Scoring of Resume and Job Description Using Word2vec and Matching Them Using Gale–Shapley Algorithm

Chapter · January 2022

DOI: 10.1007/978-981-16-2126-0_55

6 authors, including:

Shushanta Pudasaini
Tribhuvan University
2 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Sujan Adhikari
Pokhara University
9 PUBLICATIONS **126** CITATIONS
SEE PROFILE

Scoring of Resume and Job Description using word2vec & matching them using Gale Shapley Algorithm

Shushanta Pudasaini[1], Subarna Shakya[2], Sagar lamichhane[3], Sajjan Adhikari[4], Aakash Tamang[5], and Sujan Adhikari[6]

[1] Advanced College of Engineering & Management, Kupondole Road, Lalitpur
shushanta574@gmail.com
[2] Institute of engineering, Tribhuvan University, Pulchowk, Lalitpur
drss@ioe.edu.np
[3] Herald College Kathmandu, Hadigaun Marg, Kathmandu
sagar_lamichhane@yahoo.com
[4] Nagarjuna College of Information Technology, Bangalamukhi, Lalitpur
sajjanadhikari464@gmail.com
[5] Patan College For Professional Studies, Kupondole, Patan
dumjanaakash@gmail.com
[6] NAMI college, Gokarneshwor, Kathmandu
zeradt@gmail.com

Abstract. The paper introduces a quick-witted system that assists employers to find the right candidate for a job and vice-versa. Multiple approaches have to be taken into account for parsing, analyzing, and scoring documents (CV, Vanancy details). However, In this paper we have devised an approach for ranking such documents using word2vec algorithm and matching them to their appropriate pair using Gale-Shapley algorithm. When ranking a CV, different cases are taken into consideration: skills, experience, education, location. The ranks are then used to find an appropriate match of employers and employees with the use of Gale-Shapley algorithm which eases companies for higher best possible candidates. The methods experimented for the scoring and matching is explained below on the paper.

Keywords: Natural Language Processing, NER, word2vec, Gale Shapley, CBOW, word embedding, Cosine similarity

1 Introduction

To hire an employee, CVs are screened manually by the HR and based on the qualifications the certain persons are hired. This system has been running for a long time. After the advancement in Natural Language Processing, CV parsing has become an integral part of it. Furthermore, if the CV can be judged similar to the human judgement then it would be a more sophisticated system. As the system can process and manipulate large CVs in a short period of time. This

paper describes a method to identify the standard of the CV and match them based on the requirements of a certain company.

So, the scoring model has been proposed in this paper to provide a score to the CV. Several criteria have been displayed to provide the score to the CV. The designation of the job seeker, progress score, location, skills and score plays vital roles in scoring of the CV. These factors were selected after several experiments and suitable conditions have been applied. Similarly, to make it easier for job providers and seekers to find the best match among themselves the Gale-Shapley is in action. It helps to find the best pair among the both parties. As this algorithm has been used for finding pairs for marriage this paper aims to use it for pairing of employees with employers and vice versa.

2 Literature Review

Document ranking or comparing the similarity between the documents has been identified as one of the major tasks in the Natural Language Processing related activities. Different measures with different ideas have been implemented for getting a better result. In the paper [1] the author has focused on measuring the similarity between sentences. The paper shows that even though word2vec and cosine similarity gives appropriate results, it is still possible to improve the similarity result of the sentences by combining the word embeddings with the hand-engineered features. Similarly, on the paper titled as [2] shows the way of making online recruitment more efficient and effective. The paper [3] shows the way for recruiters to find the appropriate candidate by their skills using word2vec algorithm. The skills were extracted from vancacy details and classified according to the vancacy details with the same content. Then it was trained on the word2vec algorithm. The paper [4] defines a way of hiring using the combination of word embeddings and NER model. [5] is a system for personality evaluation and CV analysis where users have to manually fill up the form, give aptitude tests and upload their CV on the website. The evaluation is done on the reference to Machine Learning algorithms. [6] proposes many methods to solve the problem of profiling applications according to a specific work offer, based on vectorial and probabilistic models. Their target is a framework capable of reproducing the recruitment consultant's judgment. By using ROC curves, we have evaluated a number of similarity measures to rank candidates.

