2024 IEEE 9th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 5-7, 2024

2024 IEEE 9th International Conference for Convergence in Technology (I2CT) | 979-8-3503-9447-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/I2CT61223.2024.10543433

Semantic Similarity Analysis for Resume Filtering using PySpark

Akhilesh P, Amal Krishna K, S Karthick Bharadwaj, Manju Venugopalan
Department of Computer Science and Engineering,
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
akhileshnair2003@gmail.com , amalkrishna6003@gmail.com , iamkarthick.puc@gmail.com, v_manju@blr.amrita.edu

***Abstract*— Technology is expanding and applicants are making their resumes look better by developing their skill set for the current environment. Each resume now looks very promising and manually screening resumes and finding out the potential candidate in time consuming and is not an efficient method. In this paper, we propose a method using Natural Language Processing (NLP) and Spark to filter relevant resumes. The work proposes a cosine similarity-based approach to measure text-similarity and hence find worthy candidates based on key information provided by the employer. The implementation of the proposed approach on a Spark platform helps in handling large-scale datasets, enabling efficient semantic similarity analysis in the context of job descriptions. The experiments conclude that the proposed method can efficiently speed up the process in short – listing worthy candidates and can effectively replace the process of hiring managers going through each resume. Our model demonstrates superior performance compared to a non-Spark-based resume filtering system, achieving an average reduction in processing time by 80%, translating to an average runtime decrease from 5 seconds to 1. This efficiency gain is scalable, presenting enhanced performance in candidate shortlisting, especially with larger datasets.**

Index Terms*— Resume filtering, Semantic Analysis, Cosine Similarity, PySpark, Natural Language Processing (NLP).

I. Introduction

The process of recruitment is one of the most critical, yet resource-intensive tasks faced by organizations across diverse industries. The success of any organization heavily relies on hiring skilled and competent individuals who align with the company's values, culture, and objectives. As the job market grows increasingly competitive, recruiters must contend with a deluge of resumes for each job opening. Traditional resume screening methods, which are primarily manual, have proven to be both laborious and error-prone, leading to missed opportunities and inadequate matching of candidates with job requirements. With the rise of Artificial Intelligence (AI) and Natural Language Processing (NLP) technologies, automating the resume screening process has become an attractive proposition for recruiters and HR departments. AI-powered solutions can analyze and compare job descriptions with the content of resumes to identify the best-fitting candidates efficiently. Among various AI techniques, semantic similarity analysis [1,2,3] has emerged as a powerful approach for gauging the relevance and compatibility of job descriptions and resumes.

In this paper, we present a cutting-edge approach that harnesses the capabilities of Apache Spark [4,5,6], a fast and scalable big data processing engine, to perform semantic similarity analysis on job descriptions and resumes. Our model aims to revolutionize the resume filtering process, significantly improving the efficiency and accuracy of candidate selection. The proposed method effectively bridges the gap between the stated requirements in job descriptions and the qualifications and skills presented in resumes, allowing for a more streamlined and intelligent candidate screening process. The traditional resume screening process involves HR personnel manually scanning through resumes to shortlist candidates based on predefined criteria. However, as the number of job applications rises, this approach becomes increasingly untenable, leading to delays in the recruitment process and a higher risk of losing top talent to competitors. Moreover, manual screening is susceptible to human biases and subjectivity, potentially overlooking qualified candidates and compromising the diversity of the workforce.

By employing semantic similarity analysis [7,8,9,10] on job descriptions using Spark, our proposed model addresses these challenges head-on. The system automatically matches job requirements to the skills and qualifications mentioned in the resumes, ranking candidates based on their relevance to the job description. This data-driven approach drastically reduces human intervention, thereby expediting the screening process and eliminating human bias from the equation. Furthermore, the adoption of our proposed model enhances the overall quality of candidate selection. By identifying candidates whose qualifications align more closely with the job description, the model ensures that recruiters shortlist individuals with a higher probability of success in their roles. This improved matching of candidates to job profiles not only optimizes workforce productivity but also significantly reduces the employee turnover rate.

