**KPI RETRIEVAL APPROACHES**

**1 Introduction**

Key Performance Indicators (KPIs) play a crucial role in tracking and evaluating the success of various processes across industries. However, retrieving relevant KPIs from unstructured text poses a significant challenge due to variations in terminology, phrasing, and context. This document presents multiple **Natural Language Processing (NLP)** approaches for automated KPI retrieval, each evaluated based on accuracy, robustness, and computational efficiency.

The primary objective of this study is to explore different methodologies—from **Named Entity Recognition (NER) and Fuzzy Matching** to more advanced **TF-IDF, Word Embeddings, and Semantic Similarity Techniques**—to determine the most effective method for KPI extraction. Each approach is thoroughly documented with its **workflow, results, and performance ranking**, ensuring a clear understanding of its strengths and limitations.

Through this comparative analysis, we identify the most suitable technique for **enhancing KPI retrieval accuracy**, enabling organizations to streamline their decision-making processes using **AI-driven solutions**.

**2 Approaches to solve the problem**

**2.1 NER + Fuzzy Matching**

This approach extracts KPI-related entities from text using **Named Entity Recognition (NER)** and applies **Fuzzy Matching** to find the closest matching KPI. It works well for **exact or near-matching entities** but struggles with variations in phrasing and synonyms.

**Workflow**

**Step 1: Install Required Libraries**

* Install and import **spaCy** (for NER) and **fuzzywuzzy** (for fuzzy matching).

**Step 2: Load Pre-Trained spaCy NER Model**

* Use an existing **English language model (e.g., en\_core\_web\_sm)** to extract entities.

**Step 3: Process Input Text**

* Pass the input text through the **NER pipeline** to extract relevant entities (e.g., KPIs).

**Step 4: Apply Fuzzy Matching**

* Use **fuzzy string matching** to compare extracted entities against a predefined KPI list.
* Compute a similarity score for each KPI candidate.

**Step 5: Select the Best Matching KPI**

* Retrieve the KPI with the **highest similarity score**.

**Results**

* **Performance Ranking:** **3rd**
* Works well for structured text but fails when the KPI is expressed using synonyms or paraphrases.

Test Case 1: Worked

Query: how many patents granted for CAD in 2024 without limit?

KPI Retrieved: Patents Granted, Patents Filed

Original KPI: Patent Granted

Test Case 2: Did not work

Query: what is the applications service labor productivity ratio for DJVC for July 2018?

KPI Retrieved: Number of Generated Lesson Learn Generation(12.1), Number of Generated Lesson Learn Generation(11.1), Number of Generated Lesson Learn Generation(12.3)

Original KPI:

[**Notebook Link**](https://chatgpt.com/mnt/data/KPI_using_NER_+_Fuzzy_Matching.ipynb)**:**

**2.2 Fine-tuned NER + Fuzzy Matching**

This is an **enhanced version** of Approach 1, where the spaCy NER model is **fine-tuned** using a KPI-specific dataset. The custom-trained model improves entity recognition, leading to **better fuzzy matching results**. Unfortunately, due to the overfitting of the fine-tuned model to the list of unique KPI’s, it did not yield better results than the former model.

**Workflow**

**Step 1: Install and Import Dependencies**

* Import spaCy, fuzzywuzzy, and dataset handling libraries.

**Step 2: Load KPI Dataset**

* Use a dataset containing KPI terms and their variations.
* The dataset is annotated with KPI-specific entity labels.

**Step 3: Train a Custom NER Model**

* Fine-tune spaCy’s model using **custom annotations**.
* Train the model over multiple iterations for improved accuracy.

**Step 4: Extract KPI-related Entities**

* Apply the fine-tuned model on input text to identify KPI names.

**Step 5: Perform Fuzzy Matching**

* Compare extracted entities with the predefined KPI database using **fuzzy matching**.

**Step 6: Retrieve the Best KPI Match**

* Select the best-matching KPI using **fuzzy similarity scores**.

**Results**

* **Performance Ranking:** **5th**
* Improved entity extraction but still limited by **fuzzy matching** constraints.

**Test Case 1: Worked**

Query: how many patents granted for CAD in 2024 without limit?

KPI Retrieved: Patents Granted, Patents Filed

Original KPI: Patent Granted

**Test Case 2: Did not work**

Query: what is the applications service labor productivity ratio for DJVC for July 2018?

KPI Retrieved: Number of Generated Lesson Learn Generation(12.1), Number of Generated Lesson Learn Generation(11.1), Number of Generated Lesson Learn Generation(12.3)

Original KPI: Saudi Service Contractors: 0.33, Service Contractor Saudization: 0.35, Service Delivery Customer Satisfaction (): 0.32

**Test Case 2: Did not work**

Query: List the kpis present in cad?

KPI Retrieved: Female Representation

Original KPI: No matching KPIs

[**Notebook Link**](https://chatgpt.com/mnt/data/KPI_using_Fine_tuned_NER_+_Fuzzy_Matching.ipynb)

**2.3 TF-IDF + Cosine Similarity**

**Description**

This approach uses **TF-IDF (Term Frequency-Inverse Document Frequency)** to convert text into numerical vectors and then applies **cosine similarity** to find the closest KPI.

**Workflow**

**Step 1: Install and Import Dependencies**

* Use scikit-learn for **TF-IDF vectorization** and **cosine similarity computation**.

**Step 2: Load KPI Dataset**

* The dataset contains **KPI names and descriptions**.

**Step 3: Preprocess Text**

* Convert text to **lowercase**.
* Remove **stopwords** and **punctuation**.
* Tokenize words into meaningful units.