The paper [7] matches the employee with the offered position based on their qualification with the use of Gale Shapley algorithm. [8] also puts a new way for creating a social network and matching of the jobs with the use of Gale Shapley algorithm. [9] describes a procedure to screen and match the CV with profiles, skills and other features available in the CV with clustering algorithms. The matching model were made from 2283 CVs and job requirements.

3 Methods

3.1 Scoring

To perform scoring between two documents, we have proposed the following approaches: -

- Scoring of description with multiple CV
- Scoring of one CV with multiple vancacy details
- Scoring of CV content with vancacy details content

Before getting into these approaches, we will discuss two techniques that we have implemented in getting our result. They are: -

Word Embedding using Word2Vec Algorithm

To match the CV with vancacy details, we have used the help of word embeddings, which is just the numerical representation of words in the form of vectors. Word embeddings have been proven to be very effective when working with textual data in the field of Natural Language Processing.

Here, we have implemented the word2vec model which implements the Neural Network architecture for generating the word embeddings of the documents [10] and has been providing the state-of-the-art word embeddings results. The embedding vectors that we receive from the word2vec model is a dense one. Whereas, other embedding approaches like one hot encoding gives a sparse vector as its result. In the context of capturing the relation between different words, dense vectors give a better result than the sparse vector [11]. And as we want to capture the relationship between the words, word embedding using word2vec model was the appropriate choice for us.

A word2vec model can be generated with two different approaches: -

- Skip-gram
- CBOW (Common Bag of Words)

The functionality of the skip-gram model is that it takes a certain word and tries to predict its surrounding word or context words. Whereas in CBOW, it takes the context and predicts the target word.

The word2vec model is trained with a data corpus comprising 12,000 CVs using the CBOW model.

Measuring Similarity

To perform the text similarity, there are 3 different approaches i.e. string based, corpus based and knowledge based. Here, we will be following the string based similarity approach, where we measure how similar or dissimilar a pair of strings are. [12]

In the string based similarity method, similarity can be measured using various approaches like cosine similarity, dice's similarity, euclidean distance and

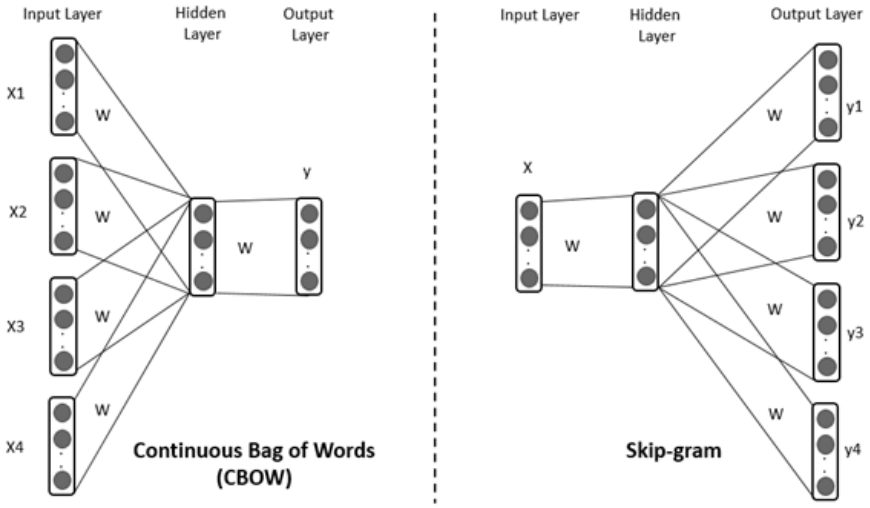

Fig. 1. Neural Network architecture of CBOW and Skip-gram Model

jaccard similarity. Among them, cosine similarity is the one which is frequently used for measuring the distance between the two given vectors in a vector space.

$$\cos(\theta) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}}$$

If the pair of strings are close to each other, then we will get a value of θ smaller (closer to 0) giving us the output value closer to 1. And if the strings are not closely related to each other, the value of θ will be higher giving the output value closer to 0.

