Exploring Naive Bayes Classifiers for Tabular Data to Knowledge Graph Matching

Brice Foko[1], Azanzi Jiomekong[1], Hippolyte TAPAMO[1], Jérémy Buisson[2] and Sanju Tiwari[3]

[1] *Department of Computer Science, University of Yaounde I, Yaounde, Cameroon*
[2] *CREA, Ecole de l'Air et de l'Espace*
[3] *Universidad Autonoma de Tamaulipas, Mexico, India*

Abstract

The present research investigates the use of Naive Bayes classifiers to match knowledge graphs and tabular data, with particular emphasis on Column Type Annotation, Cell Entity Annotation, Column Property Annotation and Table Topic Detection. Using feature extraction techniques such as number of co-occurrences and term frequency, the study evaluates the effectiveness and performance of Naive Bayes classifiers on a variety of datasets. The proposed method is straightforward and generic, making a contribution to the field of knowledge graph matching and demonstrating the potential of Naive Bayes classifiers for the integration and interoperability of tabular data and knowledge graphs.

Keywords

Tabular Data, Knowledge Graph, Tabular data to Knowledge Graph Matching, Naive bayes, TSOTSATable system

1. Introduction

The exponential growth of digital information has made both structured and unstructured data increasingly prevalent. Among structured data, tabular datasets play a crucial role in organizing and presenting information in a structured format across various domains such as digital libraries [1], food science and nutrition [2], etc. On the other hand, Knowledge Graphs (KGs) such as Wikidata[1] and DBpedia[2] provide comprehensive representations of real-world entities and their interconnections. Therefore, matching tabular data with knowledge graphs enables the enrichment of tabular datasets with semantic annotations and links to external knowledge sources, resulting in enhanced data integration, interpretation, and interoperability. However, achieving accurate and efficient matching poses significant challenges due to the heterogeneity and complexity inherent in both tabular data and knowledge graphs.

SemTab'23: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching 2023, co-located with the 22nd International Semantic Web Conference (ISWC), November 6-10, 2023, Athens, Greece

fokobrice3@gmail.com (B. Foko); fidel.jiomekong@facsciences-uy1.cm (A. Jiomekong); hippolyte.tapamo@facsciences-uy1.cm (H. TAPAMO); jeremy.buisson@ecole-air.fr (J. Buisson); tiwarisanju18@ieee.org (S. Tiwari)
0000-0002-0877-7063 (A. Jiomekong)

CEUR Workshop Proceedings (CEUR-WS.org)

[1] https://www.wikidata.org/
[2] https://www.dbpedia.org/

Naive Bayes classifiers are proven to be effective in various classification tasks due to their simplicity, efficiency, and robustness [3]. This study investigates the potential of Naive Bayes classifiers in addressing four primary tasks proposed by the SemTab 2023 challenge: Column Type Annotation (CTA), Cell Entity Annotation (CEA), Column Property Annotation (CPA), and Table Topic Detection (TTD). The aim is to uncover new insights and practical techniques to enhance the alignment and integration of tabular data with KGs. The source code used in this work is available under open source license on GitHub[3]. We also provided a document[4] demonstrating how to use the system proposed to solve the SemTab tasks.

The rest of this paper is organized as follow: Section 2 presents some related work on table annotations, Section 3 presents Naive Bayes classifier, Section 4 gives an overview of the research methodology, Section 5 presents how we processed to solve the different tasks of the challenge, Section 6 presents the results and finally, Section 7 conclude the paper.

2. Related work

The SemTab Challenge is an annual competition that evaluates table annotation systems. It requires understanding the semantics of tabular data and knowledge graphs. Its previous editions introduced three tasks: **CTA Task**, which assigns a semantic type from a KG to a table column; **CEA Task**, which matches a cell of a given table to a KG entity; and **CPA Task**, which assigns a KG property to the relationship between two columns. These tasks have been addressed by different systems using various approaches, including:

- bbw (boosted by wiki) [4]. It uses Wikidata KG to annotate CSV tables using Meta-lookup on a locally-deployed SearX metasearch engine and contextual matching. For contextual matching, exact matching is used, followed by case-insensitive matching if no results are found, and string matching with edit distance.
- MTAB [5]. It handles CTA, CEA and CPA tasks well, using a probabilistic graph model. It improves matching by using multiple services like DBpedia Lookup, DBpedia endpoint, Wikipedia, Wikidata, and a cross-lingual matching strategy, enhancing the overall efficiency.
- DAGOBAH [6]. It assumes that the entities are closed in the embedding space, and employs an embedding strategy to cluster and score them in a column. For entity clustering, it employs pre-trained Wikidata embeddings.
- TSOTSATable system [7]. Introduced by us in the previous challenge, it aims to match tabular datasets to Knowledge Graphs or ontologies, specifically applying it to TSOTSATable datasets [2]. It proposes a KG refinement approach to address the matching problem between tabular data and KGs, focusing on error correction and completing tabular data with missing entities and relations.

