QUERYING A SIGN LANGUAGE DICTIONARY WITH VIDEOS USING DENSE VECTOR SEARCH

Mathieu De Coster, Joni Dambre

IDLab-AIRO – Ghent University – imec, Belgium

ABSTRACT

To search for an unknown sign in a sign language dictionary, users typically indicate parameters of the query, e.g., hand shape and signing location. Recent advances in sign language recognition enable video-based sign language dictionary search. In such a system, users can record an unknown sign and retrieve a list of signs that look similar, preferably including the queried sign as one of the top results. We have realized such a system by interpreting it as a dense vector search task. First, we learn a mapping (embedding) from sign videos to a vector space. The dictionary can then be searched by looking for the vectors in this space that are closest to the vector corresponding to the query. We present a proof of concept on a subset of the Flemish Sign Language dictionary. Further research is required to scale up our method to the large vocabularies of entire dictionaries.

***Index Terms*—** sign language, vector search, information retrieval

1. INTRODUCTION

Sign language dictionaries provide a bidirectional mapping between signs and corresponding transcriptions or translations into a spoken language. Because sign languages are visual-gestural languages, video datasets form the backbone of these dictionaries. For every sign in a dictionary, there is a video of a person performing it and a set of corresponding annotations, for example hand shape, signing location, and meaning (typically expressed as a word in a spoken language).

Several sign language dictionaries are publicly available, e.g., the Flemish Sign Language (VGT) dictionary [1] and the dictionary of the Sign Language of the Netherlands (NGT) [2]. Sign language dictionaries can be used by sign language learners and experienced signers alike, to learn how to produce certain signs or to look up a sign one does not know.

These datasets can typically be queried in two directions. By entering a spoken language word, one can find all related signs. To query the dictionary in the other direction, users select visual characteristics of the sign. They can, for instance, choose the hand shape and signing location. The result of their query is a list of signs that share these characteristics.

It could be more user friendly to provide a video-based querying system. This would allow users to record themselves using their webcam or upload a prerecorded video and use this video as a query to the dictionary. Thus far, research into this approach has been limited, yet some initial attempts have been made. One approach is to use Dynamic Time Warping (DTW) [3] to match hand trajectories (manually labeled or extracted automatically with OpenPose [4]) [5–7]. Alternatively, deep neural networks can be trained to extract features from videos and classify a query as one of the dictionary entries [8, 9]. Neither of these approaches scales well to large dictionaries. The former only considers hand trajectories that are common to multiple signs and DTW is computationally expensive. The latter requires training a classifier with a large label space and whenever new signs are added to the dictionary, the classifier needs to be updated.

In this paper, we approach this video-based querying task as a dense vector search problem, a technique from information retrieval. It relies on mapping the data onto a high-dimensional vector space (typically using neural networks). Every data point gets assigned a vector in this space. Querying a dataset can then be reduced to mapping the query onto the same vector space and returning vectors closest to the embedded query. This is an efficient operation that can be vectorized and parallellized, and the way the vectors are generated does not have to be updated when new signs are added.

We train a deep neural network for the task of isolated sign recognition (also known as sign classification). Starting from a *different* but related dataset, in our case the VGT corpus [10], we learn a mapping from video data onto a set of signs. This network computes internal representations of the video data as part of its training and inference processes. These are vectors with fewer dimensions than the video data. We use these vectors and their corresponding vector space to model the dictionary data and support the dense vector search algorithm. This model is not trained on the entire set of signs in the dictionary, yet it is still possible to query the dictionary for signs that it has never seen before. This illustrates that new signs can be added to the dictionary without needing to re-train the network that generates the embeddings.

This paper reports on the results from a proof-of-concept demo that was developed for the 2022 HRI Winterschool in Ghent, Belgium[1]. Our contributions can be summarized as follows.

- We approach sign language dictionary search as a dense vector search task.
- We train a sign classification model and map a sign language dictionary onto a latent vector space learned by this model.
- We evaluate the proposed search algorithm on seen and unseen signs.

