2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM) | 978-1-6654-2678-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/IMCOM53663.2022.9721780

Significance of Syntactic Type Identification in Embedding Vector based Schema Matching

Fahad Ahmed Satti*†, Musarrat Hussain*, Sungyoung Lee*, and TaeChoong Chung*
*Department of Computer Science and Engineering, Kyung Hee University,
Yongin-si, South Korea
Email: {fahad.satti,musarrat.hussain,sylee}@oslab.khu.ac.kr,
{tcchung}@khu.ac.kr
†National University of Sciences and Technology, School of Electrical Engineering and Computer Science,
Islamabad, Pakistan
Email: {fahad.satti}@seecs.nust.edu.pk
Corresponding Authors: Sungyoung Lee*, TaeChoong Chung*

***Abstract*—Data Interoperability provides a bridge between information systems to store, exchange and consume heterogeneous data. In order to achieve this goal, schema maps provide the necessary foundations. Traditional solutions rely on expert generated rules, ontologies, and syntactic matching to identify the similarity between attributes in the various data schema. While previously we have presented the effectiveness of transformer based models and unsupervised learning to calculate attribute similarities, in this paper we present the additional application of a naive syntactic similarity measurement. We have evaluated the results of agreement between the computed and human annotated results, in terms of Mathews Correlation Coefficient (MCC). These results indicate that on weighted comparison there is no positive effect of including naive syntactic similarity in addition to semantic similarity.**

***Index Terms*—Schema Matching, Health Information Management, Text Processing, Transfer Learning**

I. Introduction

Data Interoperability is a key requirement for achieving ubiquity in information systems. The ability to store, exchange, and consume data among heterogeneous systems or services (IEEE 610.12 [1]) allows the stakeholders to gain a plethora of benefits, which otherwise would be operationally limited to individual applications and providers. Previous methodologies [2], [3] to provide schema mapping are based on expert generated rules, ontology engineering, and/or pattern matching. However state-of-the-art solutions in the field of Natural Language Processing (NLP) convert words and sentences into embedding vectors, which provide better automated solutions to handle unseen data. Word embedding techniques such as Word2Vec suffer from various problems, including out of vocabulary words, which are most conspicuous in the case of attribute names for the data schema, due to their non-standardized naming conventions. Attribute names such as "PatientGender", "DOB", and "systolicbp" are all valid for data schema since they are queried with hardcoded checks and expert generated rules. In order to create the schema maps automatically, the first requirement is to identify the similarity between any pair of attributes. In our previous work [4], we have created schema maps for various Healthcare Information Management Systems (HMIS) by utilizing transformer based transfer learning techniques to convert semantically enriched attribute suffixes into embedding vectors. These vectors were then compared using cosine similarity to identify similar attributes, and evaluated against a human-annotated dataset to verify the accuracy of the method. In some of the previous studies, [5], a combination of similarity between data types of the attribute instances and semantic reconciliation through domain ontologies, has been introduced as a feasible solution. Additional application of syntactic reconciliation can increase the distance measures and by extension the amount of information that is useful for determining the similarity of attributes. The premise being; for an attribute pair, a weighted comparison based on both semantic and syntactic distance measures, can capture a larger number of scenarios and provide better performance in this task. In the current study, we aim to extend our previous semantic only solution, based on pre-trained Bidirectional Encoder Representations from Transformers (BERT) Natural Language Inference (NLI) models [6] to include the syntactic elements in determining the similarities between five schemas. We will evaluate the results of our current work using different weights for syntactic and semantic distance measures and evaluate the performance of our computed methods with human-annotated data, using Mathews Correlation Coefficient (MCC) score. MCC provides a reliable measure of inter-rater agreement on our negatively imbalanced dataset, where other metrics such as accuracy, and f-score are not truly representative of the models' performance [7].

Rest of the paper is organized as follows. Section II provides information on previous approaches. Section III introduces the notations used throughout this paper, and is a necessary read before delving into the details of our methodology in section IV. The experimental setup follows in section V, and its results are then presented in section VI. Finally, section VII concludes the paper.

