Integrating Datasets with Discrete and Natural Language Annotations for Person Retrieval

Harsh Tripathi
BITS Pilani, K K Birla Goa Campus
Goa, India
htripathi6@gmail.com

Jay N. Chaudhari
Ahmedabad University
Ahmedabad, India
jay.chaudhari@ahduni.edu.in

Hiren Galiyawala
RyDOT Infotech Pvt Ltd
Ahmedabad, India
hireng@rydotinfotech.com

Paawan Sharma
Pandit Deendayal Energy University
Gandhinagar, India
paawan.sharma@sot.pdpu.ac.in

Mehul S Raval*
Ahmedabad University
Ahmedabad, India
mehul.raval@ahduni.edu.in, *Corresponding author

***Abstract*—Person retrieval video using natural language description (NLD) is an emerging research area and depends largely on dataset diversity. Unifying datasets increases overall quality; therefore, the paper presents a case study on merging two different style data sets; one has NLD with images (CUHK-PEDES), and the other has discrete annotations with videos (AVSS). The unifying framework brings out the practical challenges and their solution. Explicit discussions on data set merging frameworks are missing in the literature, and our work will facilitate the researchers' requirements.**

***Index Terms*—Attribute, dataset merging, person retrieval, soft-biometrics, surveillance, videos**

I. Introduction

Given the textual query containing attributes, the person retrieval aims to spot the person of interest in an image gallery or video [1]–[3]. It is a complex problem requiring computer vision (CV) and natural language (NL) and depends on the data set's quality. The usual practice is to merge data sets and then train a machine learning (ML) model. The combined data sets for person retrieval have several advantages, such as increased sample diversity, reduced bias, and better generalisation. Several attempts at data set merging have been made; e.g., Specker et al. [4] proposed UPAR: Unified Pedestrian Attribute Recognition and Person Retrieval data set by merging PA100K [5], Market1501 [6], PETA [7], and RAPV2 [8]. Further, PETA data set [7] is a combination of 10 different data sets. The researchers have shown that merging data sets improves person attribute recognition (PAR) performance, as shown in Table I and Table II. PAR using NLDs aims to recognise individuals without establishing their identity or infringing on privacy.

TABLE I: Results of PAR due to merging of data sets [4]. mAP-S: Mean Average Precision when trained on a single data set, mAP-M: mAP when trained on multiple data sets, F1-S: F1 score when trained on a single data set, F1-M: F1 score when trained on multiple data sets.

mAP-S	mAP-M	F1-S	F1-M
67.0±2.5	72.6±2.4	74.2±4.5	81.4±2.3

TABLE II: State-of-the-art person retrieval result due to merger of RAP [8] and AVSS data set [9]. IOU: Intersection Over Union, TPR: True Positive Rate.

Methods	Average IOU	IOU $\geq$ 0.4	TPR(%)
[10]	0.667	0.856	85.30

The literature review suggests that PAR data set mergers are done manually, and no automated framework is specified. Also, manually creating a large-scale dataset requires resources and money. Moreover, the datasets have images or videos with different spatial resolutions. They are imbalanced, contain pose and viewpoint variations, and capture conditions vary, as shown in Fig. 1. The annotations with the datasets are either Natural Language Description (NLD) or discrete annotations (DA); the number of attributes and their values varies; the structure of NLD varies due to variations in language use.

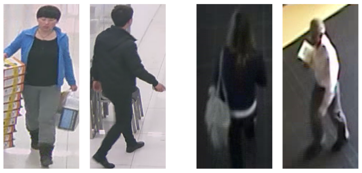

Fig. 1: Samples of RAP [8] and AVSS [9] data set showing challenges due to viewpoint, background, and capturing environment.

Each annotation style has certain advantages - DA allows standardisation as it has fixed values, reducing ambiguity in the attribute values. Operationally, they are also easier to manage using fewer resources. On the other hand, NLD describes the context, offers variations in the language, which is useful for training, and depicts how humans communicate information.

Merging data sets with DA and NLD will provide an opportunity to reduce ambiguity using contextual information.