Despite the large number of annotation systems available, they are often complicated to implement. This year, we focus on implementing a highly flexible machine learning method, exploring Naive Bayes classification on table to KG matching problems.

[3] https://github.com/fokobrice3/STProbClass/tree/main/MNB_2023
[4] https://github.com/fokobrice3/STProbClass/blob/main/GUIDE.pdf

3. Naive Bayes classifier

A Naive Bayes classifier is a simple probabilistic classification method based on Bayes' theorem, which calculates the probability of a specific class based on observed features. For any occurrences of A and B, the Bayes' theorem asserts the rule given by the equation 1.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{1}$$

- P(A|B): this is the probability of event A given that event B has occurred.
- P(B|A): this is the likelihood of observing evidence B if the event A is true.
- P(A): this is the initial belief or knowledge about the probability of A before considering any evidence.
- P(B): represents the overall probability of observing B, regardless of the occurrence of event A.

Bayes' theorem is useful for inferring causes from their effects, as it simplifies determining the likelihood of an effect based on its presence or absence [8]. We used this theorem as our background in multiclass-classification of tokens on labels, where the predicted class is determined by multiplying the prior probability of each class with the conditional probabilities of each feature.

4. Research methodology

This section describes the research methodology employed in this work. We developed the research methodology by relying on what we know in empirical research in software engineering [9, 10]. The methodology is adapted to the specific tasks and datasets at hand, including CTA, CEA, CPA, and the recently introduced Table Topic Detection (TTD). In the following paragraphs, we present the research question and the empirical research method used in this research.

4.1. Research question

The research question: "How to use Naive Bayes classifiers to match tabular datasets to knowledge graph?" was used as the guideline of this work. To reply to this question, one should provide a system that accepts a tabular dataset and a set of labels (classes or properties from a knowledge graph) as inputs, and produces the annotated dataset with these labels. To this end, the following questions should be replied:

- How can a column of tabular data be classified using a knowledge graph class? This task is known as CTA and the fundamental query is "Which features must be used to classify a column with a label?"
- How can data from a tabular data cell be classified using knowledge graph entities? The CEA is presented here. The fundamental query is "Which features must we use to classify a cell with a label?"

- How can a relationship between two columns of tabular data be classified using knowledge graph property? This task is known as CPA. The fundamental query is "Which features must we use to classify a relationship with a label?"
- How can knowledge graph class be used to classify the topic of tabular data ? This is the TTD. The fundamental query is "Which features must we use to identify a topic with a label?"

4.2. Empirical methods

The research methodology combines case study research, action research, and experimental research, three empirical research techniques used in software engineering [11]. This involves investigating, testing, evaluating potential solutions, and proposing a solid solution that can be applied to annotate any tabular data with a KG entity, class or property. Actually, the SemTab organizers gave us three case studies to use in order to solve the tabular data to KG matching including:

- Annotation of WikidataTables[5] using Wikidata,
- Annotation of tFood[6] using Wikidata [12],
- Annotation of SOTAB[7] using Schema.org and DBpedia.

To enable the proposed solution to be applied in any situation, it is important to gain a deeper understanding of the tabular data used in the knowledge graph matching problem through the study of these case studies. We used the Scrum process, we ran, and improved our system using the Sprint, iterative and incremental practices.

5. Solving the SemTab challenge

The exploration of the use of Naive Bayes classifiers to solve the SemTab challenges tasks allowed us to come up with a generic pipeline presented by Fig 1. This pipeline involves data pre-processing, feature extraction, classification of known labels and prediction of new labels components. In the following paragraphs, we present the different components of this pipeline and the implementation of the system.

