UDC 004.043

K. Martseniuk, A. Deveciogullari

EVALUATION OF UNSTRUCTURED RESUMES USING THE WORD2VEC MODEL

Abstract: The article addresses the problem of evaluating candidates' resumes for job vacancies using various natural language processing (NLP) methods.

Traditional text processing algorithms were analyzed, and a critical drawback of these methods was identified—their inability to account for semantic relationships between words, which is particularly important in the context of resume evaluation.

The BERT model was also considered, but it was dismissed due to its high computational complexity and excessive functionality for this task.

The primary choice for evaluating resume relevance was the Word2Vec method, which accounts for semantic relationships between words, contributing to greater objectivity and accuracy in the evaluation process.

The study results confirm the effectiveness of using Word2Vec compared to other methods in the context of resume analysis.

Keywords: machine learning, resume evaluation, natural language processing, Word2Vec, synonyms and related words, cosine similarity.

Introduction

Resumes are a crucial element in the employee selection process, providing employers with essential information about candidates' professional experience, skills, and qualifications. In recent years, the proliferation of online job search platforms has led to a significant increase in the number of applications submitted for job vacancies, complicating the work of HR managers and employers.

Manual processing of a large number of applications can result in errors and insufficiently objective evaluations of candidates, as humans are not always able to quickly and accurately process large volumes of information.

The rapid development of natural language processing (NLP) and machine learning has opened up new possibilities for the automated analysis and classification of textual data. These technologies enable the efficient processing and analysis of large amounts of information, identifying key points that can be utilized in the context of unstructured resumes.

Automated systems can greatly enhance the personnel selection process by providing more objective and unbiased evaluations of candidates. They can analyze the text of resumes, highlighting main qualifications and experiences, and compare them with the employer's requirements. This allows for faster identification of the most suitable candidates with less time and fewer resources required for this process [1].

 ISSN 1560-8956

Moreover, such technologies can help uncover talented specialists who might otherwise go unnoticed during manual resume processing, and they can reduce the likelihood of bias in the selection process.

Analysis of analogues methods

There is a significant amount of scientific research dedicated to the use of various text processing methods in the context of resume analysis [1-8]. Let's consider some methods:

Bag of Words (BoW) is a simple text vectorization method used in many natural language processing tasks. In this model, each sentence or document is treated as a collection of individual words, and the text itself is seen as a "bag of words" disregarding the order of words in the document. Thus, for each document, a vector is created where each value represents the frequency of a specific word occurring in that document.

$$BoW = \text{number of times the term appears in the document} \#(1)$$

One drawback of this method is its strict attention to the frequency of each word occurrence in the text, which can lead to the consideration of unimportant words such as "and," "but," "or," and others.

These words, although insignificant in the context of text content analysis, may still have high frequencies of occurrence in various documents and, accordingly, high values in the vector representation of the text using this method, which can distort the weight and significance of other, more meaningful words in the text.

Term Frequency-Inverse Document Frequency (TF-IDF) [4] is an enhanced method that considers not only the frequency of each word occurrence in the text but also the importance of the word to the text as a whole.

TF-IDF is defined as the product of two components: Term Frequency (TF), which determines how often a word appears in the text, and Inverse Document Frequency (IDF), which indicates how unique the word is across the entire document corpus.

Method allows for considering the likelihood that a word is important to a specific text while also taking into account its overall prevalence in the document collection.

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of therms in the document}} \#(2.1)$$

$$IDF = \log\left(\frac{\text{number of the document in the corpus}}{\text{number of documents in the corpus contain the term}}\right) \#(2.2)$$

$$TF - IDF = \text{TF} * \text{IDF} \#(2.3)$$

In general, both of these methods demonstrate effectiveness in vectorizing textual data for further processing and analysis. However, they have a significant drawback: they do not account for semantic relationships between words. This means that words with similar or

related meanings or used in similar contexts may have radically different vector representations.

The use of these methods results in the inability of the system to recognize equivalence even if a candidate's resume contains skills that essentially match the job requirements but are expressed with different words or terms. For example, a job listing might specify a requirement for "Project management," while the candidate's resume lists skills such as "Agile," "Scrum," "Kanban," "JIRA," "Trello," "Risk Management," "MS Project," or "Asana." This leads to a low rating of relevant resumes because the system fails to recognize that these terms have similar or related meanings. As a result, a candidate who actually possesses the necessary skills for the job may be rejected, and the employer may lose a valuable specialist.

To overcome these limitations, more advanced techniques that capture semantic relationships between words are needed. Methods such as word embeddings, including Word2Vec offer a solution by representing words in continuous vector spaces where semantically similar words are positioned closely together.

Additionally, transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) have further advanced the field by understanding the context of words in a sentence, providing even more accurate semantic matching.