In addition to its impact on individual organizations, the efficient and accurate filtering process introduced by our model has broader implications for the job market. A faster and more precise hiring process translates into a reduced time-to-hire metric, which is crucial in today's fast-paced business landscape. A quicker time-to-hire allows organizations to onboard talent promptly, ensuring that they stay ahead of their competitors in acquiring the best candidates. In summary, our research aims to present a pragmatic solution for the age-old challenge of resume filtering in the recruitment process. By leveraging the power of Apache Spark and semantic similarity analysis, our proposed model promises to streamline the hiring process, enhance candidate selection, and foster greater efficiency and fairness in the job market. In the subsequent sections, we delve into the technical details and implementation of the proposed model, presenting compelling results from our experiments on real-world job data.

Section 2 presents a survey on relevant approaches for arriving at textual semantic similarity. Section 3 explains the proposed approach for resume filtering based on semantic similarity on a PySpark framework. Section 4 presents the results of implementation and finally Section 5 concludes the findings and points to possible future directions.

II. RELATED WORKS

The survey has focused on understanding the relevant approaches for assessing text similarity among document and some works are specifically oriented towards resume filtering application.

Saurabh et al. [11] explores various methods and significance of word embeddings, which are widely used for determining semantic similarity between words and finding similar words and introduces a novel model that generates embedding and predicts similarity between input text. The models are evaluated using SICK dataset. The primary object is to assess and compare different Natural Language Processing methods for calculating semantic similarity scores between various types of texts. Future extensions could involve techniques for comparing similarity between sets of images embedded within distinct documents. Another interesting work [12] explores algorithms and measures to measure semantic similarity, with a focus on the latent semantic analysis (LSA) method. It emphasizes constructing structured models for text, particularly the widely used vector space model. The paper extensively explores text mining technology, including methods and challenges. It also outlines plans to enhance similarity measures and LSA algorithms for online education's e-assessment tasks, aiming to bolster advanced analysis techniques for online assessments in education.

Jitender Sharma et al. [13] focused on automating resume screening using text similarity measures for job applicants. Three measures were tested: Cosine, Sqrt-Cosine, and ISC (Improved Sqrt-Cosine) similarity. This work utilized datasets of real case studies collected from two companies. The ISC measure proved most accurate, mimicking human decision-making. Future research could delve into advanced methods and hybrids for improved automated recruiting. Overall, the study showcased ISC similarity's potential in streamlining candidate selection, aiding recruiters' decision-making. P. Sravanthi et al. [14] propose an unsupervised approach to calculate sentence-level similarities automatically, relying solely on word-level similarities without external knowledge from other ontologies. They discuss various measures, including path-based, information content-based, and feature-based methods. They implement a feature-based approach to compute similarity scores at both the word meaning and definition levels, which are then compared to existing measures to optimize results.

Another engaging work [15] presents a system with three key components: training, matching, and extracting. In the training phase, domain-trained word embeddings are generated. The matching block identifies top candidates by assessing semantic similarity between resumes and job descriptions. The proposed model demonstrates both high accuracy and quick processing time. Niti Khamker et al. [16] introduce an e-recruitment system that enhances HRM operations in the internet age. Using cosine similarity techniques, this system efficiently matches candidate resumes with job descriptions, presenting the degree of similarity to recruiters. In another notable work [17], advanced WordNet-based hierarchy concept tree (HCT) and hierarchy concept graph (HCG) models are created to represent diverse relationships. These models enhance concept similarity calculations independently of external resources, providing flexibility for semantic similarity assessments in structures resembling WordNet. Chirag Daryani et al. [18] propose a Natural Language Processing method to extract information from unstructured resumes, creating concise profiles. Using a vectorization model and cosine similarity, they align resumes with job descriptions, generating ranking scores to identify suitable candidates. This approach enhances recruitment efficiency and ensures fairer candidate evaluations.

In summary, the survey has offered a comprehensive overview of the evolving landscape of text similarity assessment, with a specific focus on its applications in resume filtering and job matching. Various studies have explored an array of methodologies, from word embeddings to latent semantic analysis, highlighting their significance in quantifying semantic similarity between texts. Notably, the ISC similarity measure emerged as a promising tool for automating resume screening, mirroring human decision-making effectively. These research endeavors also shed light on the potential for advanced NLP techniques to revolutionize HR operations in the internet age, enhancing both accuracy and efficiency in candidate selection. Moreover, the innovative use of WordNet-based hierarchy models and vectorization techniques further extends the possibilities for fairer and more efficient candidate evaluations in the recruitment process.