5.1. Data Preprocessing

The first step in our approach is data pre-processing, which prepares the raw datasets for training and matching. For each dataset (test and train), the pre-processing steps consist of:

1. Remove special characters,
2. Set characters to lowercase,
3. Remove stopping word,

[5] https://zenodo.org/record/7829583
[6] https://zenodo.org/record/7828163
[7] http://webdatacommons.org/structureddata/sotab/

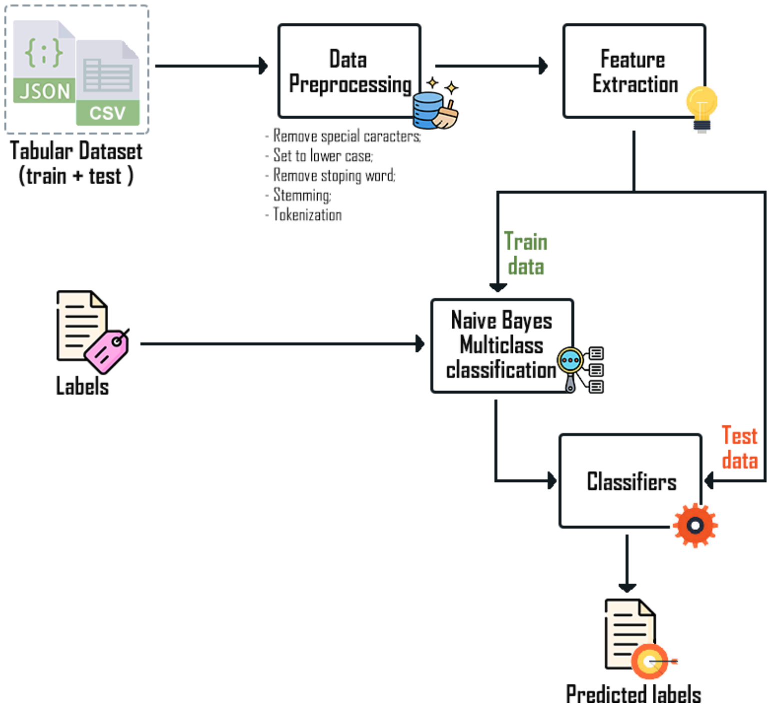


Figure 1: Pipeline of the annotation process

4. Stemming with Porter stemmer,
5. Tokenization.

Once processed, the dataset contains cleaned data that can be used for feature extraction. The training datasets were provided by the SemTab organizers and are presented in the table 1:

Table 1
Statistics of training datasets

Datasets	Tables	CTA targets	CEA targets	CPA targets	TTD targets
WikidataTables	500	623	4247	710	-
tFood (entity)	849	-	2265	3437	849
tFood (horizontal)	438	1089	24951	2084	438
SOTAB-Round1	42733	55360	-	52424	-
SOTAB-Round2 (SCH)	71337	115562	-	97967	-
SOTAB-Round2 (DBP)	60536	85561	-	62128	-

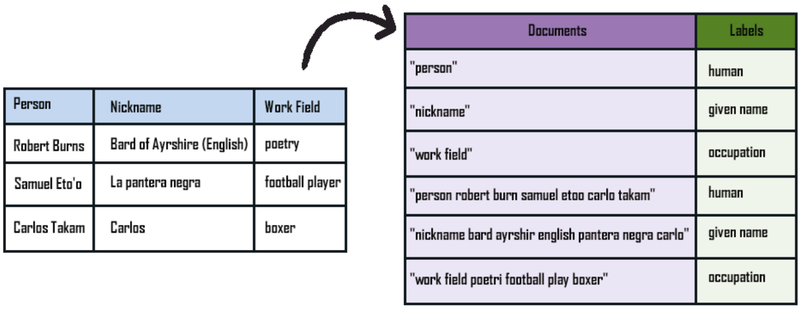
5.2. Feature Extraction

After data pre-processing, we proceed to feature extraction to transform the data into suitable formats for Naive Bayes classifiers. The specific features extracted vary depending on the task being addressed.

5.2.1. Column Type Annotation

For the CTA task, features are extracted from the table columns, including column headers when taken into account, and column descriptions. During the training phase, these features will be connected to the annotation (class/label) that was taken from the training data as presented by Fig. 2.

- **Column Headers:** names of the columns,
- **Column Descriptions:** a bag of words/tokens related to the column content.



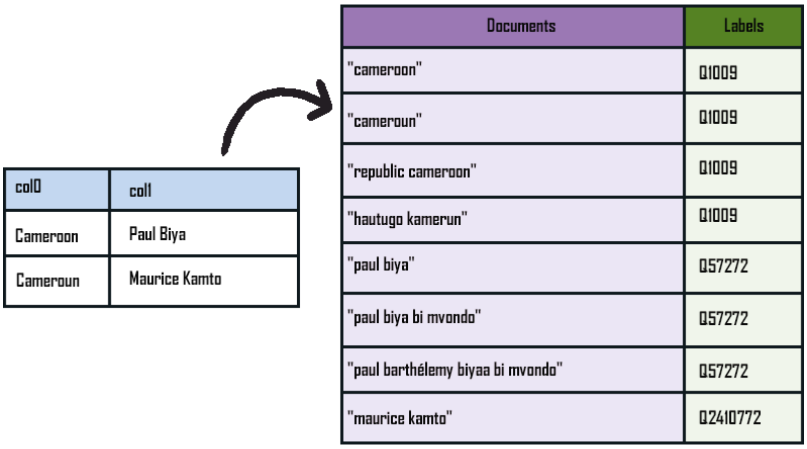
Person	Nickname	Work Field
Robert Burns	Bard of Ayrshire (English)	poetry
Samuel Eto'o	La pantera negra	football player
Carlos Takam	Carlos	boxer