978-1-6654-2678-7/22/$31.00 ©2022 IEEE

II. Related Work

Ubiquitous healthcare service delivery is dependent on the stability and flexibility of its data storage engine and Electronic Health Records(EHR). EHR can provide a rich source of a patient's medical history, however, within medical silos the EHRs, without data interoperability, are not sufficient enough to build a comprehensive and true representation of the patient. Various hospitals, clinics, and other health centers rely on a variety of data storage solutions, which can cause data and information miss-match. In the past, many research and industry-led initiatives have attempted to bridge this gap by generating schema maps, which can enable heterogeneous HMIS to not only exchange healthcare data but also consume it. Massmann et al. [8] utilized a combination of various matching algorithms to provide a semi-autonomous solution to match schema elements in XML, relational, and Web Ontology Language (OWL) formats. Their work was further enhanced by [9] to include related concepts (such as is-a or part-of relationship) using linguistic semantics to determine the correspondence between various schema. Mehdi et al. [10], utilized a regular expression based approach to match data instances, for a corresponding pair of attributes. The authors, first convert the data instances for the source attribute into regular expressions, which are then used to match random values from data instances in the target attribute belonging to a different schema. With advancements in information technology, the research was then shifted towards using machine learning along with semantic information to generate more accurate maps. Koutras et al. [11] proposed the Relational Embedding MAtcher (REMA), which uses the relational embedding technique to convert database instances, attributes, and schema information into embedding vectors. The vectors for a source schema are then compared with target schema vectors to determine the correspondence among their respective attributes. Lomonosov et al. [12] combined lexical schema matching with semantic techniques to boost the performance of machine learning models (Naive Bayesian, logistic regression, and gradient tree) for correctly identifying similarities between attribute pairs. Portisch et al. [13] performed a brief survey to determine the effectiveness of background knowledge in schema matching. The authors also introduced their knowledge graph embedding-based approach, which utilizes the RDF2Vec [14] to convert resource labels into vectors. The similarity between the vectors can then be used to determine the similarity between the schema.

III. Notations

The set of Data Schema (S) represents the participating schema used as an input to our attribute similarity classification process. This set contains partial schema from various open sourced HMIS systems (s_i). A represents the set of attributes within S, while a_{ij}, represents the j^{th} attribute of the i^{th} schema. Each attribute a_{ij} is further enriched with a context, type, and semantics to create an amplified attribute λ_{ij}. The structure of λ_{ij} is shown in Table I.

TABLE I
The contextually, syntactically, and semantically enriched attribute λ_{ij}.

Attribute Context		
Schema Name	Table Name	Attribute Name
Schema Version	Source	Recorded Date
Attribute Type		
Data Type	Possible Value	
Attribute Semantics		
Suffix Array	Concept Array	

The "Attribute Context" section contains metadata elements such as the name of the schema and table that a particular attribute belongs to. "Attribute Type" contains information on the possible values and data type of the attribute. Here the "Possible Value" is collected from the data instances, while the "Data Type" is a calculated metric. In this study the valid values for "Data Type" are restricted to "Long" (64 bit integer), "Double", "Date", "DateTime" (in ISO format), "Object", "List", and "String". It is pertinent to note here that we have greatly simplified the data formats, using regular expressions and Java's typecasting APIs. However, for a better solution to this problem and to increase the data formats alternate techniques such as the deep neural network-based solution on relations provided by Hulsebos et al. [15] or one of the other deep learning-based techniques on data tables as highlighted by Bonfitto et al. [5], should be used. "Attribute Semantics" section contains the suffix array generated from the attribute's name and concept array which contains concepts belonging to each valid suffix. From λ, our methodology collects the syntactic measures in the form of the attribute's format f_{ij} and semantic measures e_{ij}. These two metrics are used to calculate a similarity score for syntactic comparison (c_{syn}) and a semantic one (c_{sem}). The two similarity scores are then converted into an overall similarity score c, by weighted (w_k) summation of c_{syn} and c_{sem}.