III. PROPOSED METHODOLOGY

In this section, we present our proposed methodology for semantic similarity analysis on job descriptions using Apache Spark which is depicted in Fig.1. Our approach aims to streamline the recruitment process by automatically identifying resumes that best match predefined sets of required keywords or skills. The dataset used for the experiments and the methodology is explained in the following subsections.

A. Dataset

The foundation of our methodology lies in the data preprocessing phase. In the model, this is demonstrated by reading resume data from a CSV file. The dataset used in this study was sourced from Kaggle[1], a renowned platform for data science and machine learning enthusiasts. Kaggle provides a diverse collection of datasets contributed by the data science community and organizations worldwide.

We obtained a dataset comprising 3,447 rows, with four columns: *resumeID*, *resumeStr*, *resume_html*, and *resumeCategory*. For the purpose of our analysis, we focused exclusively on the *resumeStr* column, which contains the textual content of the resumes. The *resume_html* and *resumeCategory* columns, being irrelevant to our objectives, were disregarded. It is worth noting that, on average, the *resumeStr* column contains approximately 548 words per cell. This underscores the richness and depth of textual data available for our semantic similarity analysis. In practice, organizations would gather a comprehensive dataset of job

[1] https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset

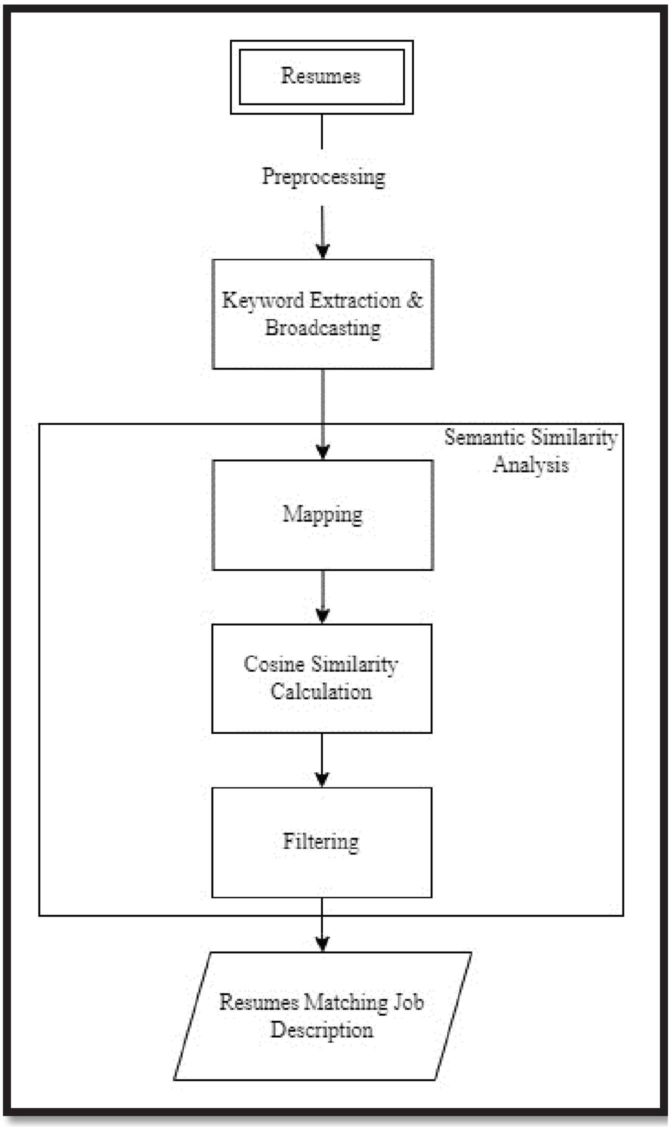
Resumes
Preprocessing
Keyword Extraction & Broadcasting
Semantic Similarity Analysis
Mapping
Cosine Similarity Calculation
Filtering
Resumes Matching Job Description

Fig.1 Work flow of the proposed approach for semantic similarity-based resume filtering

descriptions and candidate resume from various sources, such as job portals or internal databases and that can be incorporated into our existing model with ease.