Documents	Labels
"person"	human
"nickname"	given name
"work field"	occupation
"person robert burn samuel etoo carlo takam"	human
"nickname bard ayrshir english pantera negra carlo"	given name
"work field poetri football play boxer"	occupation

Figure 2: Example of feature extraction for the CTA task during the training phase

5.2.2. Cell Entity Annotation

Concerning the CEA task, features are extracted from both the tabular data and the knowledge graph to capture the relevant information for aligning each cell with the appropriate entity. These features include cell contents, entity labels and contextual information that provide additional context for matching as presented by Fig. 3.



col0	col1
Cameroon	Paul Biya
Cameroun	Maurice Kamto

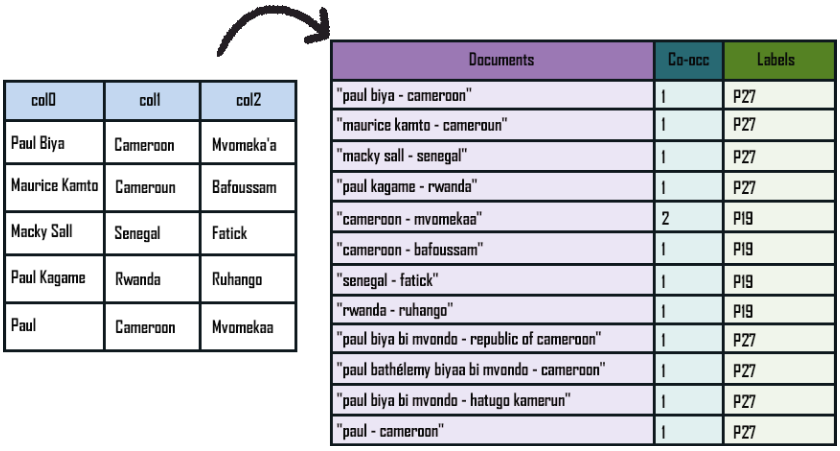
Documents	Labels
"cameroon"	Q1009
"cameroun"	Q1009
"republic cameroon"	Q1009
"hautugo kamerun"	Q1009
"paul biya"	Q57272
"paul biya bi mvondo"	Q57272
"paul barthélemy biyaa bi mvondo"	Q57272
"maurice kamto"	Q2410772

Figure 3: Example of feature extraction for the CEA task during the training phase

5.2.3. Column Property Annotation

For the CPA task, features are extracted to capture the relationships or properties between two columns in the tabular data. These features provide insights into the associations or connections that exist between the columns. Examples of features for CPA (presented by Fig. 4) may include:

- **Co-occurrence count:** which is the number of times specific values or combinations of values appear together in the two columns. This gives us information about the likelihood or frequency of values appearing in both columns at the same time. Only the tokens with a co-occurrence $>=1$ are taken into account.
- **Relationship:** relationships are obtained by linking pairs of tokens together. This consists of combining tokens in the different cells of the target columns in the training dataset for the CPA task.



col0	col1	col2
Paul Biya	Cameroon	Mvomeka'a
Maurice Kamto	Cameroun	Bafoussam
Macky Sall	Senegal	Fatick
Paul Kagame	Rwanda	Ruhango
Paul	Cameroon	Mvomekaa

Documents	Co-occ	Labels
"paul biya - cameroon"	1	P27
"maurice kamto - cameroun"	1	P27
"macky sall - senegal"	1	P27
"paul kagame - rwanda"	1	P27
"cameroon - mvomekaa"	2	P19
"cameroon - bafoussam"	1	P19
"senegal - fatick"	1	P19
"rwanda - ruhango"	1	P19
"paul biya bi mvondo - republic of cameroon"	1	P27
"paul bathélemy biyaa bi mvondo - cameroon"	1	P27
"paul biya bi mvondo - hatugo kamerun"	1	P27
"paul - cameroon"	1	P27