Table 1 provides examples of skills and related skills, illustrating how different terms that essentially mean similar things may not be recognized as such by Bag of Words (BoW) and TF-IDF methods. Therefore, to improve the accuracy and effectiveness of candidate search, it is necessary to employ more modern methods that consider semantic relationships between words.

Table 1.

Skill-possible related skills

Primary Skill	Possible related Skills
Python programming	Django, Flask, Data Analysis, Machine Learning, Pandas, NumPy, SciPy, Automation scripting
Project management	Agile, Scrum, Kanban, JIRA, Trello, Risk Management, MS Project, Asana
Data analysis	SQL, R, Python, Machine Learning, Tableau, Power BI, Excel, Statistics
Web development	HTML, CSS, JavaScript, Frontend, Backend, React, Angular, Node.js, Vue.js
Sales	CRM, Negotiation, Lead Generation, Sales Strategy, Customer Relations, B2B Sales, B2C Sales

 ISSN 1560-8956

Ende der Tabelle 1

Primary Skill	Possible related Skills
Graphic design	Adobe Photoshop, Illustrator, UX/UI Design, InDesign, Branding, Typography, Layout Design
Digital marketing	SEO, PPC, Social Media Marketing (SMM), Email Marketing, Google Analytics, Content Marketing, SEM
Network administration	Cisco, Network Security, Firewall Management, TCP/IP, Network Infrastructure, LAN/WAN, Wireless Networking

BERT (Bidirectional Encoder Representations from Transformers) is an advanced natural language processing model developed by Google, built on the transformer architecture. Due to its bidirectional nature, BERT can more accurately understand the meanings of words in the context of a sentence, making the model extremely effective for various text processing tasks such as classification, question answering, named entity recognition, and sentence comparison [2].

While the model offers many advantages, its large size and high computational requirements make it suboptimal for this research. In resume evaluation, it's often necessary to compare keywords and phrases rather than full sentences or complex textual structures. Therefore, using BERT for this task may be excessive and inefficient.

For such tasks, it is more advisable to use lighter models, such as Word2Vec, which are less resource-intensive and can provide a sufficient level of accuracy.

Word2Vec model

Word2Vec is one of the most popular methods, the main idea of which is to position words in a multi-dimensional space in such a way that semantically similar words have close vector representations. This property allows these vectors to be used for various natural language processing (NLP) tasks, including text classification, semantic similarity analysis, and more.

Word vector representations are obtained by training the model on a large volume of data. During the training process, the model learns to detect similarity between words based on their context, i.e., words that often appear together in sentences.

The operation principle of Word2Vec is based on two main architectures: Continuous Bag of Words (CBOW) and Skip-Gram (Fig. 1):

In the CBOW model, context words are used to predict the target word. This means that the model is trained based on the average vector representation of words surrounding the target word in the text. The input data for CBOW consists of context words (for example, if the target word is "read," the context could be "I love __ books"), and the output data is the target word ("read").

One advantage of CBOW is that it is more efficient and faster to train since it utilizes averaged vectors of context words. However, CBOW is less accurate in handling rare words.

On the other hand, the Skip-Gram model uses the target word to predict context words: for each word in the text, the model attempts to predict the words surrounding it. The input data for Skip-Gram is the target word (for example, "read"), and the output data is the context words (for example, "I love __ books").

Skip-Gram works better with rare words and provides a more accurate vector representation for words that are rarely encountered. However, one drawback of Skip-Gram is that this model is more resource-intensive and slower to train compared to CBOW.