We provide an overview of related work in Section 2. We describe the used datasets in Section 3. The sign classification model that forms the backbone of the proposed system is described in Section 4.1. Section 4.2 presents the dictionary querying system and its results are discussed in Section 5. We provide a conclusion and directions for future research in Section 6.

Mathieu De Coster's research is funded by the Research Foundation Flanders (FWO Vlaanderen): file number 77410. This work has been conducted within the SignON project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017255.

[1] https://hriwinterschool.com/

2. RELATED WORK

2.1. Searching in sign language video datasets

Video-based querying of sign language dictionaries is still in its infancy. This paper is one of the first to tackle this challenge using deep learning.

Athitsos et al. [7] manually annotate the locations of the hands and face of signers in video data. They then use the DTW algorithm to match the trajectories of the hands (normalized with respect to the location of the face) in query videos with a dataset of 921 signs in American Sign Language (ASL). Before searching, the user of the system must indicate whether the sign input is one-handed or two-handed. This allows the system to eliminate a substantial amount of search results. They obtain a top-20 accuracy of 67%. The limitations of their system are the reliance on the manual selection of handedness and the lack of non-motion information.

Fragkiadakis et al. [5, 6] also use DTW, but they automatically extract the locations of the hands and fingers with the keypoint estimator OpenPose [4]. The handedness of the signer is automatically determined by comparing the velocity of both hands. The poses are flipped horizontally such that all videos are right handed. A top-10 accuracy of 87% is achieved for 100 signs from the Ghanaian Sign Language (GSL) lexicon. Interestingly, the accuracy drops when finger trajectories are included. This may be because these trajectories are more noisy than wrist trajectories. Additionally, more proficient signers obtain more relevant search results. This highlights the importance of correctly executing the query sign.

Both approaches are limited in the sense that they only consider trajectories as a representation of the sign. Movement is only one of the five sign parameters defined by Stokoe [11] and Battison [12]: the remaining four are hand shape, orientation, signing location, and non-manual features.

Wijkhuizen et al. [8] interviewed proficient and novice signers to design a similarity measure for signs. This measure is used to rank search results in an order that is satisfactory to end-users of the system, according to the findings of Hassan et al. [13]. They collect a training set of 300 videos (one for every considered sign) and fine-tune EfficientNetV2-S [14] as a one-shot classifier. 253 different recordings are used as a validation set to evaluate the system. Their system achieves a top-20 accuracy of 60%. Wijkhuizen et al. show that it is essential to not only consider movement, but also location and hand shape.

Sign language dictionary search systems should return multiple results, in case the best match is not the correct result to the user's query. Hassan et al. [15] propose a method to filter the search results, by integrating the feature based search approaches that are currently used to query sign language dictionaries. They find that this increases user satisfaction.

Fink et al. [9] present a video-based dictionary search system in production. Their system is a transformer based deep neural sign classifier that uses keypoints as inputs.

2.2. Sign classification using deep learning

Isolated sign classification aims to classify a video containing a single sign with the label corresponding to that sign. It is a video classification task with applications in, for instance, automatic sign language corpus annotation [16]. It is also used as pre-training for sign language machine translation [17].

We specifically look at keypoint-based sign classification. Rather than training, e.g., a 2D Convolutional Neural Network (CNN) from scratch on the limited sign language data, we can extract keypoint data from sign language videos using a human pose estimator. For every frame of the video, a lower-dimensional representation is extracted. This representation consists of keypoints, i.e., 3D or 2D Cartesian coordinates corresponding to landmarks of the body (elbows, wrists, individual finger joints, etcetera). Popular pose estimators for sign language processing include OpenPose [4] and BlazePose [18] (as part of MediaPipe).

The main idea behind using keypoints is to reduce the problem complexity and improve the generalizability of the downstream models. Ko et al. [19] use OpenPose keypoints as inputs for a sign language recognition system based on recurrent neural networks. De Coster et al. [16] use a similar technique, but show that the remaining inaccuracies in OpenPose keypoints can lead to reduced performance when compared to image based networks [20]. Moryossef et al. [21] compare OpenPose and MediaPipe and also illustrate how failure cases of keypoint estimators can lead to classification errors. Despite this drawback of human pose estimators for sign classification, they remain an interesting choice because of their capability to generalize to new signers. Vazquez et al. [22] show that in some cases transfer learning with models trained with pose data can improve sign classification performance by a large amount.