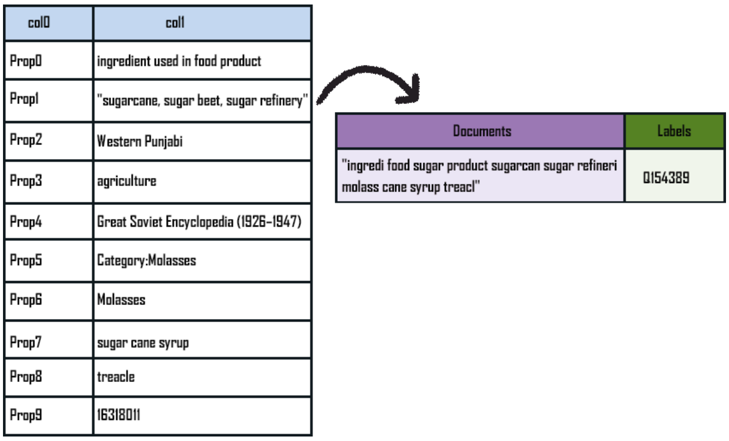
Figure 4: Example of feature extraction for the CPA task during the training phase

5.2.4. Table Topic Detection

Concerning the TTD task, features related to the overall content and structure of the table are extracted. They help determine the primary topic or subject matter of the table. Examples of features for TTD (presented by Fig. 5) may include:

- **Table Headers:** the names of the columns provide valuable information about the table's content. These headers can be extracted as textual features that contribute to the classification of the table topic when they are available.
- **Key Terms:** relevant tokens from the table content can serve as features for the TTD task. These terms are extracted using Term Frequency (TF) algorithms [8].

[8] https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/



col0	col1
Prop0	ingredient used in food product
Prop1	"sugarcane, sugar beet, sugar refinery"
Prop2	Western Punjabi
Prop3	agriculture
Prop4	Great Soviet Encyclopedia (1926-1947)
Prop5	Category:Molasses
Prop6	Molasses
Prop7	sugar cane syrup
Prop8	treacle
Prop9	16318011

Documents	Labels
"ingredi food sugar product sugarcan sugar refineri molass cane syrup treacl"	Q154389

Figure 5: Example of feature extraction for the TTD task during the training phase

5.3. Implementation of Naive Bayes Classifiers

This Section presents how we implemented the Naives Bayes classification for each task using the label from the KG inside the train datasets and the extracted features.

5.3.1. Goal

The objective is to build the learning function $f(X) -> C$

- C is one of the classes/labels in the training dataset(e.g Number, Boolean, Food, Person, Hotel, P18, P651, Q625, Q31, Q54389 etc.).
- $X = (w_1, w_2, ..., w_d)$ is the representation of the data that comes in a bag of tokens/-words.

Given that we have a set of classes/labels $C_1, C_2, ..., C_n$, and we want to determine the probability of an instance X to belong to each class/label. The Bayesian probability of a class/label is obtained using the equation 2.

$$P(label|data) = \frac{P(data|label) * P(label)}{P(data)} \tag{2}$$

- $P(label|data)$: this is the conditional probability to be calculated.
- $P(data|label)$: this is the likelihood of the features of X under the assumption that it belongs to class/label C_i, $P(X|C_i) = P(w_1|C_i) * P(w_2|C_i) * ... * P(w_d|C_i)$
- $P(label)$: represents the initial belief or knowledge about the probability of an instance belonging to class/label C_i before considering any evidence.
- $P(data)$: represents the overall probability of observing instance X, regardless of its class/label.

To classify an instance X, we compute $P(C_i|X)$ for each class C_i and select the class with the highest probability. The implementation is tailored to the specific requirements of each dataset. The following paragraphs give an overview.

5.3.2. WikidataTables Dataset

This dataset consists of tables in CSV format. The CTA, CEA, and CPA targets have to be classified using Wikidata's classes and properties. Three Naive Bayes classifiers were trained for each task, using the labeled dataset on train. The labels, provided by this dataset, represent the ground truth annotations obtained from the Wikidata KG. The entity, semantic type, and relationship predictions for new, unseen tabular data are then predicted using the three trained classifiers.

5.3.3. tFood Dataset

The tFood dataset consists of tables in CSV format. There, the Wikidata class and properties have to be used to classify CTA, CEA, CPA, and TTD targets. Using the same process on WikidataTables dataset, four Naive Bayes classifiers are trained using the labeled dataset of the tFood training data. The first column of a table is ignored as it lacks relevant information for the TTD task (e.g., prop0, prop1, prop2, etc.). Since no corpus is provided for each table, we also adjust the TF-IDF to term frequency. Due to time-consuming training, a limit of 50 cells from each CSV file are randomly extracted per table during training.