IV. Methodology

Using the attribute context, a pair of amplified attributes λ_{ij} are collected, which belong to distinct schema s_i. The comparison process starts with the application of "Context Identifier" in Fig. 1. The contextual information of the attributes a_{11} and a_{12} is present in λ_{11} and λ_{21}, respectively. This is matched to check if the schema s_1 and s_2 are the same or not, based on the name of the schema. If the two schemas are equal, no further processing is performed for this pair. On the other hand, if the schema is not equal, a simple attribute name comparison is used to determine if the attributes are equal or not (such as "ID" and "id" are considered equivalent). In most cases, a simple schema name check should be enough, however, to provide a general solution and to maintain transparency of operations, the complete context of the attribute is maintained in λ_{ij}. Next the "Syntactic Type Identifier", extracts the type of attribute f_{11} and f_{12}. This operation checks if the data

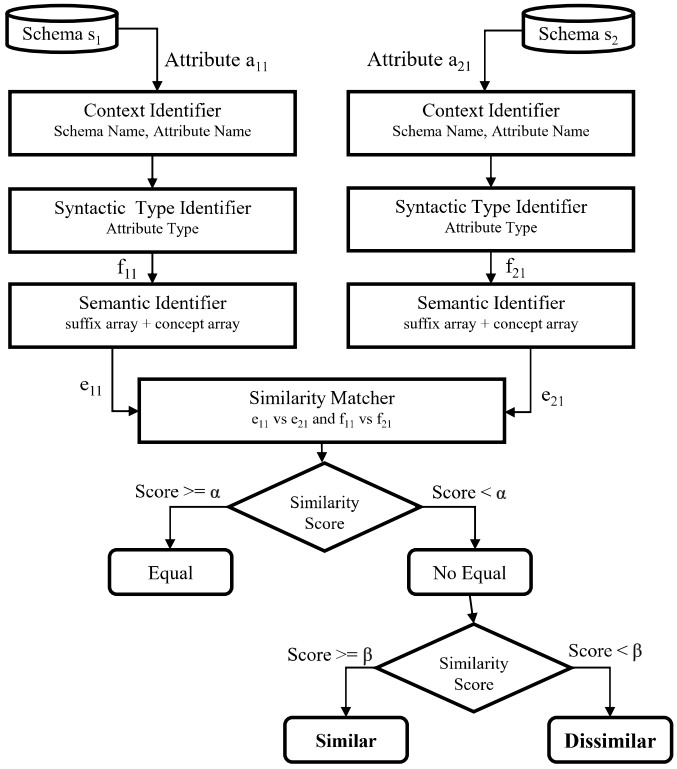

Fig. 1. The workflow for measuring the similarity between attributes.

instances associated with the attribute in "Possible Values", can be typecast into "Long" or not, followed by typecasting into "Double", then ISO "date" and "date-time" objects. If this process fails, we then check if the value can be an object or list, by looking at the encapsulating characters ("{}", and "[]" respectively) for each element of the "Possible Values" field in λ_{ij}. If at this point, we are still not able to determine the type of the attribute, it is set as a string. The determined type is used to update the λ_{ij} and to calculate the syntactic similarity, later on.

"Semantic Identifier" first breaks the attribute name into a suffix array using three methods; forward suffix generation, backward suffix generation, and regular expressions. In forward suffix generation, each attribute name is split into suffixes of length 2 $\rightarrow$ length of attribute name, while in backward suffix generation the attribute name is split in reverse order. In the regular expression-based method the suffix is split on case changes and special characters, such as "-", "_", and others. The three lists of suffixes thus produced are then added to a "TreeSet" which alphabetically sorts them as well.

For each suffix, we then query Unified Medical Language System (UMLS), to identify if the suffix has an associated concept in the healthcare domain. If the suffix is not recognized by UMLS it is discarded from the set. We additionally, collect the concepts associated with each suffix from UMLS and add them to a concept array. This concept array is then converted into a sentence (by joining concept descriptions with the white space character) by the "Semantic Identifier". This sentence is passed to the transformer models, which convert it into an embedding vector.

The "Similarity Matcher" converts the syntactic types f_{11} and f_{21} into syntactic similarity c_{syn} by assigning it a value of 0, if the "Data Type" elements of λ_{11} and λ_{21} are different and 1, if they are either same or convertible. For simplicity, we consider "Long" to be convertible to "Double", "Date" to "DateTime" and vice versa. It also coverts the embedding vectors representing e_{ij} into c_{sem} by calculating the cosine similarity between them.