B. PreProcessing

The textual data (i.e., the text from *resumeStr* column in the dataset) is then subjected to preprocessing steps, including tokenization, lowercasing, and the removal of stop words and special characters. Additionally, stemming is applied to standardize word forms. These preprocessing steps ensure that the textual data is uniform and ready for analysis.

C. Keyword Extraction and Broadcasting

To enable semantic similarity analysis, we define a list of required keywords or skills that are crucial for a particular job opening. In the system, this list is represented as *requiredKeywords*. In practice, this list can be customized for each job vacancy, ensuring that the analysis is tailored to the specific job requirements. The system efficiently broadcasts these required keywords to all worker nodes in the Spark cluster. Broadcasting is an optimization technique that enhances the efficiency of data sharing across distributed systems. It ensures that the required keywords are readily available for semantic similarity calculations on all nodes

D. Semantic Similarity Analysis

The core of our methodology, as illustrated revolves around semantic similarity analysis. The objective is to assess how well each resume matches the predefined set of required keywords. This is achieved through the following steps:

1) **Mapping Resumes**: Each resume is mapped to a key-value pair, where the key is the resume ID, and the value initially represents a default similarity score which is set to 0 in our model.

2) **Calculating Semantic Similarity**: The model then calls a user defined function for each resume. This function computes the similarity between the keywords in the resume and the required keywords. To quantify the semantic similarity between resume texts and the required keywords, we employ the concept of cosine similarity [19], a widely-accepted mathematical metric. This technique enables us to gauge the degree of alignment between a resume's keyword usage and the specified set of required keywords.

a) ***Keyword Frequency Vectors:*** In our approach, Vector A represents the keyword frequency vector derived from a resume text. Each dimension within this vector corresponds to a unique keyword found in the resume, and the value in each dimension signifies the frequency of that keyword within the resume. Vector B, on the other hand, embodies the keyword frequency vector based on the required keywords. Each dimension in this vector corresponds to a unique keyword essential for the job role, with values indicating the frequency of each required keyword.

$$CS(A,B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 \cdot \sum_{i=1}^{n} B_i^2}} \quad (1)$$

where:
CS(A,B) represents cosine similarity between the two vectors A and B.
A is the vectorized form of text from the resumeStr column
B is the vectorized form of required keywords
(A · B) represents the dot product of vectors A and B.
||A|| represents the magnitude (Euclidean norm) of vector A.
||B|| represents the magnitude (Euclidean norm) of vector B.

b) ***Calculation of Cosine Similarity:*** The core calculation behind the cosine similarity (1) lies in the dot product (A · B) of these two keyword frequency vectors. This computation sums the products of corresponding dimensions of the two vectors, essentially quantifying the similarity between the keywords present in the resume and the required keywords for the job. To gauge the relevance of these vectors, we also calculate the magnitudes (or lengths) of Vector A (||A||) and Vector B (||B||). The magnitude of each vector represents the "length" of the respective keyword frequency vector, taking into account the frequency distribution of keywords.

The final cosine similarity value (θ) is then computed by dividing the dot product (A · B) by the product of the magnitudes (||A|| * ||B||) (1). This mathematical operation normalizes the similarity score to a range from -1 (indicating complete dissimilarity) to 1 (representing complete similarity), with 0 signifying no similarity.

*3) **Filtering:*** In our model's implementation, we use this cosine similarity metric to filter and identify resumes that exhibit compatibility with a specific job posting. Resumes with a cosine similarity score exceeding a predefined threshold of 0.4 are considered as potential matches, aligning with the job's keyword requirements.The filtering treshold can be customized according to the user's requirements. This scalable and robust approach, anchored in cosine similarity analysis, stands as the cornerstone of our approach to automate and optimize the resume filtering process, ensuring that candidate qualifications align closely with job descriptions.