5.3.4. SOTAB Dataset

The SOTAB dataset tables are provided in GZ-compressed JSON files. The task was to classify CTA and CPA with schema.org and DBpedia classes. Using the labeled dataset of the SOTAB training data, we trained six classifiers, including two in round 1 and four in round 2. We also limited the number of data elements that could be retrieved from each JSON file to 20 by file at random for each task because many JSON files contained a lot of data, which made the training too time-consuming.

5.3.5. Development environment

The development environment was composed of VSCode as code editor, Node.js as the JavaScript runtime, and npm as the package manager. For feature extraction and data preprocessing, we implement the different utilities from scratch and we consider the Natural[9] npm package for Porter Stemming and building classifiers. We used a desktop with a Ryzen 1700 8 core processor and 16Gb RAM. We also consider multi-instance activity as performance optimization techniques to enhance the efficiency of the implementation.

6. Results

The SemTab 2023 challenge consisted of two (02) rounds lasting from April 14 to June 22, 2023. The pipeline presented by Fig. 1 was applied to each dataset. The following paragraphs present the results provided by the SemTab organizers for the different datasets and a short discussion.

[9] https://naturalnode.github.io/natural

6.1. WikidataTables Dataset

WikidataTables dataset was provided in Round 1 of the challenge. The test data are presented in Table 2 and the challenge's CTA, CEA and CPA results on this dataset are shown in Fig. 6.

Table 2
WikidataTables test set number of tables and targets statistics

Dataset	Tables	CTA	CEA	CPA
WikidataTables	9187	12331	64542	14413

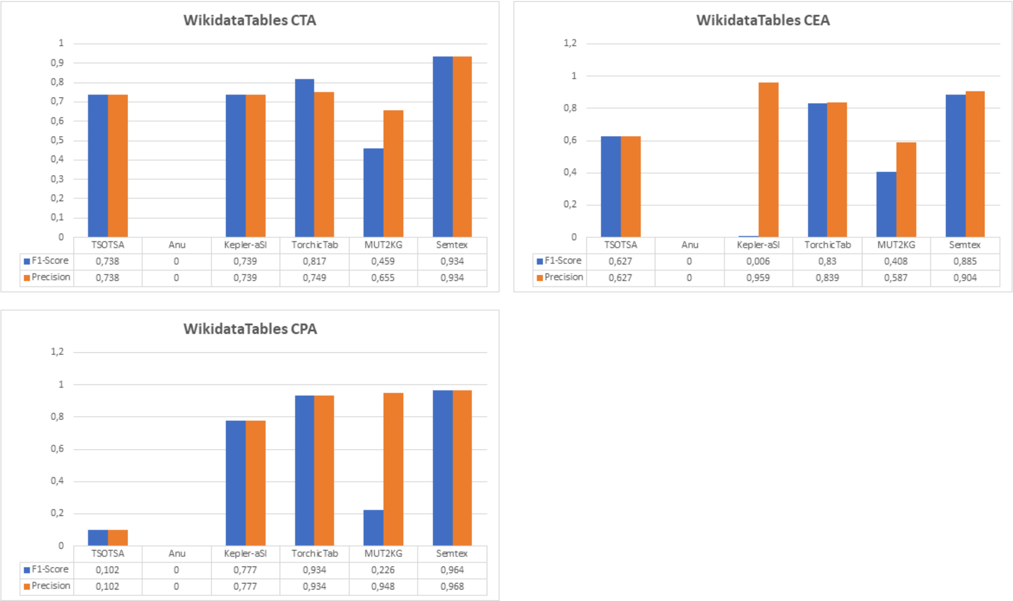


Figure 6: SemTab 2023 results on wikidataTables

The dataset is relatively time-efficient compared to tFood and SOTAB, but the main challenge lies in presenting the best features for classification and avoiding redundant data. Training takes 9 hours, while inference takes around 12 hours.

6.2. tFood Dataset

The tFood dataset was provided in Round 1 of the challenge. This dataset was divided into two main parts: tFood-horizontal and tFood-entity. The test data are presented in Table 3.

Table 3
tFood test set number of tables and targets statistics

Datasets	Tables	CTA	CEA	CPA	TTD
tFood (entity)	7643	-	19777	-	7643
tFood (horizontal)	3945	8200	99613	15863	3945

The tFood datasets revealed challenges in training and inference time, as well as the lack of a corpus to accurately extract the key terms for TTD task. In addition, we discovered that the table's redundancies and unclear data were not appropriate to the approach presented in this paper. Unfortunately, to reduce training time issues, we took a subset in each table with a maximum of 50 rows per table. Typically, 6-8 hours were spent on training and 9-10 hours for inference per task. Fig. 7 presents the results of the challenge on these datasets.