The two similarity scores, c_{sem} and c_{syn}, are then individually normalized between "0" and "0.5". We then calculate the overall similarity score between the pair of attribute using the method shown in "(1)".

$$c = w_k * c_{syn} + (1 - w_k) * c_{sem} | w_k \in [0.0 - 0.9] \quad (1)$$

The similarity score c is then used to classify its corresponding attribute pair $< a_{11}, a_{21} >$, into "equal", "similar", or "dissimilar". This three-class classification problem is dependent on two thresholds α and β. Here α represents the boundary line, above and equal to which the attribute instance can be classified as equal, while β is the boundary line below α, above and equal to which the attributes are similar. Instances with a similarity score below β are dissimilar. In order to calculate α and β, we evaluate class wise MCC score and then identify the points where MCC_{equal} is maximized, followed by $MCC_{similar}$, and finally $MCC_{dissimilar}$. This process is repeated for w_k values from 0.0 $\rightarrow$ 0.9, which eventually provides the best thresholds for all three thresholds (w_k, α, and β). These thresholds are then used to evaluate the test dataset using the MCC score, which provides an overall agreement between the computed similarities and the human-annotated dataset.

V. Experimental Setup

In this study, the five open source data schema from the domain of healthcare were made a part of the set S. These include EMRBOTS (s_1) [16], OpenEMR (s_2), Pan et. al(s_3) [17], MedTAKMI-CDI (s_4) [18], and our custom implementation (s_5) [19]. For experimental purposes, we only select partial views of s_i, which eventually produce a disjoint set of 254 attributes A. The attributes are cross-compared with each other only if they belong to different s_i. Thus the total number of comparisons between our selected attributes is 20349 (with a total of 64,516 possible comparisons, 23,818 belong to the same schema, and reduction to half to avoid commutative comparisons). For generating sentence embeddings, we have used five BERT based NLI models, trained on the Semantic Textual Similarity benchmark dataset (STSb) [20]. These include, "bert-base-nli-stsb-mean-tokens", "bert-large-nli-stsb-mean-tokens", "roberta-base-nli-stsb-mean-tokens", "roberta-large-nli-stsb-mean-tokens", and "distilbert-base-nli-stsb-mean-tokens". For establishing the baseline truth dataset, the 2d matrix of attribute comparisons was annotated by four human experts. We then calculated