The paper proposes a dataset merging framework using two data sets with contrasting styles of annotations - DA and NLD, and different modalities - images and videos. It combines CUHK-PEDES [11] (NLD style source data set I (S-I)) with AVSS [9] (DA style source data set II (S-II)), and the merger is mapped to target data set (TD) which is AVSS [9] again. Also, we note that the NLDs are converted to DA during the merging process. This is a first-of-its-kind framework for merging, as existing methods did not consolidate data into a specific target format but instead concatenated datasets on top of each other through crowdsourcing or manual methods.

II. Preliminaries: Data Set Merging

We resort to *hybrid merging* as it involves - merging new cases and adding new variables as the target data set will include both new examples and features. The two main ways of hybrid merging are as follows. **Merging new cases:** It is also known as adding or removing data by rows and assumes that the variables from both sources are compatible - new instances are independent, identically distributed, and derived from the same population [12]. **Adding new variables:** It introduces a new attribute with the same observations from both source data sets [12]. They assume that the new attribute adds meaningful information, is compatible with observation, and does not correlate strongly with the existing observations.

Row merging increases the size of samples, whereas column merging increases the span of the soft biometric data set.In the case of observations where the new attribute is similar to an existing attribute, the relationship (keys) between the latter and the corresponding value is maintained. However, the overlapping observations are removed to reduce the dimensionality.

III. The Proposed Framework

The proposed framework extracts adjectives and nouns from NLD and uses dictionaries for matching the source and target data set formats. The dictionaries provide flexibility and scalability, and the mapping is learned through a matrix, which provides robustness. Using functions, dictionaries are searched to find adjectives and encoded as per the TD. The framework is discussed in the following sections.

A. Preprocessing to extracting adjective-noun pairs

The steps followed in the extraction of pairs are as follows:

1) **The SpaCy matcher [13]:** It parses all the sentences in the CUHK-PEDES data set to extract the ADJective (ADJ)-NOUN pairs. The SpaCy matcher uses a rule base to map tokens to entities in the NLD. It provides efficient and flexible matching of the complex patterns occurring in NLD and is fast, and generates clean pairs. The first word of each pair is inserted into an *adjective list*, and the second word is inserted into a *noun list*. These lists are appended each time a new pair is extracted and saved in memory which will be loaded and used later.
2) **Customized ADJ-NOUN extraction method:** SpaCy matcher [13] misses out on certain pairs, for example, the Adjective(ADJ)-and-ADJ-NOUN(e.g., black and white t-shirt), ADJ-ADJ-NOUN(e.g., long white pants). We use a customized method on top of the SpaCy matcher to ensure none of the pairs in NLD are missed. The customized method uses the adjective and noun lists obtained in Step 1. It takes the NLD, splits it into individual words, and parses through each word, checking whether it belongs to the adjective list. If it does, the method extracts the current and following words as an ADJ-NOUN pair.
3) **Final List:** Given the NLD, the proposed framework extracts pairs via two parallel channels: 1. SpaCy matcher stores them in a *matcher list*; 2. the customized method stores them in a *custom list*. The union operation is performed on both to get a 'finalist.' The Fig.2 highlights the above processing. It is necessary to use both the custom and matcher lists to construct the final list so as not to miss the ADJ-NOUN pair.

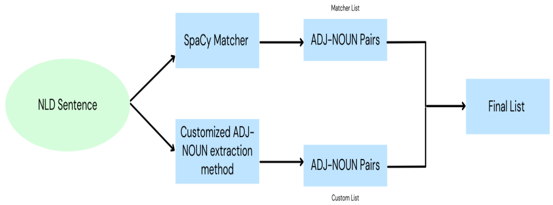

Fig. 2: NLD to finalList processing as per Subsection III-A.

E.g., if the NLD is "A short woman with blue and red top that is wearing brown shoes". Matcher List: Short Woman, Brown Shoes Custom List: Blue and Red Top Final List: Short Woman, Brown Shoes, Blue and Red Top

B. Target attribute dictionary with synonyms and classification lists for noun - D_{S_C}.

The NL queries are imprecise; for example, an observer can describe the gender as - Male, Boy, Man, or Female, Woman, or Girl. The proposed framework solves this variation by creating a list of synonyms. All the TD attributes are stored in the D_{S_C} and form the keys. Their value is a list of the following types:

1) **Synonyms list:** Synonyms words can be interchangeable for the attribute. E.g., the target attribute 'skinColor,' is the key, and the corresponding value could be a list having words 'hue,' 'tone,' and 'complexion.'
2) **Classification list:** Many words can be grouped or classified as per the attribute, e.g., 'footwear' (key) may have values as - 'shoes,' 'sandals,' and 'slippers.'