3. DATASETS

We use two existing datasets, one for training our sign classification model (Section 4.1) and one for evaluating the dictionary search algorithm (Section 4.2). For the classification model, we use a dataset extracted from the VGT corpus, containing 24967 examples for 292 signs. We call this the *pre-training set*. The classification model is trained on signs in context (cut from continuous signing videos). The *evaluation set* corresponds to a subset of the VGT dictionary. These videos are recordings of individual signs with clear beginning and end points. The signs are produced more slowly and more deliberately than the signs in our pre-training set. We consider a random subset of 500 videos (each corresponding to one sign) in the dictionary as the evaluation set.

Table 1. Summary of the signs in the query set.

ID Gloss	Seen?	Number of executions
HEBBEN-A-4801	✓	13
TELEFONEREN-D-11870	✓	9
HAAS-A-16146	✓	9
STRAAT-A-11560	✓	9
PAARD-A-8880	✓	12
BOUWEN-G-1906	✗	11
WAAROM-A-13564	✗	13
MELK-B-7418	✗	11
VALENTIJN-A-16235	✗	9
HERFST-B-4897	✗	9

Finally, we collected a small third dataset containing query videos. These were recorded by twenty hearing inexperienced or first-time signers, who were asked to imitate signs from a set of ten. These signs were selected because they seemed easy to execute for a novice signer. Five signs were present in the pre-training set and five were not. This allows us to distinguish between the performance of seen and unseen signs. See Table 1 for an overview of the chosen

Table 2. Translations of the glosses in Table 1.

ID Gloss	English translation
HEBBEN-A-4801	To have
TELEFONEREN-D-11870	To make a phone call
HAAS-A-16146	Hare
STRAAT-A-11560	Street
PAARD-A-8880	Horse
BOUWEN-G-1906	To build
WAAROM-A-13564	Why
MELK-B-7418	Milk
VALENTIJN-A-16235	Valentine's Day
HERFST-B-4897	Autumn

signs[2]. Table 2 contains the translation in English of the corresponding glosses. In total, this *query set* contains 105 sign executions. As the query videos are recorded by hearing non-signers, we simulate the use case of learning a sign language as a second language. For future work, it would be interesting to compare the difference in retrieval rate between signers and non-signers.

Both the evaluation set and the query set contain signs that did not occur in the pre-training set.

4. METHODOLOGY

4.1. Sign classification

The sign classification model is a modified pose transformer network [20]. The input to the model is a sequence of 3D keypoints extracted with MediaPipe Holistic [23]. We extract 67 keypoints per video frame: 21 per hand and 25 for the upper body. We translate the keypoints such that the origin of the pose is located at the center of the chest. Moreover, we divide all coordinates by the Euclidean distance between the shoulders to reduce the impact of the distance from the camera and personal characteristics. We do the same for each hand, translating to the wrist and dividing by the distance between the wrist and the middle finger knuckle. In some cases, MediaPipe will not predict keypoints [21]. We account for this by linearly interpolating missing keypoints from the nearest non-missing frames or copying from a later or an earlier frame. Keypoints for which this is not possible are replaced by zeros.

These normalized and imputed keypoints form the input to the pose transformer network. This network first learns a dense 128-dimensional *frame embedding* to capture non-linear relationships between the keypoints. A self-attention network performs temporal processing of the frame embeddings within the processed sequence. This results in a 128-dimensional *sequence embedding* vector which is passed to the softmax classifier. We refer to this vector as `[CLS]`.

4.2. Dictionary querying

Consider a dictionary with a vocabulary of N signs (in our case, $N = 500$). For each of these signs, the evaluation set contains a single video $\mathbf{d}_i \in \mathcal{D}$, $i \in [1, N]$. The trained embedding network `[CLS]` represents a function m that maps each video $\mathbf{d}_i$ to a vector space $\mathcal{V} \subset \mathbb{R}^{128}$ in which it is assigned a *key* vector $\mathbf{k}_i = m(\mathbf{d}_i)$.