Scoring of one vancacy details with multiple CV
This scoring module is used when a score for vancacy details is required in comparison to multiple CVs.

The result we receive is of a json type, having two keywords i.e. "user_profiles" and 'job'. The keyword "user_profiles" holds the record of multiple CVs that includes their id, personal information, experience details and technical and soft skills details. The other keyword 'job' holds the information of the vancacy details It includes its id, job title, vancacy details and location.

The score which is generated from the model, is based on multiple entities (condition). Each entity holds a certain weightage that has been assigned to it after appropriate testing. Higher the score, higher similarity between the vancacy details and CV. The entity along with their respective weightages are: -

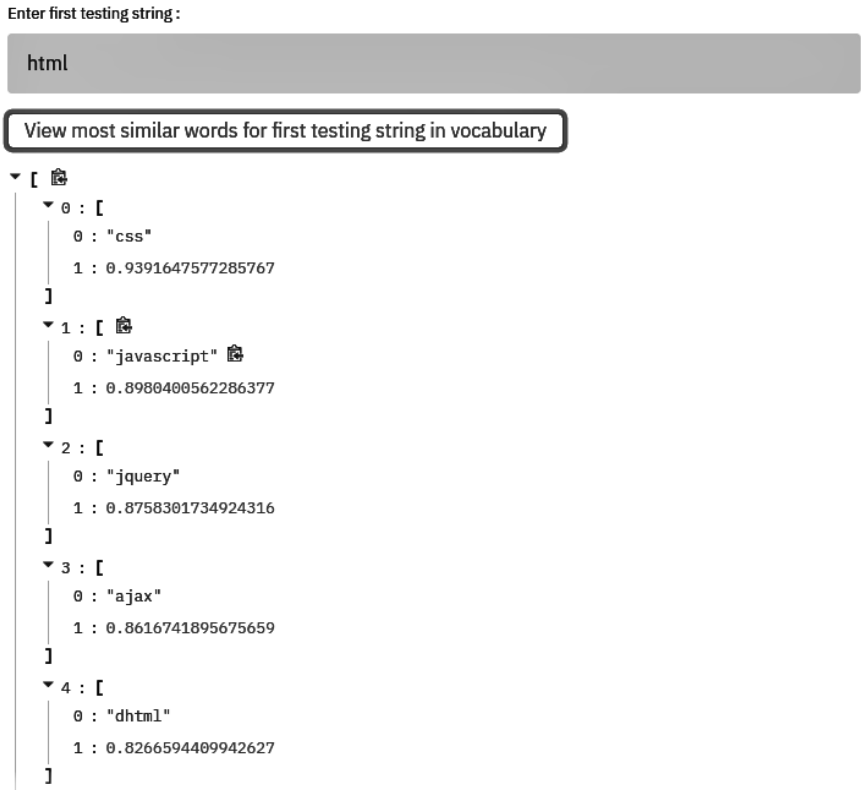

Fig. 2. most similar tokens of a given word

Scoring entity	Description
Skills Score	25
Experience Score	15
Designation Score	40
Distance Score	5
Progress Score	15
Total Score	**100**

In the scoring function, the skill sets are used to calculate the skills score. And the locations are used to calculate the distance score. The nearer the CVs location, higher the score. Likewise for scoring the designation, the system checks if the job-title is provided on the json and the designation from the job-content is parsed using the custom trained NER tagger then added together to form a designations list. The system checks if any values in user-designation and the job designation list match and if they do the score is generated.