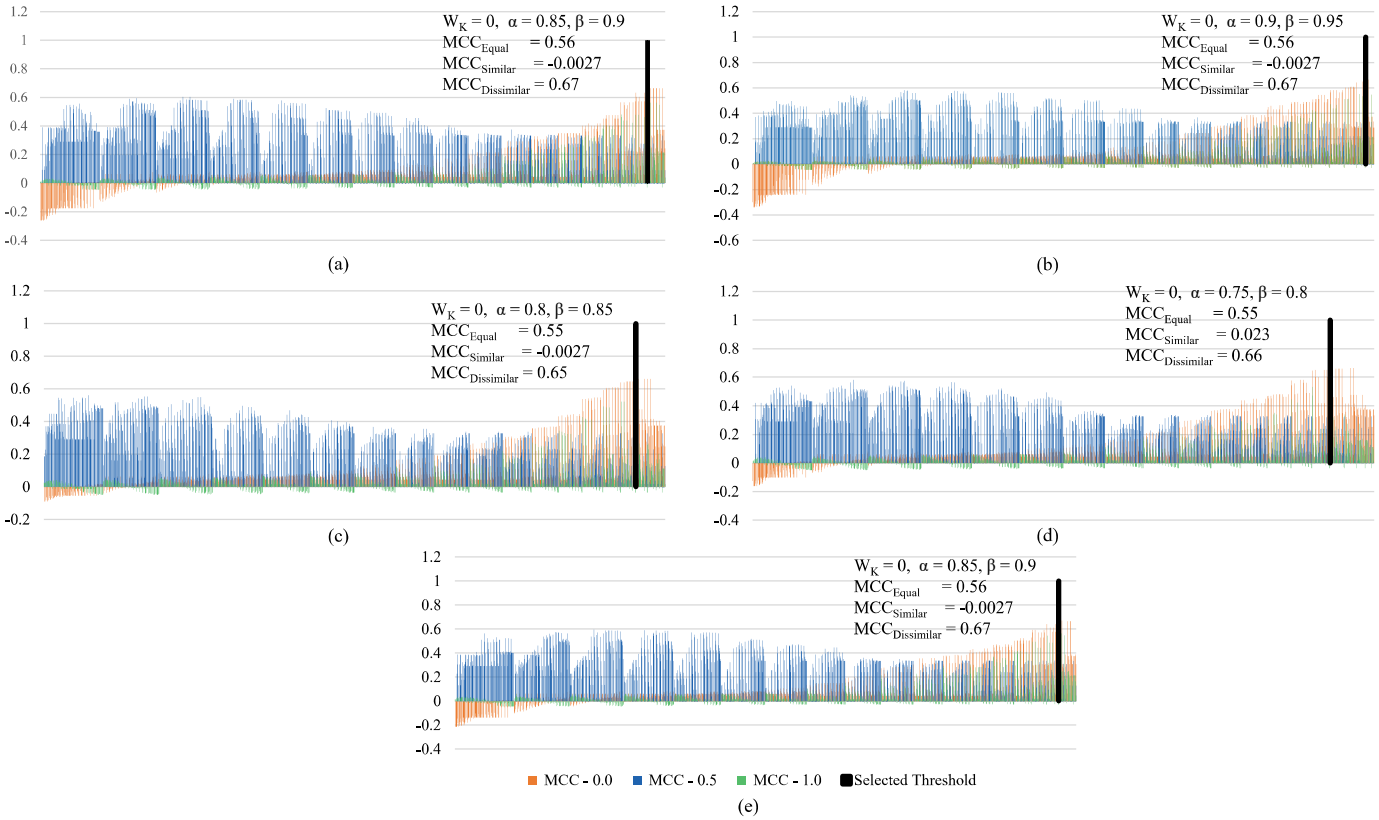

Fig. 2. Threshold selection for the five embedding generation models (a) "bert-base-nli-stsb-mean-tokens", (b) "bert-large-nli-stsb-mean-tokens", (c) "roberta-base-nli-stsb-mean-tokens", (d) "roberta-large-nli-stsb-mean-tokens", and (e) "distilbert-base-nli-stsb-mean-tokens"

the mode of the annotation to create one set of attribute comparisons, with similarity values (equal, similar, dissimilar) based on the most commonly selected labels.

VI. Result

In order to calculate the results, we first split the computed and mode of the annotated dataset into training and test sets with 70%:30% split. We then calculated the optimal thresholds for the five sentence embedding generation models, by calculating the MCC score for the training sets. The MCC scores were evaluated using one vs all class evaluation. The dataset was sorted on the maximum value for MCC_{equal}, followed by maximum value for $MCC_{similar}$, and finally $MCC_{dissimilar}$. The threshold points where the maximum value was achieved, for each model are shown in Fig. 2. In case of "bert-base-nli-stsb-mean-tokens" and "distilbert-base-nli-stsb-mean-tokens", multiple values of α and β had the same MCC scores, where we selected the lowest thresholds. The graphs are plotted based on increasing values of w_k, α, and β. In Fig. 2 (a), the optimal threshold for "bert-base-nli-stsb-mean-tokens", is at α greater than or equal to 0.85, while β greater than or equal to 0.9. Here the MCC_{equal} and $MCC_{dissimilar}$ scores are above 0.5, indicating good agreement between the results of this model and the annotated data. However, the $MCC_{similar}$ value of -0.0027 indicates random classification. Due to the biased nature of our dataset, with a very large number of dissimilar elements and a small number of equal and then similar elements, these numbers are acceptable. For Fig. 2 (b), the values of α and β both move ahead to 0.9 and 0.95, however, the MCC scores achieved by this model against the annotated dataset remain the same.

The results obtained from the application of "roberta-base-nli-stsb-mean-tokens" and "roberta-large-nli-stsb-mean-tokens" embedding models are shown in Fig. 2 (c) and Fig. 2 (d), respectively. The α and β threshold values for these model are also closely related, as in the previous case. Finally, Fig. 2 (e) shows the results for the "distilbert-base-nli-stsb-mean-tokens" embedding model. As shown by these graphs, the MCC scores for class equal, similar, and dissimilar are all very close to each other. While the $MCC_{similar}$ score for "roberta-large-nli-stsb-mean-tokens" is higher than the other four models, it is still very small, relative to the other classifications.

