures for named entity recognition," arXiv preprint, 2016.
9. C. Dyer, M. Ballesteros, W. Ling, A. Matthews, N. A. Smith, "Transition-based dependency parsing with stack long short-term memory," arXiv preprint, 2015.
10. Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, "Hierarchical attention networks for document classification," Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480-1489, 2016.
11. M. Anujna, A. Ushadevi, "Converting and deploying an unstructured data using pattern matching," American Journal of Intelligent Systems, vol. 7, no. 3, pp. 54-59, 2017.
12. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling
13. The data source for the skills used in NER training, Data source
14. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: Efficient Estimation of Word Representations in Vector Space 1301.3781
15. The data source for training the word embeddings model (word2vec) taken from the stack exchange website: Data source
16. Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li, A Survey on Deep Learning for Named Entity Recognition 1812.09449
17. James Vincent, Article by the Verge: Amazon reportedly scraps internal AI recruiting tool that was biased against women by James Vincent Article
18. Alfirna Rizqi Lahitani; Adhistya Erna Permanasari; Noor Akhmad Setiawan: Cosine similarity to determine similarity measure
19. Pinky Sitikhu1, Kritish Pahi, Pujan Thapa: A Comparison of Semantic Similarity Methods for Maximum Human Interpretability
20. Y. Bengio, R. Ducharme, P. Vincent. A neural probabilistic language model. Journal of Machine Learning Research, 3:1137-1155, 2003
21. Source for image of word2vec architecture Source

AUTHORS PROFILE

Suhas H E, computer science undergrad at RV College of Engineering, Bengaluru. Having worked on several projects in the domain of Machine learning and Deep learning, this is my first publication. As a developer, I have participated and won in several national level hackathons including Microsoft Imagine cup.

Manjunath A E, Currently, working as an assistant professor in the Department of Computer Science, R V College of Engineering. I have completed a Postdoctoral Fellowship (PDF) at the National Institute of Technology (NIT), Tiruchirappalli. Four patents have been filled and waiting for the examination for granting the patents. Received the Young Scientist award from the Government of Karnataka in the year 2015. I have published 15 research papers and 2 book chapters in Springer publication.