**Step 4: Compute TF-IDF Vectors**

* Convert KPI descriptions into **TF-IDF vectors**.

**Step 5: Compute Cosine Similarity**

* Compute the similarity between **input text** and **KPI dataset**.
* Rank KPIs based on similarity scores.

**Step 6: Return the Best Match**

* Select the KPI with the **highest cosine similarity score**.

**Results**

* **Performance Ranking:** **2nd**
* Captures textual similarities well but **fails with synonyms**.

**Test Case1: Worked**

Query: How about Test Phishing Email Failure?

KPI Retrieved: Number of Phishing Test Recipients: 0.33, Test Phishing Email Failure: 0.75, Test Phishing Email Failure (Repeated Violators): 0.56

**Test Case2: Did not work (Cannot match synonyms of KPIs like labor – manpower)**

Query: what is the applications service labor productivity ratio for DJVC for July 2018

KPI Retrieved: No matching KPI found

Original KPI: Saudi Service Contractors: 0.33, Service Contractor Saudization: 0.35, Service Delivery Customer Satisfaction (): 0.32

[**Notebook Link**](https://chatgpt.com/mnt/data/KPI_Retrieval_using_TF_IDF_and_Cosine_Similarity.ipynb)

**Approach 4: TF-IDF + Cosine Similarity + Word2Vec**

This approach builds upon TF-IDF similarity by integrating **Word2Vec embeddings** to capture semantic relationships between words. Although, it was able to find semantic relationships, the model gets confused and retrieved erroneous KPI at times.

**Workflow**

**Step 1: Install and Import Dependencies**

* Use scikit-learn for **TF-IDF** and gensim for **Word2Vec**.

**Step 2: Load KPI Dataset**

* Contains KPI names and descriptions.

**Step 3: Preprocess Text**

* Convert text to lowercase.
* Tokenize text into words.
* Remove stopwords and punctuation.

**Step 4: Compute TF-IDF Vectors**

* Transform KPI text into **TF-IDF vectors**.

**Step 5: Generate Word2Vec Embeddings**

* Train Word2Vec on KPI descriptions.
* Convert words into dense vector representations.

**Step 6: Compute Combined Similarity Score**

* Compute **cosine similarity** between input text and KPI dataset.
* Weight both **TF-IDF similarity** and **Word2Vec similarity**.

**Step 7: Retrieve the Best KPI Match**

* Select the KPI with the **highest combined score**.

**Results**

* **Performance Ranking:** **4th**
* Captures **semantic relationships**, but **requires extensive training**.

**Test Case1: Worked**

Query: What is the total number of Traffic Violations across all the periods in 2024?

KPI Retrieved: Number of Traffic Violations: 0.58, Traffic Violations: 0.50, Traffic Violations – AVL: 0.43

**Test Case2: Did not work**

Query: what is the applications service labor productivity ratio for DJVC for July 2018

KPI Retrieved: Headcount Tracked for USB Waiver: 0.41, Manpower Tracked for Certification- Competency: 0.39, Total Service Contractors: 0.31

Original KPI:

[**Notebook Link**](https://chatgpt.com/mnt/data/KPI_Retrieval_using_TF_IDF_and_Word2Vec_(Word_Embeddings).ipynb)

**Approach 5: TF-IDF + Cosine Similarity + WordNet (Synonyms)**

This approach improves upon TF-IDF similarity by **leveraging WordNet** to incorporate **synonyms**, enhancing the model’s ability to match KPI descriptions.

**Workflow**

**Step 1: Install and Import Dependencies**

* Use NLTK for **WordNet synonym extraction**.

**Step 2: Load KPI Dataset**

* Contains KPI terms and descriptions.

**Step 3: Preprocess Text**

* Convert text to lowercase.
* Tokenize text into words.
* Remove stopwords and punctuation.
* **Expand words using WordNet synonyms**.

**Step 4: Compute TF-IDF Vectors**

* Convert text into **TF-IDF vectors**.

**Step 5: Compute Cosine Similarity**

* Compute similarity between **expanded KPI descriptions** and input text.

**Step 6: Retrieve the Best KPI Match**

* Select the KPI with the **highest similarity score**.

**Results**

* **Performance Ranking:** **1st**
* Best accuracy due to **synonym recognition**.

**Test Case1: Worked**

Query: Which group in CAD has the highest % of women representation?

KPI Retrieved: Female Representation: 0.40

**Test Case2: Worked**

Query: what is the applications service labor productivity ratio for DJVC for July 2018?

KPI Retrieved: Saudi Service Contractors: 0.33, Service Contractor Saudization: 0.35, Service Delivery Customer Satisfaction (): 0.32

[**Notebook Link**](https://chatgpt.com/mnt/data/KPI_Retrieval_using_TF_IDF_and_Cosine_Similarity_with_WordNet.ipynb)

| **Approach** | **Ranking** | **Strengths** | **Weaknesses** |
| --- | --- | --- | --- |
| **NER + Fuzzy Matching** | 3rd | Simple, efficient | Fails with paraphrased KPIs |
| **Fine-tuned NER + Fuzzy Matching** | 5th | More accurate entity recognition | Still depends on fuzzy matching |
| **TF-IDF + Cosine Similarity** | 2nd | Good for exact matches | Cannot handle synonyms |
| **TF-IDF + Word2Vec** | 4th | Captures contextual meaning | Requires large dataset |
| **TF-IDF + WordNet** | 1st | Best accuracy with synonyms | Computationally intensive |