One must note that the order in which the target attributes are set in the dictionary is crucial for further indexing and search. The structure of D_{S_C} is shown in Fig. 3.

C. Generate a mapping matrix (MM)

Need for a matrix: NLD has variation in its structure and vocabulary, i.e., in one query, the first ADJ-NOUN pair describes the *height*, and in the other, it may describe the

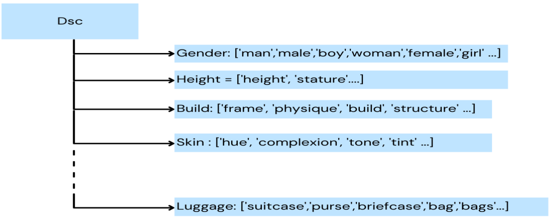
Dsc
Gender: ['man','male','boy','woman','female','girl' ...]
Height = ['height', 'stature'....]
Build: ['frame', 'physique', 'build', 'structure' ...]
Skin : ['hue', 'complexion', 'tone', 'tint' ...]
Luggage: ['suitcase','purse','briefcase','bag','bags'...]

Fig. 3: Target attributes dictionaries with synonyms and classification lists for nouns.

hair. The pair occurs randomly at different locations in the NLD. The proposed framework accommodates this variation by generating the mapping matrix to index the attributes described by the ADJ-NOUN pair.

Matrix structure: With the 'finalList,' matrix filled with zeros is initiated for each NLD. The number of columns equals the number of target attributes, and the rows equal the number of pairs in the 'finalList.' Considering the worst-case scenario, all attributes may occur per NLD; hence the maximum size of this matrix is $n \times n$ where n is the number of attributes. In this study, we considered 13 attributes from the AVSS data set. Therefore, the mapping matrix will be 13×13, with each column representing a specific attribute.

Filling the matrix: The column indices in the matrix are in the *same order* as the attributes organized in the dictionary D_{S_C}. Thus, if the first attribute corresponds to 'height,' in the D_{S_C}, the first column in the matrix will also correspond to the 'height.' The framework parses the 'finalList' and splits each entry into adjectives and nouns. Then it picks the noun, finds it in the D_{S_C}, and locates the attribute to which it belongs. Using the attribute's index, the framework finds the column index in the matrix and places a 1, a flag to note the attribute present in the NLD. The MM structure is shown in Fig 4.

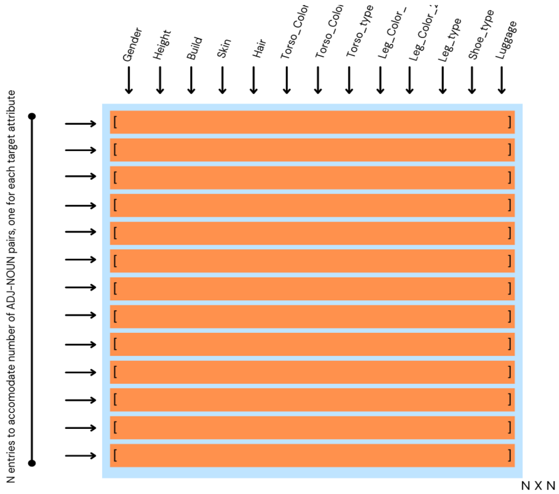

Fig. 4: The mapping matrix structure. N rows for ADJ-NOUN pairs assuming at most one ADJ-NOUN pair per target attribute. N columns for each target attribute.

D. Target attribute dictionary for numeric encoding of adjectives - D_A

The framework uses Python dictionaries(D_A) for each target attribute. These dictionaries are used to find the value of each attribute in MM. The attributes in D_A are arranged in the same order as D_{S_C} and columns of MM. It assigns a number to the adjective describing the attribute. The mapping is as target data set; for example, attribute 'height' has a value 'Very short' and is assigned 0 per the AVSS. Please refer to Table III for AVSS-style numeric encoding. The dictionaries also have a many-to-one mapping to handle synonyms in the NLD. For example, the attribute 'gender' has values like male, man, and boy, all encoded with '0' and female, woman, and girl mapped to'1' as per AVSS format. The Fig. 5 shows the structure of dictionaries with numeric encoding.