IV. Result & Analysis

The compatible resume IDs, obtained from the filtering step, are collected into a list. These IDs are then printed to the console, enabling quick and easy identification of resumes that closely match the job's required keywords. Our methodology leverages the power of Apache Spark to perform distributed computing, making it well-suited for processing large volumes of resumes efficiently. Spark's parallel processing capabilities enable the analysis of diverse datasets, making it adaptable to organizations with varying recruitment needs.

As part of our comprehensive methodology, it is crucial to address the evaluation of the semantic similarity analysis model. Traditionally, this evaluation encompasses assessing key metrics such as accuracy, precision, recall, and the F1-score to gauge the model's effectiveness in matching resumes with job descriptions. But the focus of our work was to contribute a scalable model which even works well with large volumes of resumes which would be an ideal realistic scenario.

Our dataset poses a unique challenge due to its substantial size, consisting of 3,447 resumes. Given the sheer volume of data, manual verification or manual assessment of results becomes practically unfeasible within reasonable timeframes. Consequently, traditional evaluation metrics that rely on manually verified ground truth labels may not be directly applicable. In light of this challenge, our evaluation strategy importance of exploring this alternative evaluation strategy. Our alternative approach centers on comparing the runtime of

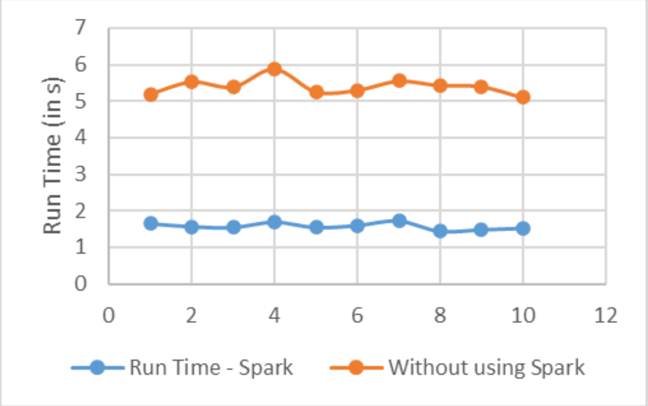

Fig.2 Run Time differences of our model on a PySpark platform vs a sequential execution

prioritizes robustness and scalability. While manual verification may be unattainable, we emphasize the our model, which harnesses the distributed processing capabilities of Apache Spark, against the runtime of a standard Python program running the same task without utilizing Spark's functionality which is showcased in Fig.2. This comparison is visually presented in Fig.2, which illustrates the runtime differences between our Spark-based model and a comparable Python model without Spark integration. To provide comprehensive insights, we conducted this comparison across 10 distinct test cases, each involving a change in the required keywords as input to the models. The runtime comparison serves as a benchmark for evaluating the efficiency of our Spark-based semantic similarity analysis model. Our model significantly outperforms a standard Python program in terms of runtime, hence demonstrating the value of leveraging distributed computing for this task.

V. Conclusion & Future scope

In this paper, we have introduced a scalable and robust approach for semantic similarity analysis on job descriptions using Apache Spark, with a specific focus on its application in resume filtering. Our approach offers an automated and data-driven solution to the challenges associated with the manual screening of resumes, significantly improving the efficiency and effectiveness of candidate selection. Our model, as implemented, has showcased its potential to revolutionize the hiring process across industries. By automating the intricate task of matching candidate qualifications to job requirements, we have significantly reduced the time and resources traditionally invested in this laborious process. The scalability and robustness of our solution, underpinned by Apache Spark, ensure that organizations can seamlessly handle the ever-increasing volumes of job applications. Furthermore, our model introduces a critical dimension of objectivity into the hiring process. The elimination of inherent human biases and subjectivity is a crucial advancement in the pursuit of fair and equitable recruitment practices. By relying on data-driven decision-making, our model levels the playing field, ensuring that candidates are evaluated solely on the basis of their qualifications and merits.