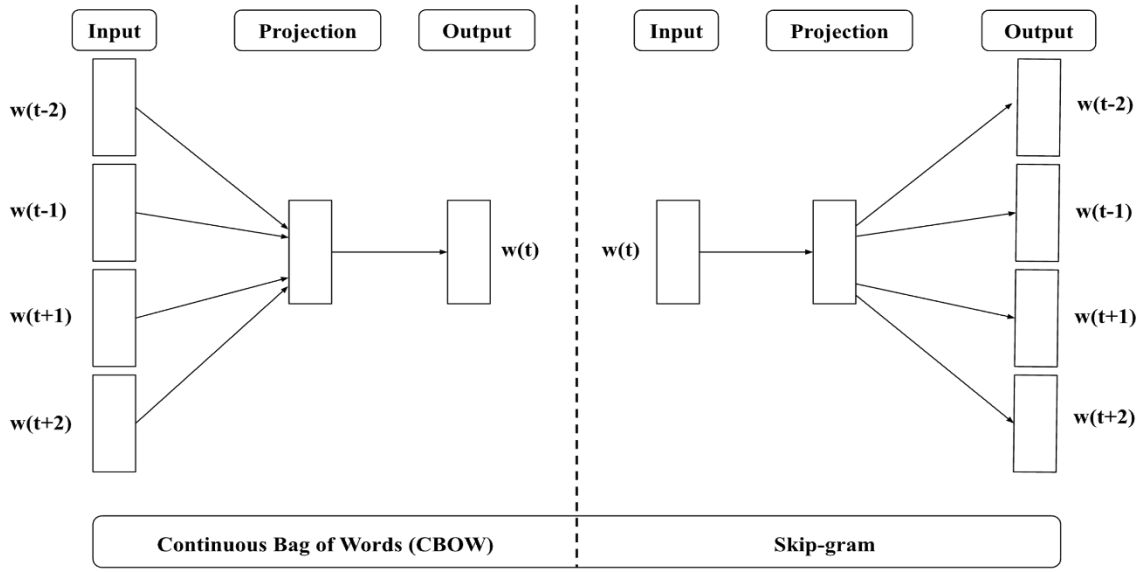

Figure 1. Word2Vec Model

Comparing these two architectures, we can conclude that for matching resumes to job requirements and working with relevant skills, it is better to use Skip-Gram. It works better with rare words and terms, allowing the model to more accurately reflect specialized words that are often encountered in resumes. This is especially important for technical skills and specific terms that may be key to a particular job vacancy.

This formula represents the probability that a given context word $w_{c,j}$ is the correct context for the input word w_I. The model aims to maximize this probability for correct word pairs and minimize it for incorrect pairs.

$$p(w_{c,j} = w_{O,c}|w_I) = \mathrm{y_{c,j}} = \frac{\exp(\mathrm{u_{c,j}})}{\sum_{\mathrm{j}=1}^{\mathrm{V}} \exp(\mathrm{u_j})}, \#(3)$$

where: $w_{c,j}$ - context word at position j; $w_{O,c}$ - expected (correct) context word; w_I - input word; $\mathrm{u_{c,j}}$ - dot product of the vectors of the input word and the context word.

 ISSN 1560-8956

Methodology

To evaluate the effectiveness of the Word2Vec model, cosine similarity measure was used [4]. This method allows determining the degree of similarity between two vectors by measuring the cosine of the angle between them (Fig. 2). The formula for cosine similarity is as follows:

$$cos(\theta) = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}, \#(4)$$

The cosine similarity value varies from -1 to 1, where 1 indicates complete similarity, 0 - no similarity, and -1 - complete dissimilarity.

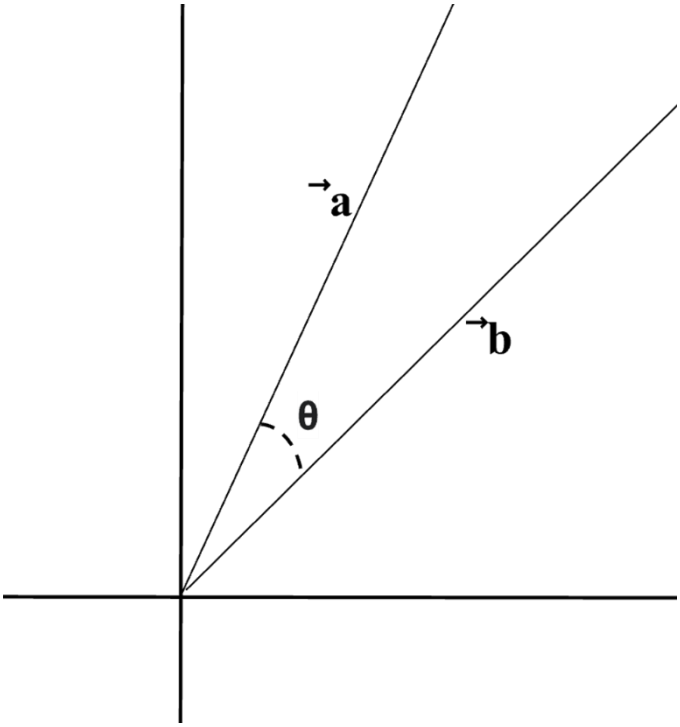

Figure 2. Cosine similarity

Cosine similarity is a valuable metric often used in various fields to measure the similarity between two vectors. Its popularity stems from several key advantages.

Firstly, it is independent of the vector's magnitude, focusing solely on the orientation of vectors in a multi-dimensional space. This property makes it reliable and robust, especially when dealing with high-dimensional data where vector magnitudes can vary significantly.

Secondly, cosine similarity is scale-independent, meaning it remains insensitive to changes in the scale of vector components. This allows for meaningful comparisons between vectors regardless of their size or units of measurement, making it suitable for diverse datasets.