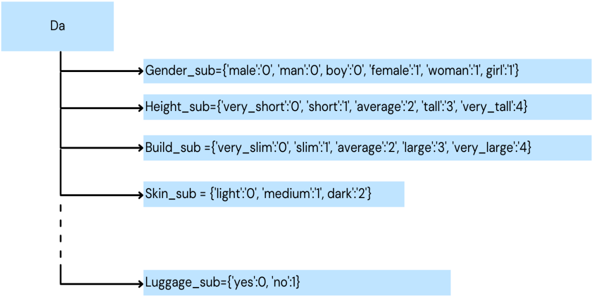
Da
Gender_sub={'male':'0', 'man':'0', boy':'0', 'female':'1', 'woman':'1', girl':'1'}
Height_sub={'very_short':'0', 'short':'1', 'average':'2', 'tall':'3', 'very_tall':4}
Build_sub ={'very_slim':'0', 'slim':'1', 'average':'2', 'large':'3', 'very_large':'4}
Skin_sub = {'light':'0', 'medium':'1', dark':'2'}
Luggage_sub={'yes':0, 'no':1}

Fig. 5: Target attribute dictionaries for numeric encoding of adjectives D_A.

E. Finding value and its numeric encoding

The framework uses 'FindFeatureValue' functions which take 'finalList' as input and splits each element into nouns and adjectives. Next, it uses the noun and finds the attribute by searching dictionary D_{S_C}. Using this attribute, the framework maps into the D_A where the adjective is located in the list associated with the attribute. The adjective is encoded with a number as per the AVSS format. For example, if the ADJ-NOUN pair is 'Red Shoes,' the framework would use noun *shoes* and map it to 'footwear' attribute in the D_{S_C}. The framework then uses the adjective *Red* and assigns it a numeric value defined in the list associated with *shoes* in the D_A.

F. Finding column index for mapping into data frame

The proposed framework loop through each row of the MM. The framework uses *'FindFeature'* function that takes the 1D vector as input and finds the index at which the value '1' is present. This allows the framework to locate the attribute index in the data frame where its numerically encoded value will be placed.

G. Tabulating data

The framework creates a data frame with rows corresponding sentences, and columns are ordered like the mapping matrix and dictionaries -D_{S_C} and D_A. It processes the 'finalList' for each sentence as discussed in this section, and the process repeats for a new sentence.

IV. Experiments and Discussions

The proposed method uses the CUHK-PEDES [11] testing file containing 6156 NLDs of subjects observed in 3076 images with two structured sentences describing each picture. The methods need pre-processing, especially for ADJ-and-ADJ or ADJ-ADJ type input, e.g., *A black and white shirt*. Such input is converted to the format ADJ_and_ADJ or ADJ_ADJ. The framework has been customized to map the primary and secondary adjectives.

An NLD processing requires assigning the attribute's values to different numbers. For example, *Skin tone* has four values - "Unknown," "Light," "Medium," and "Dark" mapped to numbers -1, 0, 1, and 2. Table III shows the encoding of some attribute values to ás per the AVSS [9] dataset.

Attr.Val	N
Unknown	-1
Light	0
Medium	1
Dark	2

(a) Skin tone.

Attr.Val	N
Unknown	-1
Very Slim	0
Slim	1
Average	2
Large	3
Very Large	4

(b) Build.

Attr.Val	N
Unknown	-1
Very Short	0
Short	1
Average	2
Tall	3
Very Tall	4

(c) Height.

TABLE III: Adjective dictionary D_A for attributes skin, build, and height. Different values are assigned numbers as per AVSS [9]. Attr.val: Value of the attribute, N: Number assigned to a value.

A. Output of the proposed framework

We showcase the mapping of attributes to their numeric values through the following preprocessed NLDs.