While our proposed methodology represents a significant advancement in resume filtering, there are several avenues for future research and improvement. While cosine similarity has been the cornerstone of our model, future investigations could delve into alternative similarity metrics, including Jaccard similarity, Dice coefficient, or Mahalanobis distance. A comparative study of strengths and weaknesses. Deep learning models have proven transformative in various NLP tasks. Integrating deep learning architectures, such as Siamese networks or transformer-based models, into our framework could further enhance the precision and granularity of similarity analysis. In an increasingly globalized job market, extending our model to support multiple languages and cross-lingual semantic similarity analysis is both challenging and promising. Such an extension would facilitate diverse and inclusive hiring practices. In conclusion, while our proposed methodology has already introduced significant improvements to resume filtering, these future directions, including the potential integration of word embeddings, offer opportunities for further refinement and expansion. By embracing these challenges, we can continue to enhance the capabilities of our model and provide organizations with even more effective tools for talent acquisition in the evolving job market, potentially leading to increased accuracy in candidate selection.

REFERENCES

[1] Muppavarapu, Vamsee, et al. "Knowledge extraction using semantic similarity of concepts from Web of Things knowledge bases." Data & Knowledge Engineering 135 (2021): 101923.

[2] Sajeev, G. P., and P. T. Ramya. "Effective web personalization system based on time and semantic relatedness." 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2016.

[3] Gupta, Deepa, K. Vani, and Charan Kamal Singh. "Using Natural Language Processing techniques and fuzzy-semantic similarity for automatic external plagiarism detection." 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2014.

[4] Chandrasekaran, Dhivya, and Vijay Mago. "Evolution of semantic similarity—a survey." ACM Computing Surveys (CSUR) 54.2 (2021): 1-37.

[5] Miller, George A., and Walter G. Charles. "Contextual correlates of semantic similarity." Language and cognitive processes 6.1 (1991): 1-28.

[6] Salloum, Salman, et al. "Big data analytics on Apache Spark." International Journal of Data Science and Analytics 1 (2016): 145-164.

[7] Zaharia, Matei, et al. "Apache spark: a unified engine for big data processing." Communications of the ACM 59.11 (2016): 56-65.

[8] Saravanan, S. "Performance evaluation of classification algorithms in the design of Apache Spark based intrusion detection system." 2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2020.

[9] Menon, Remya RK, S. Gargi, and S. Samili. "Clustering of words using dictionary-learnt word representations." 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2016.

[10] Ramachandran, Raji, Dhiti P. Nair, and J. Jasmi. "A horizontal fragmentation method based on data semantics." 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). IEEE, 2016

[11] Saurabh Agarwala, Aniketh Anagawadi , Ram Mohana Reddy Guddeti. "Detecting Semantic Similarity of documents using Natural Language Processing."Procedia Computer Science 189 (2021): 128-135.

[12] Rozeva, Anna, and Silvia Zerkova. "Assessing semantic similarity of texts–methods and algorithms." AIP Conference Proceedings. Vol. 1910. No. 1. AIP Publishing, 2017.

[13] Alsharef, Ahmad, Hasan Nassour, and Jitender Sharma. "Exploring the Efficiency of Text-Similarity Measures in Automated Resume Screening for Recruitment." 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2023.

[14] Sravanthi, Pantulkar, and B. Srinivasu. "Semantic similarity between sentences." International Research Journal of Engineering and Technology (IRJET) 4.1 (2017): 156-161.

[15] Najjar, Arwa, Belal Amro, and Mário Macedo. "An intelligent decision support syste for recruitment: resumes screening and applicants ranking." Informatica 45.4 (2021).

[16] Khamker, Niti, Y. Khamker, and M. Butwall. "Resume match system." International Journal Of Innovative Science And Research Technology (2021).

[17] Liu, Hongzhe, Hong Bao, and De Xu. "Concept vector for semantic similarity and relatedness based on wordnet structure." Journal of Systems and software 85.2 (2012): 370-381.

[18] Daryani, Chirag, et al. "An automated resume screening system using natural language processing and similarity." ETHICS AND INFORMATION TECHNOLOGY [Internet]. VOLKSON PRESS (2020): 99-103.

[19] Rahutomo, Faisal, Teruaki Kitasuka, and Masayoshi Aritsugi. "Semantic cosine similarity." The 7th international student conference on advanced science and technology ICAST. Vol. 4. No. 1. 2012.