Fig. 3 shows the MCC result of various w_k values, when the thresholds α and β are kept constant. These results show the ineffectiveness of incorporating syntactic similarity measures in determining the similarity between the attribute pair. The best MCC scores are achieved when w_k is at 0.0. This value of w_k causes the syntactic similarity measure to become 0.0 while the corresponding semantic similarity is maximized. Thus, at this stage, the results show that only the BERT based NLI models provide significant results in terms of identifying the

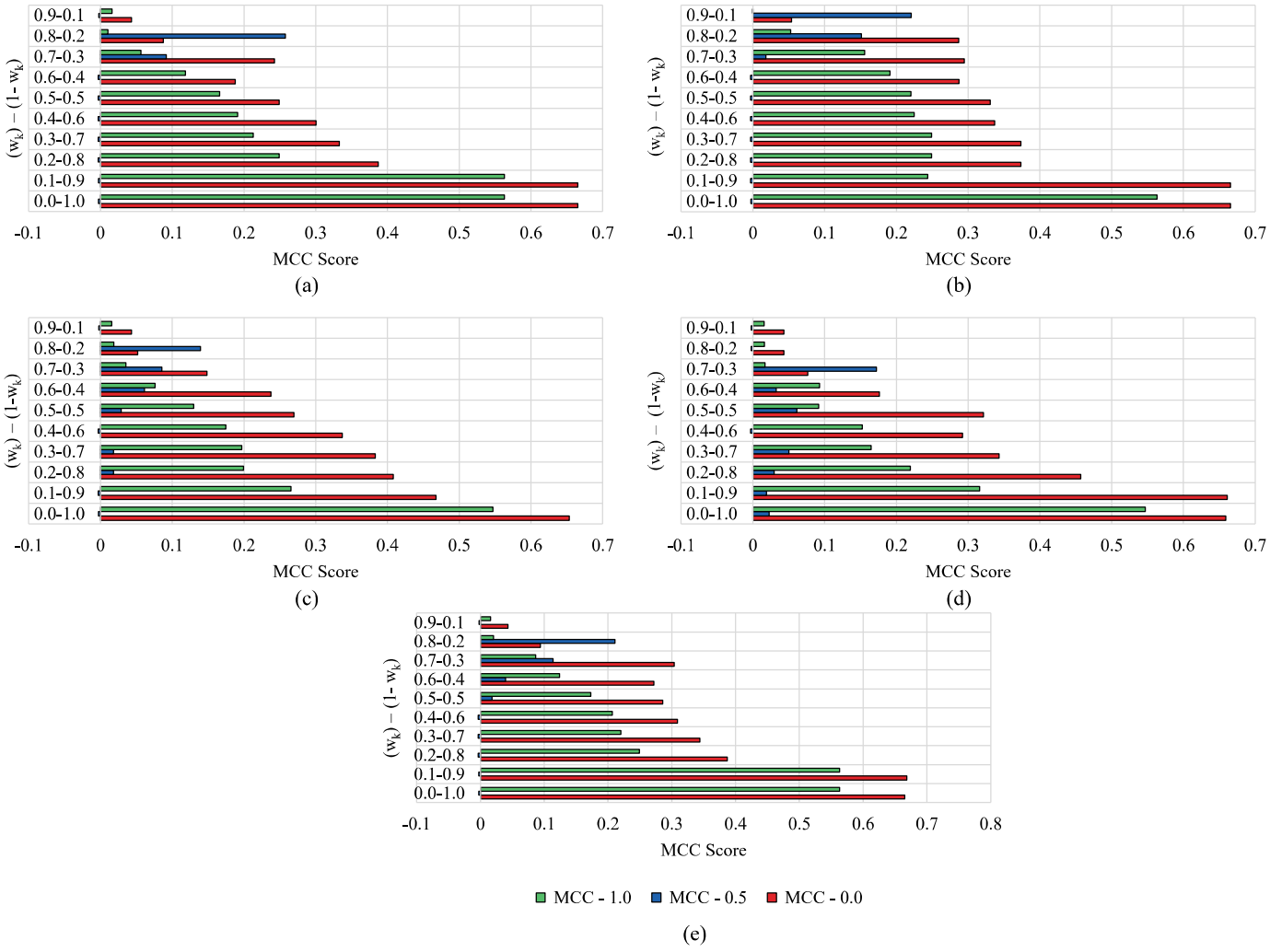

Fig. 3. Weight Selection (w_k), at best thresholds (α and β) for the five embedding generation models (a) "bert-base-nli-stsb-mean-tokens", (b) "bert-large-nli-stsb-mean-tokens", (c) "roberta-base-nli-stsb-mean-tokens", (d) "roberta-large-nli-stsb-mean-tokens", and (e) "distilbert-base-nli-stsb-mean-tokens"