It is particularly suitable for tasks related to textual data, such as document similarity analysis, document clustering, and information retrieval. Its ability to capture semantic similarity between textual documents based on word or document vectors makes it indispensable in natural language processing tasks.

Results

For analysis, one job vacancy with specific requirements and four resumes were taken. The first candidate had a resume that fully matched the job requirements, while the other three candidates had skills directly related to the requirements but did not match exactly; additionally, in some cases, synonymous terms to the requirements were used. Cosine similarity was calculated between the job vacancy vector and the vectors of each resume using TF-IDF and Word2Vec (skip-gram). The results were presented in Table 2.

Table 2.

Results of experiment (TF-IDF vs Word2Vec)

Resume	**Cosine similarity**	
	TF-IDF	**Word2Vec (skip-gram)**
Candidate 1	0.8522	0.8654
Candidate 2	0.3534	0.6340
Candidate 3	0.3328	0.6493
Candidate 4	0.2493	0.7045

The results for the first candidate indicate a high level of compliance with the job requirements for both models: TF-IDF has a cosine similarity value of 0.8522, while Word2Vec slightly outperforms with a score of 0.8654. This indicates that both models effectively recognize exact matches.

For the second candidate, the results show a significant difference between the models: TF-IDF yields a value of 0.3534, indicating low correspondence, while Word2Vec shows a much higher level of similarity – 0.6340. This confirms that Word2Vec is capable of effectively recognizing similar skills and synonyms that the traditional TF-IDF method overlooks.

A similar situation is observed for the third and fourth candidates. The cosine similarity values for TF-IDF are significantly lower, while for Word2Vec, they are higher. This again demonstrates the advantage of Word2Vec.

Thus, the experiment results clearly indicate the advantages of the Word2Vec (skip-gram) model compared to TF-IDF. It has proven to be significantly more effective in cases

 ISSN 1560-8956

where candidates' skills were related or synonymous with the job requirements. This speaks to its ability to better consider contextual and semantic relationships between words, making it more suitable for analyzing unstructured resume data.

Conclusions

The article examines the use of the Word2Vec model to assess the compatibility of resumes with job requirements.

Various neural network training methods were analyzed, demonstrating that using models based on word embeddings provides a more accurate and relevant assessment in the context of unstructured resumes.

During the experiment, it was found that the Word2Vec model demonstrates higher accuracy in cases where synonyms or related terms are used in resumes, which are not recognized by traditional methods. This enhances the efficiency of candidate selection, minimizing the risk of missing qualified candidates who use different wordings to describe their skills.

Thus, the experimental results confirm that the Word2Vec model can significantly improve automated resume screening systems, ensuring a more relevant candidate selection.

The research findings presented in this article may have practical implications for employee selection and the recruitment process as a whole. Therefore, the use of NLP and ML not only streamlines the resume processing process but also makes it more transparent and effective, which is crucial for the successful operation of modern companies.

REFERENCES

1. An automated resume screening system using natural language processing and similarity / c. Daryani et al. *Ethics and information technology*. 2020. URL: https://doi.org/10.26480/etit.02.2020.99.103

2. Pre-trained models for natural language processing: A survey / X. Qiu et al. Science China Technological Sciences. 2020. Vol. 63, no. 10. P. 1872–1897. URL: https://arxiv.org/pdf/2003.08271

3. *Gopalakrishna S. T., Varadharajan V.* Automated Tool for Resume Classification Using Sementic Analysis. *International Journal of Artificial Intelligence & Applications*. 2019. Vol. 10, no. 01. P. 11–23. URL: https://doi.org/10.5121/ijaia.2019.10102

4. A cosine similarity-based resume screening system for job recruitment. *International Research Journal of Modernization in Engineering Technology and Science*. 2023. URL: https://doi.org/10.56726/irjmets35945

5. Resume Screening using Machine Learning and NLP: A proposed system / Bhushan Kinge et al. *International Journal of Scientific Research in Computer Science,*

Engineering and Information Technology. 2022. P. 253–258. URL: https://doi.org/10.32628/cseit228240

6. Resume Parser using Natural Language Processing / Mr. B. Venkata Satish Babu et al. *International Journal of Advanced Research in Science, Communication and Technology*. 2022. P. 161–167. URL: https://doi.org/10.48175/ijarsct-7616

7. Survey on Resume Screening Mechanisms / T. M. Harsha et al. *International Journal of Computer Science and Engineering*. 2022. Vol. 9, no. 4. P. 14–22. URL: https://doi.org/10.14445/23488387/ijcse-v9i4p103

8. A smart resume screening tool for customized shortlisting / P. Tijare et al. *ITM Web of Conferences*. 2023. Vol. 56. P. 04001. URL: https://doi.org/10.1051/itmconf/20235604001