The procedure of scoring on the basis of "Progress score" is quite different from the others.It first visualizes the 'designations' and 'designation_dates' graphically and calculates the slope of the graph. Working on this method we got to know about the priority of categorizing a designation as per its hierarchy. However there wasn't any predefined way to do this. So, we created a rule based file parsing method which takes a file containing every possible designation and categorizes them into Junior Employee, Senior Employee, Intermediate Employee or Manager. This normalizes the designation and the relevant slope can be calculated. The slope provides the stability of candidates for the designations.

Scoring of one CV with vancacy details content

One of the major differences between this scoring approach and the other previous approaches is that here we do not parse the documents. The CV contents and vancacy details contents are preprocessed with steps such as text cleaning, stopwords removal, lemmatization and so on. Then the filtered tokens from the CV and vancacy details are passed to the custom trained Word2Vec model. Evaluation of word embeddings of every token is performed and the mean embedding representing whole textual content is calculated. After getting the mean embedding from both CV and vancacy details, cosine distance similarity is applied to calculate the similarity score.

3.2 Matching using Gale Shapley Algorithm

After the completion of generating scores for the required document, we use those scores for recommending appropriate CVs to any given job-descriptions and vice-versa using an accredited pairing algorithm called "Gale Shapley" which was devised by David Gale and Llyod Shapley in 1962. This algorithm is capable of solving any pairing problems across the world like students choosing best universities in online educational platforms or women looking for men and vice-versa in dating sites and connecting users to an internet service in the smallest amount of time. In all of these cases, we are required to create a stable set of