NLD - 1: *A man with a tall height having very_slim build, light skin, dark hair is wearing an orange_and_white T-shirt. It is a short_sleeved T-shirt. He also wears black_and_blue long_shorts and yellow shoes. He carries a blue bag.*

NLD - 2: *A tall stature man with a slim build and medium skin is wearing a green T-shirt. It is a short_sleeved T-shirt, and it is styled with grey long_shorts with black shoes. He also carries a small bag.*

The mapping generated by the proposed approach for NLD - 1 is shown in Table IVa. All attributes are correctly extracted and mapped to their respective labels, e.g., *tall* height to 3, *very slim* build to 0, *light skin* tone to 0. The correctness of the generated output is validated by comparing Table IVa with dictionary D_A defined in the Table III.

The NLD sometimes may not have an attribute or its values. Such a case is shown in NLD - 2 in which *Hair*, secondary upper clothing color *(Torso 2)*, and lower body clothing color*(Leg 2)* attributes are missing. They are mapped to number -1 as they are unknown in the NLD - 2, and the output is shown in Table IVb.

The method also generated erroneous results during the extraction of ADJ-NOUN pairs. A few sample sentences and corresponding erroneous outputs are as follows: **NLD - 3:** *A woman with dark hair and wearing a knee-length striped dress looks downward as she walks and carries a green shopping bag.* **Output ADJ-NOUN - 1:** *dark hair, green shopping* **NLD - 4:** A man is wearing a grey and black T-shirt and denim shorts. He is leaning on a scooter. **Output ADJ-NOUN - 2:** *black tshirt* In NLD -3, we observe that 'green shopping' is extracted instead of 'green shopping bag.' Here, shopping is treated as a noun; thus, the program clubs it with the preceding adjective. In NLD - 4, the framework does not recognize the word *denim* as an adjective and thus fails to extract the succeeding noun with it. The following Subsection IV-C discusses handling such errors and other challenges.

Attr.Val	N
Gender	0
Height	3
Build	0
Skin	0
Hair	2
Torso 1	5
Torso 2	9
TorsoType	1
Leg 1	0
Leg 2	1
LegType	3
Shoes	10
Luggage	0

(a) NLD - 1

Attr.Val	N
Gender	0
Height	3
Build	1
Skin	1
Hair	-1
Torso 1	3
Torso 2	-1
TorsoType	1
Leg 1	4
Leg 2	-1
LegType	3
Shoes	0
Luggage	0

(b) NLD - 2

TABLE IV: Outputs of NLD - 1 and NLD - 2 to values of AVSS [9]. Attr.Val: Values of the attribute, N: Number assigned to the value. Torso 1: primary upper body clothes color, Torso 2: Secondary upper body clothes color, Leg 1: Primary lower body clothes color, Leg 2: Secondary lower body clothes color.

B. Alternatives Methods for ADJ-NOUN Extraction

The ADJ-NOUN extraction can be improved using a large language model (LLM), e.g., ChatGPT API [14] [15]. But as a consequence, the model would need persistent cloud connectivity and external resource availability. It also hampers the goal of an end-to-end pipeline as dependency on other programs increases. Other models, like BERT [16], have been used, but they require more training resources and time. Thus SpaCy was used as the preprocessing tool to understand the merger. The other methods used to extract the ADJ-NOUN pairs were TextRank [17] and processes using NLTK. The methods other than SpaCy were more noisy and were more computationally expensive. SpaCy took under 3 minutes to extract ADJ-NOUN pairs, whereas TextRank extraction [17] took 2 hours to extract the same ADJ-NOUN pairs. Thus, SpaCy proved to be more time efficient in our framework.

C. Practical challenges and their resolution during merger

1) **Handling coordinate structure in English**: The presence of two adjectives in the description causes difficulty when extracting the attributes. For example, *"A black and white shirt"* must extract *the black and white* as an adjective for the shirt. As discussed above, in a preprocessing step, we use ADJ_and_ADJ, and it will allow the framework to identify and solve the challenge.