[2]For visual examples of these and other signs discussed in this paper, see `https://users.ugent.be/~mcdcoste/shared/sltat2023/`.

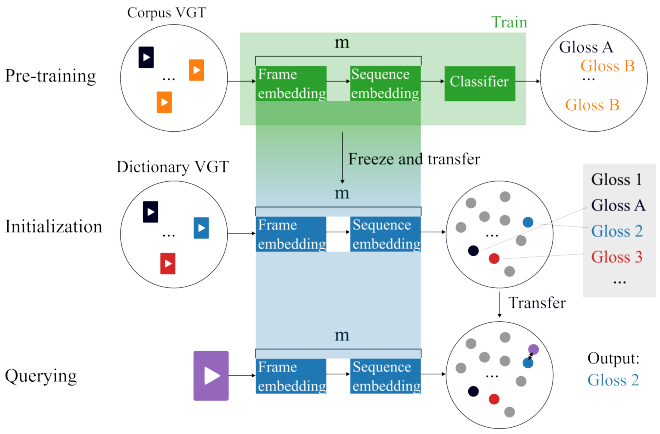

Fig. 1. Dense vector search algorithm for video-based sign language dictionary search based on sign classification pre-training.

Every sign in the dictionary also has a corresponding *value* y_i. In our case, this is a unique identifier (ID Gloss) that links the sign video to its spoken language translation.

When performing a query, the user input video $\mathbf{t}$ is mapped onto the *query* vector $\mathbf{q}$ in the same way: $\mathbf{q} = m(\mathbf{t})$. The search algorithm then compares the query vector $\mathbf{q}$ to the key vectors $\mathbf{k}_i$ using the Euclidean distance in embedding space.

In a nearest neighbour search, the search result y_a is the value corresponding to the key that minimizes this distance,

$$a = \underset{i \in [1,N]}{\arg\min} ||\mathbf{q} - \mathbf{k}_i||_2. \quad (1)$$

Fig. 1 visually illustrates this dense vector search algorithm applied to the case of video-based sign language dictionary querying and its relation to the pre-training task described in Section 4.1.

This algorithm is extended to retrieve an arbitrary number of results by returning values for multiple closest keys.

5. RESULTS

5.1. Quantitative analysis

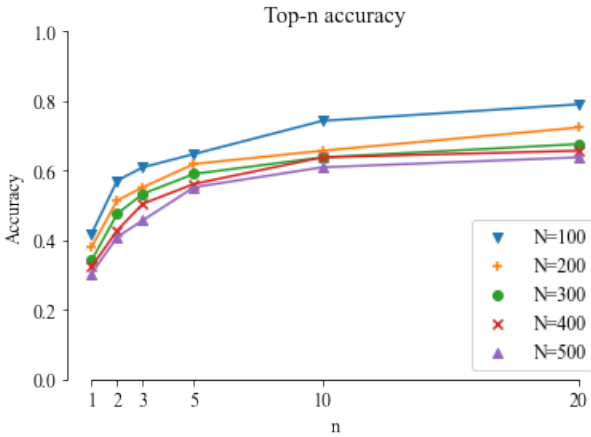

Fig. 2. Top-n accuracy for increasing dictionary sizes.

The correct dictionary entry is the first search result in 30.48% of the cases. When we consider the five first search results, we achieve an accuracy of 55.24%, and the top-20 accuracy is 63.81%. This is illustrated in Fig. 2, which also shows that increasing the number of entries N in the dictionary has a negative impact on the accuracy.

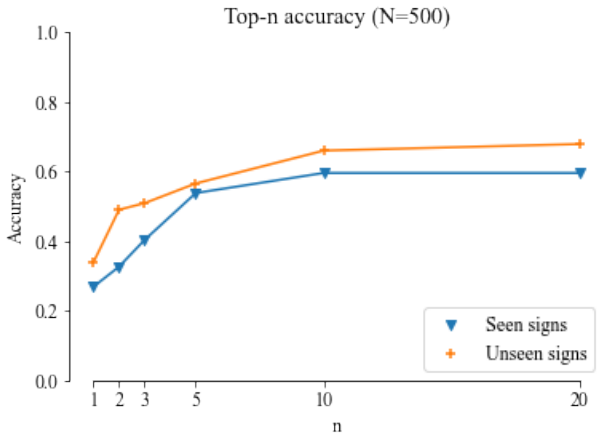

Fig. 3. Top-n accuracy for seen and unseen signs.