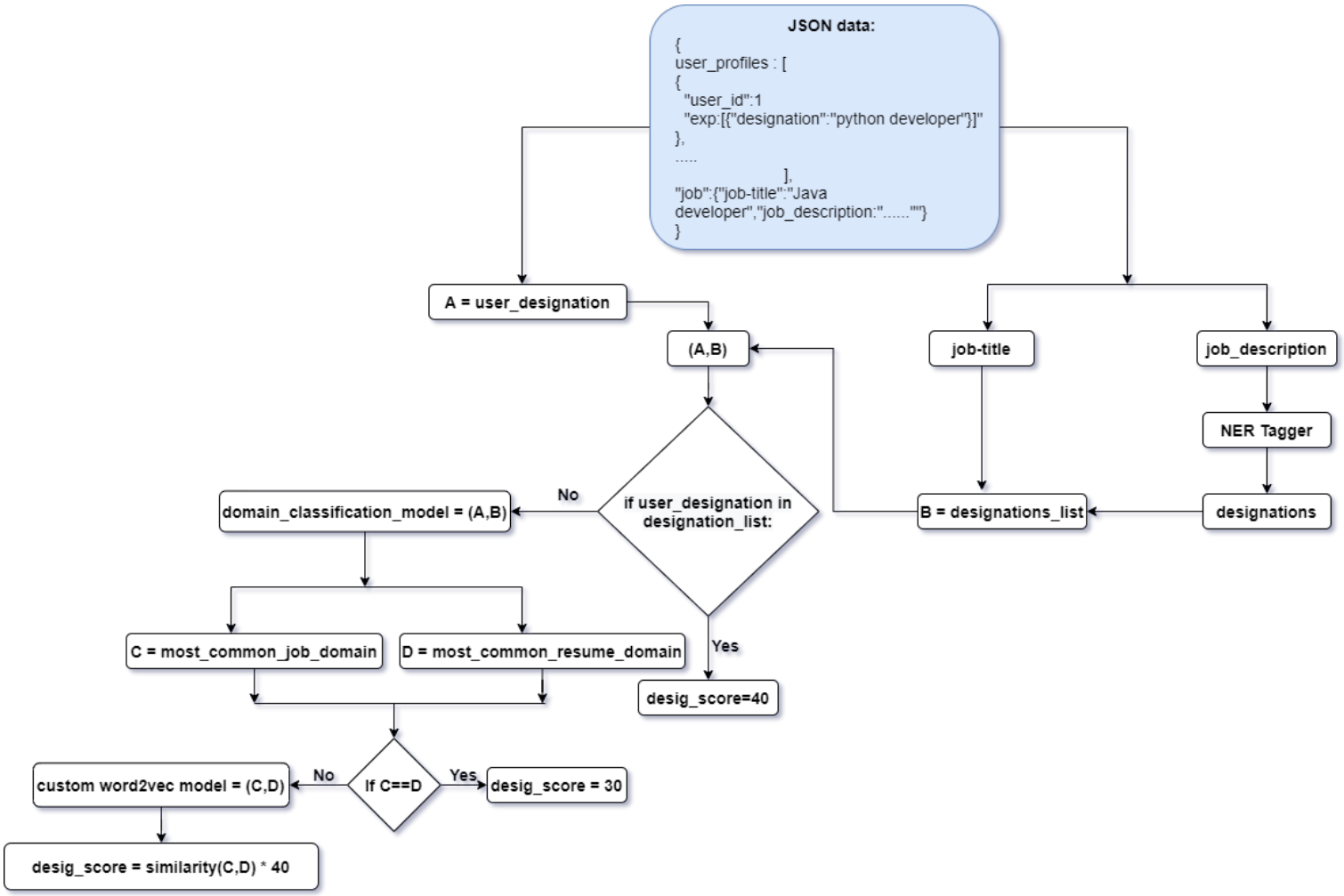

Fig. 3. Flow diagram of the designation scoring model

pairs that needs to satisfy a given criteria. Theoretically, a stable pair is formed when a pair (A, B) has no better options than each other.

We made use of the proposed pseudocode for the stable marriage problem, to solve the pairing problem in our system. Like in the proven example of stable marriage, we have two sets of pairs in our system, CV and job-descriptions. And each element has to find the best candidate on the other set, i.e. finding the best candidate for a given company or finding the best company for any given candidate.

These two sets of data are acquired as a json file in our system. With a keyword representing its respective section. The first keyword holds the data for each job-description with its respective score for each CV. This score is provided by the scoring endpoint that scores one job-description with multiple CV.

```
employerPrefs = {
 'Job_1':  ['Job_seeker_1', 'Job_seeker_2', 'Job_seeker_3', 'Job_seeker_4', 'Job_seeker_5', 'Job_seeker_6'],
 'Job_2':  ['Job_seeker_5', 'Job_seeker_6', 'Job_seeker_3', 'Job_seeker_2', 'Job_seeker_1', 'Job_seeker_4'],
 'Job_3':  ['Job_seeker_2', 'Job_seeker_4', 'Job_seeker_6', 'Job_seeker_3', 'Job_seeker_1', 'Job_seeker_5'],
 'Job_4':  ['Job_seeker_6', 'Job_seeker_5', 'Job_seeker_2', 'Job_seeker_1', 'Job_seeker_4', 'Job_seeker_3'],
 'Job_5':  ['Job_seeker_4', 'Job_seeker_6', 'Job_seeker_1', 'Job_seeker_3', 'Job_seeker_5', 'Job_seeker_2'],
 'Job_6':  ['Job_seeker_5', 'Job_seeker_3', 'Job_seeker_6', 'Job_seeker_2', 'Job_seeker_1', 'Job_seeker_4']
}
```

Fig. 4. json representation of one jd and multiple CV

Similarly, the other keyword holds the data for each CV with its respective score for each job-description. This score is manually provided by the user to the respective job-description.

```
candidatePrefs = {
 'Job_seeker_3':  ['Job_3', 'Job_2', 'Job_4', 'Job_5', 'Job_1', 'Job_6'],
 'Job_seeker_1':  ['Job_2', 'Job_4', 'Job_5', 'Job_6', 'Job_3', 'Job_1'],
 'Job_seeker_2':  ['Job_6', 'Job_3', 'Job_1', 'Job_5', 'Job_2', 'Job_4'],
 'Job_seeker_6':  ['Job_3', 'Job_4', 'Job_1', 'Job_5', 'Job_2', 'Job_6'],
 'Job_seeker_5':  ['Job_5', 'Job_3', 'Job_6', 'Job_1', 'Job_4', 'Job_2'],
 'Job_seeker_4':  ['Job_3', 'Job_5', 'Job_4', 'Job_6', 'Job_2', 'Job_1']
}
```