2) **Non-uniformity in descriptions**: Some words in NLD can be both nouns and verbs. For example, AVSS has an annotation as *'skin'*, which can be used as a verb or noun. Further, it can also be used as an adjective, e.g., *"skin-colored pants or shirt"*. We solve the problem with the help of preprocessing by seeking user intervention. For example, where the *skin color* is to be used as a noun, users are instructed to input it as *skin_color* to remove any ambiguity, and then the framework will process it as a noun.
3) **Increase in the number of attributes**: An increase in the number of target attributes will increase the time complexity by $\mathcal{O}(p)$ where p is the number of new features added. The overall complexity of mapping is of $\mathcal{O}(m \times n)$ where there are *n* elements in the ADJ-NOUN being mapped to *m* target attributes. Efficient search techniques can manage the time complexity due to increased features.
4) **Scalability of the framework**: There are two ways by which this framework is scaled. The first approach is to append adjectives to the adjective array or list when new adjectives are encountered. For example, as discussed in the Subsection IV-A (NLD -4), the adjective *denim* is not initially recognized as an adjective. After adding the word to the adjective array, all future instances of that word will be classified correctly, leading to the proper extraction of ADJ-NOUN pairs. The second approach is appending the synonyms list of all the nouns and attributes. For instance, the attribute "skin" can be referred to in many ways. Someone can use "tone" to describe the person's skin or the word 'hue' to describe the skin. In such examples with more than one word per attribute, appending a new synonym in the predefined dictionary, D_{S_C} resolves this query.
5) **Lack of co-existence of DA and NLD:** The NLD does not have ground truth DAs for CUHK-PEDES, making evaluating performance difficult due to subjectivity.

V. Conclusion

The proposed framework merges complex data sets with different annotation (NLD and DA) styles and data types - image galleries and videos. It can handle language complexities and be modulated to accommodate increased attributes. The framework is independent of the data set like attributive adjectives, non-uniformity in the NLD, and coordinate structure of English. The framework can handle noisy descriptions and generate clean attribute encoding as the target data set's format. Generating a unique mapping matrix per the NLD provides robustness and generalisation to the framework. The framework is scalable using predefined dictionaries D_{S_C} and D_A. By changing entries in the dictionaries, it handles an increase or decrease in the number of attributes, values, or numeric encoding levels. In future, we will reduce user intervention using advancements in large language models and test different dataset unification.

VI. Acknowledgement

We acknowledge the support of the Gujarat Council of Science and Technology (GUJCOST), Department of Science and Technology, Government of Gujarat, for a research grant GUJCOST/STI/2021-22/3858 under STI Policy to carry out this research work.

References

[1] Hiren Galiyawala and Mehul S Raval. Person retrieval in surveillance using textual query: a review. *Multimedia Tools and Applications*, 80(18):27343–27383, 2021.

[2] Hiren Galiyawala, Mehul S Raval, and Shivansh Dave. Visual appearance based person retrieval in unconstrained environment videos. *Image and Vision Computing*, 92:103816, 2019.

[3] Priyansh Shah, Mehul S Raval, Shvetal Pandya, Sanjay Chaudhary, Anand Laddha, and Hiren Galiyawala. Description based person identification: use of clothes color and type. In *National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics*, pages 457–469. Springer, 2017.

[4] Andreas Specker, Mickael Cormier, and Jürgen Beyerer. Upar: Unified pedestrian attribute recognition and person retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 981–990, 2023.

[5] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *Proceedings of the IEEE international conference on computer vision*, pages 1–9, 2017.

[6] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[7] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 789–792, 2014.

[8] Dangwei Li, Zhang Zhang, Xiaotang Chen, and Kaiqi Huang. A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios. *IEEE transactions on image processing*, 28(4):1575–1590, 2018.

[9] Michael Halstead, Simon Denman, Clinton Fookes, YingLi Tian, and Mark S Nixon. Semantic person retrieval in surveillance using soft biometrics: Avss 2018 challenge ii. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[10] Hiren Galiyawala, Mehul S Raval, and Meet Patel. Person retrieval in surveillance videos using attribute recognition. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–13, 2022.

[11] Wanli Ouyang and Xiaogang Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3258–3265. IEEE, 2012.

[12] Jéssica S Santos, Aline Paes, and Flavia Bernardini. Combining labeled datasets for sentiment analysis from different domains based on dataset similarity to predict electors sentiment. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 455–460. IEEE, 2019.

[13] Ines Montani et al. explosion/spaCy: v3.6.0: New span finder component and pipelines for Slovenian, July 2023.

[14] OpenAI. ChatGPT — openai.com. https://openai.com/chatgpt, 2022. [Accessed 04-08-2023].

[15] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Preprint, 2020.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

[17] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.