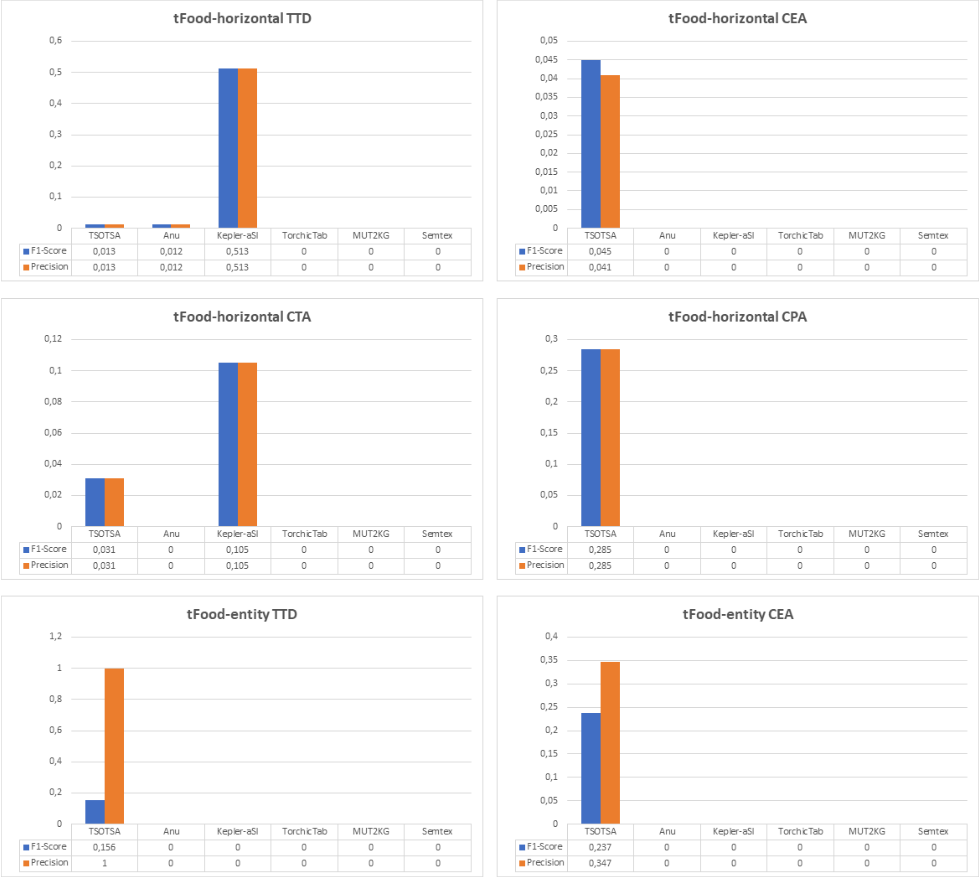


Figure 7: SemTab 2023 results on tFood

6.3. SOTAB Dataset

The SOTAB dataset was provided in Rounds 1 and 2. An overview of the test data is presented in Table 4.

Due to the time required for training and inference, and the limited performance of our training environment, we just submitted to round 2 with this dataset.

The SOTAB dataset has a larger number of tables and rows in JSON format compared to other datasets. To reduce training and inference time, we significantly decreased the number of training tables , and the rows were reduced to 20 per table. We also point out that the

Table 4
SOTAB test set number of tables and targets statistics

Datasets	Tables	CTA	CPA
SOTAB-Round1	402	590	643
SOTAB-Round2 (SCH)	667	1112	1623
SOTAB-Round2 (DBP)	591	681	904