Fig. 5. json representation of one CV and multiple jd

Values in both of these sets are ordered in the descending order, i.e. the most preferred CV or jd is at first. However, this data represents more than one stable set of arrangements. And this is where we make use of the algorithm that later provides a single set of arrangement for each CV or job-description.

```
Best Match between candidate and companies:
  Job_seeker_1 prefers Job_1,
  Job_seeker_2 prefers Job_3,
  Job_seeker_3 prefers Job_2,
  Job_seeker_4 prefers Job_5,
  Job_seeker_5 prefers Job_6,
  Job_seeker_6 prefers Job_4
```

Fig. 6. Stable set of arrangement of CV and job-description

4 Conclusion and Future Works

In this paper, the word2vec algorithm using CBOW model is implemented for creating the required embedding vectors and the cosine similarity for measuring the similarity score between CV and vancacy details. There are various approaches to generate such word embeddings like skip-gram model, TF IDF model and so on. However, the custom Word2Vec model trained on our custom data produced highly relevant results. Google pretrained Word2Vec model was also used for generating such word embeddings. However, a more accurate result was obtained by training such a custom Word2Vec model. The word2vec model for the algorithm was created using the Gensim library which was trained on 1220 documents consisting of CV and vancacy details. The result obtained on the test data for the scoring reflected a satisfactory result. While measuring the similarity of tokens, sentences or documents using the cosine similarity gave better results than other methods. Such better results were obtained due to the contextual awareness of such trained custom Word2Vec models.

The word2vec model could be replaced by other powerful embedding moldes such as ELMO, BERT etc. Such method can give better result than the current one. The data for creating the word embeddings can also be increased to get better result.

References

1. Misra, A., Ecker, B. and Walker, M., 2017. Measuring the Similarity of Sentential Arguments in Dialog
2. Faliagka, E., Ramantas, K. and Tsakalidis, A., 2012. Application of Machine Learning Algorithms to an online Recruitment System. The Seventh International Conference on Internet and Web Applications and Services
3. Le, Van-Duyet & Vo, Minh-Quan & Quang-An, Dang. (2017). Skill2vec: Machine Learning Approaches for Determining the Relevant Skill from Job Description.
4. Suhas H E, Manjunath A E, 2020. Differential Hiring using a Combination of NER and Word Embedding.
5. Rout, Jayashree & Bagade, Sudhir & Yede, Pooja & Patil, Nirmiti. (2019). Personality Evaluation and CV Analysis using Machine Learning Algorithm.

International Journal of Computer Sciences and Engineering. 7. 1852-1857. 10.26438/ijcse/v7i5.18521857.
6. Rauch, Jan; Raś, Zbigniew W.; Berka, Petr; Elomaa, Tapio (2009). [Lecture Notes in Computer Science] Foundations of Intelligent Systems Volume 5722 —— Job Offer Management: How Improve the Ranking of Candidates.
7. Elviwani, Elviwani & Siahaan, Andysah Putera Utama & Fitriana, Liza. (2018). Performance-based Stable Matching using Gale-Shapley Algorithm. 10.4108/eai.23-4-2018.2277597.
8. Arcaute, Esteban & Vassilvitskii, Sergei. (2009). Social Networks and Stable Matchings in the Job Market. 10.1007/978-3-642-10841-9_21.
9. Rodriguez, L. G., & Chavez, E. P. (2019). Feature Selection for Job Matching Application using Profile Matching Model. 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS).
10. Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space
11. Chiu, B., Crichton, G., Korhonen, A. and Pyysalo, S., 2016. How To Train Good Word Embeddings For Biomedical NLP. pp.166-174
12. H.Gomaa, W. and A. Fahmy, A., 2013. A Survey of Text Similarity Approaches. International Journal of Computer Applications