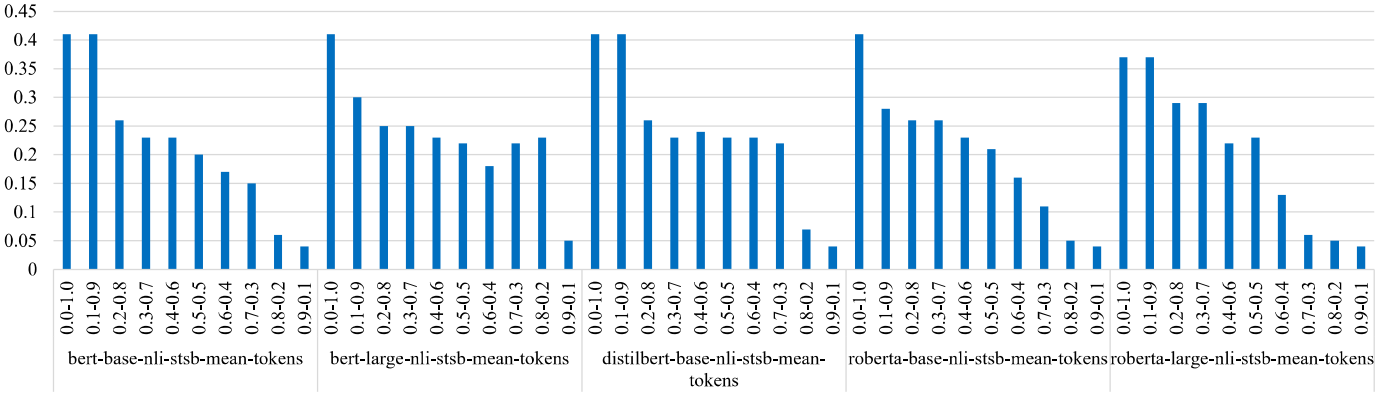

Fig. 4. MCC inter-rater agreement between test instances of the five computed models and the mode of annotated dataset

similarity between the attributes.

On the test dataset the MCC score, achieved at various syntactic and semantic thresholds (w_k and 1-w_k), for the five models is shown in Fig. 4. Similar to the results observed in the threshold selection phase, the best MCC values are achieved by the models when the weight of the semantic similarity is maximized. In general, as we move the weights, in favor of syntactic similarity the inter-rater agreement between

the five computed models and mode of the annotated dataset, decreases, eventually dropping below 0.1. When compared to our previous results, as presented in [4], all models reflect acceptable performance, with a normalized MCC score around 0.4. In absolute terms, this score indicates a weak level of agreement [21] between computed and mode of annotated data.

VII. Conclusion

Traditional approaches to resolving heterogeneity in data schemas rely on a combination of syntactic and semantic similarities, driven by expert-generated rules, which lack flexibility. Previously, we have proved the applicability of state-of-the-art transformer-based BERT NLI models in determining the similarity between attribute pairs and by extension generating the schema map. With empirical results, we now know that BERT models, trained on the STSb dataset are enough in accomplishing this similarity classification task. The addition of naive syntactic similarity measures only reduces the effectiveness of the linguistic semantic models. However, as the syntactic similarity measurements grow stronger, they might be able to produce an impact and in the future, we will look towards evaluating the effectiveness of these newer approaches on our proposed schema map generation process.

Acknowledgment

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-0-01629) supervised by the IITP(Institute for Information & communications Technology Promotion), by the Korea government MSIT (Ministry of Science and ICT) grant (No.2017-0-00655), by the MSIT Korea, under the Grand Information Technology Research Center support program(IITP-2020-0-01489),(IITP-2021-0-00979)supervised by the IITP (Institute for Information & communications Technology Promotion) and NRF2019R1A2C2090504.