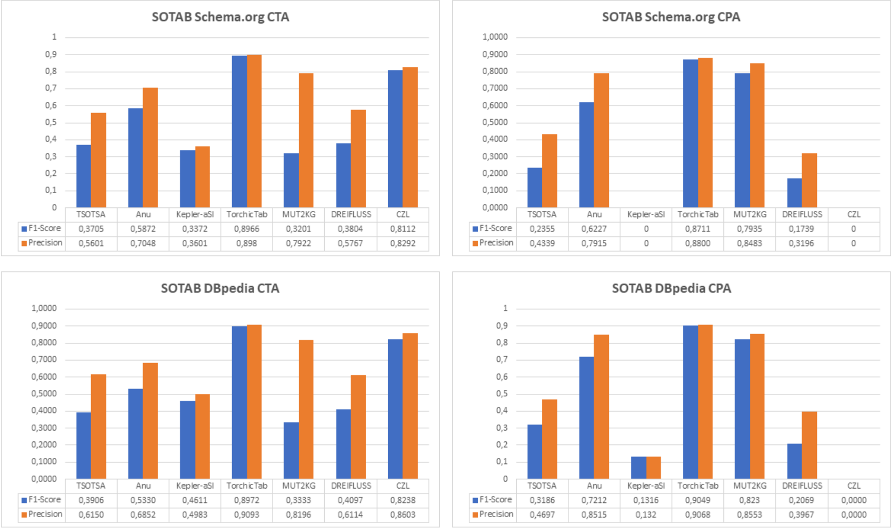


Figure 8: SemTab 2023 Round2 results on SOTAB

redundancy and incorrect data in the tables were not helpful for the proposed approach. Each task took 10-12 hours for training and 8-9 hours for inference.

7. Conclusion

This paper outlines the method we propose for annotating tabular data with knowledge graph classes, entities, and properties using Naives Bayes multiclass classifiers. Data pre-processing aims to ensure the accuracy and integrity of the tabular data as well as to make the subsequent matching process easier. Feature extraction techniques such as the number of co-occurrences and the frequency of terms provide useful information to capture the semantic relationships between tabular data and knowledge graphs. Due to redundant and misleading data in the training dataset, the computation time, the approach was severely limited but we are exploring further solutions to improve our feature extraction and computation times.

Acknowledgment

We are grateful to SemTab organizers for having given us the opportunity to share this work with the community.

References

[1] A. Oelen, M. Stocker, S. Auer, Creating a scholarly knowledge graph from survey article tables, in: Digital Libraries at Times of Massive Societal Transition, Springer International Publishing, 2020, pp. 373–389. URL: https://doi.org/10.1007%2F978-3-030-64452-9_35. doi:10.1007/978-3-030-64452-9_35.

[2] A. Jiomekong, C. Etoga, B. Foko, V. Tsague, M. Folefac, S. Kana, M. M. Sow, G. Camara, A large scale corpus of food composition tables, 2022, pp. 34–36. URL: https://ceur-ws.org/Vol-3320/paper4.pdf.

[3] S. Ting, W. Ip, A. Tsang, Is naïve bayes a good classifier for document classification?, International Journal of Software Engineering and its Applications 5 (2011).

[4] R. Shigapov, P. Zumstein, J. Kamlah, L. Oberländer, J. Mechnich, I. Schumm, bbw: Matching csv to wikidata via meta-lookup, in: SemTab@ISWC, 2020. URL: https://api.semanticscholar.org/CorpusID:229242235.

[5] P. Nguyen, I. Yamada, N. Kertkeidkachorn, R. Ichise, H. Takeda, Semtab 2021: Tabular data annotation with mtab tool, in: SemTab@ISWC, 2021. URL: https://api.semanticscholar.org/CorpusID:247363605.

[6] V.-P. Huynh, J. Liu, Y. Chabot, F. Deuzé, T. Labbé, P. Monnin, R. Troncy, Dagobah: Table and graph contexts for efficient semantic annotation of tabular data, in: SemTab@ISWC, 2021. URL: https://api.semanticscholar.org/CorpusID:247363666.

[7] A. Jiomekong, B. Foko, Towards an approach based on knowledge graph refinement for tabular data to knowledge graph matching, 2022, pp. 111–122. URL: https://ceur-ws.org/Vol-3320/paper12.pdf.

[8] B. Alsafy, Z. Mosad, W. Mutlag, Multiclass classification methods: A review, 2020.

[9] A. Jiomekong, H. Tapamo, G. Camara, Combining Scrum and Model Driven Architecture for the development of the EPICAM platform, in: CARI 2022, Yaounde, Cameroon, 2022. URL: https://hal.archives-ouvertes.fr/hal-03712484.

[10] J. Azanzi, H. Tapamo, G. Camara, Combining Scrum and Model Driven Architecture for the development of an epidemiological surveillance software, Revue Africaine de Recherche en Informatique et Mathématiques Appliquées Volume 39 - 2023 (2023). URL: https://arima.episciences.org/11537. doi:10.46298/arima.9873.

[11] P. Ralph, et al., Empirical standards for software engineering research, 2021. URL: https://arxiv.org/abs/2010.03525. arXiv:2010.03525.

[12] N. Abdelmageed, E. Jimènez-Ruiz, O. Hassanzadeh, B. König-Ries, tFood: Semantic Table Annotations Benchmark for Food Domain, 2023. URL: https://doi.org/10.5281/zenodo.7828163. doi:10.5281/zenodo.7828163.