References

[1] Anne Geraci, Freny Katki, Louise McMonegal, Bennett Meyer, John Lane, Paul Wilson, Jane Radatz, Mary Yee, Hugh Porteous, and Fredrick Springsteel. *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*. IEEE Press, 1991.

[2] Ali A Alwan, Azlin Nordin, Mogahed Alzeber, and Abedallah Zaid Abualkishik. A survey of schema matching research using database schemas and instances. *International Journal of Advanced Computer Science and Applications*, 8(10), 2017.

[3] Martijn G Kersloot, Florentien JP van Putten, Ameen Abu-Hanna, Ronald Cornet, and Derk L Arts. Natural language processing algorithms for mapping clinical text fragments onto ontology concepts: a systematic review and recommendations for future studies. *Journal of biomedical semantics*, 11(1):1–21, 2020.

[4] Satti, F., Hussain, M., Hussain, J., Ali, S., Ali, T., Bilal, H., Chung, T. & Lee, S. Unsupervised Semantic Mapping for Healthcare Data Storage Schema. *IEEE Access*. pp. 1-12 (2021)

[5] Bonfitto, S., Casiraghi, E. & Mesiti, M. Table understanding approaches for extracting knowledge from heterogeneous tables. *Wiley Interdisciplinary Reviews: Data Mining And Knowledge Discovery*. pp. e1407 (2021)

[6] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[7] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.

[8] Massmann, S., Raunich, S., Aumüller, D., Arnold, P., Rahm, E. & Others Evolution of the COMA match system. *Ontology Matching*. **49** pp. 49-60 (2011)

[9] Arnold, P. & Rahm, E. Enriching ontology mappings with semantic relations. *Data & Knowledge Engineering*. **93** pp. 1-18 (2014)

[10] Mehdi, O., Ibrahim, H. & Affendey, L. Instance based matching using regular expression. *Procedia Computer Science*. **10** pp. 688-695 (2012)

[11] Koutras, C., Fragkoulis, M., Katsifodimos, A. & Lofi, C. REMA: Graph Embeddings-based Relational Schema Matching.. *EDBT/ICDT Workshops*. (2020)

[12] Bulygin, L. Combining lexical and semantic similarity measures with machine learning approach for ontology and schema matching problem. *Proceedings Of The XX International Conference "Data Analytics And Management In Data Intensive Domains"(DAMDID/RCDL'2018)*. pp. 245-249 (2018)

[13] Portisch, J., Hladik, M. & Paulheim, H. Background Knowledge in Schema Matching: Strategy vs. Data. *ArXiv Preprint ArXiv:2107.00001*. (2021)

[14] Ristoski, P., Rosati, J., Di Noia, T., De Leone, R. & Paulheim, H. RDF2Vec: RDF graph embeddings and their applications. *Semantic Web*. **10**, 721-752 (2019)

[15] Hulsebos, M., Hu, K., Bakker, M., Zgraggen, E., Satyanarayan, A., Kraska, T., Demiralp, Ç. & Hidalgo, C. Sherlock: A deep learning approach to semantic data type detection. *Proceedings Of The 25th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining*. pp. 1500-1508 (2019)

[16] Uri Kartoun. A methodology to generate virtual patient repositories. *arXiv preprint arXiv:1608.00570*, 2016.

[17] Lijun Pan, Xiaoting Fu, Fangfang Cai, Yu Meng, and Changjiang Zhang. Design a novel electronic medical record system for regional clinics and health centers in china. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 38–41. IEEE, 2016.

[18] Akihiro Inokuchi, Koichi Takeda, Noriko Inaoka, and Fumihiko Wakao. Medtakmi-cdi: interactive knowledge discovery for clinical decision intelligence. *IBM Systems Journal*, 46(1):115–133, 2007.

[19] Taqdir Ali and Sungyoung Lee. Reconciliation of snomed ct and domain clinical model for interoperable medical knowledge creation. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2654–2657. IEEE, 2017.

[20] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I. & Specia, L. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *ArXiv Preprint ArXiv:1708.00055*. (2017)

[21] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.