We obtain different accuracy values for seen and unseen signs: see Fig. 3. Our model is clearly capable of retrieving unseen signs. Surprisingly, this figure suggests it is better at predicting the unseen signs from the selected set. STRAAT-A-11560, a seen sign, has particularly low accuracy values (0% top-1, 11% top-10, and 22% top-20 accuracy). This sign is performed close to the hips, which are not always in the video frame. Moreover, it consists of movement towards the camera with a lot of self-occlusion in the hands. These are common failure cases for MediaPipe and we observe a lot of missing keypoints. When not considering this sign, the accuracy values for seen (32.56%) and unseen (29.03%) signs are closer.

5.2. Qualitative analysis

When the correct dictionary entry is not the first search result, it is often confused with signs that look similar. This can be due to similar movements (e.g., PAARD-A-8880 and GELIJK-A-4155), signing in a close-by location (e.g., WAAROM-A-13564 and MOP-A-17352), or using a similar hand shape (e.g., MELK-B-7418 and EENZAAM-B-3396). There are also confusions that we cannot explain (e.g., HEBBEN-A-4801 and ASVERSTROOIING-A-705). This is a drawback of using a black box approach like ours.

When monitoring the participants, we observed that search results were less relevant if the sign was not executed correctly. For example, one participant executed the sign WAAROM-A-13564 in the wrong location (on the left shoulder instead of the right shoulder). After we pointed this out, the participant tried again and obtained the correct search result. Note that this influences the accuracy values that we obtain.

5.3. Runtime

Processing a two-second query sampled at 15 frames per second takes 3656 milliseconds on a laptop CPU (an eight-core i7-8650U clocked at 1.9GHz) on average. The majority of the time is spent in MediaPipe (3620 ms or 120 ms per frame). The normalization and imputation of the keypoints takes 13 ms. Transforming the NumPy array to a PyTorch tensor of the correct shape takes 1.3 ms. Model inference takes 16.4 ms. Comparing the embedding to all 500 pre-extracted embeddings takes 5.4 ms. Counting the recording time, a search operation takes about 5.6 seconds in total. This could be improved by a pipelined execution in which MediaPipe is run on every frame as soon as it becomes available. This optimization detail is left for future work.

6. CONCLUSIONS AND FUTURE WORK

We describe a proof of concept for a system to query a sign language dictionary with video data. We pre-train a model on the task of isolated sign recognition and map the dictionary videos onto the embedding space learned by this model. We then perform a dense vector search in this embedding space to query the dictionary. The system is able to return an arbitrary number of results, ordered by increasing distance to the query in the vector space. The proposed algorithm is able to retrieve not only seen signs, but also signs that our model was not trained on (essentially performing one-shot classification).

The limitations of our approach are as follows. First, we only consider a small subset of the VGT dictionary (500 out of 10025 signs). Expanding the searchable dictionary will increase the probability of mistakes. Second, our evaluation set is small. Only 105 executions of ten signs were considered. Third, we do not take left-handed signers into account. Fourth, our evaluation set consists of videos recorded by inexperienced hearing signers. This adds a confounding factor to the evaluation of our system in the form of signing mistakes. Finally, we do not explicitly model sign parameters. Doing this may lead to more accurate search results. It may also allow end users to filter the search results based on these parameters, which could lead to increased user satisfaction.

In future work, a robust evaluation strategy needs to be defined and implemented in collaboration with members of sign language communities. A larger and more varied evaluation set should be collected. The videos in this evaluation set should be recorded by inexperienced and experienced signers alike. The proficiency levels of these signers should be documented to allow for a more robust evaluation of the system in various scenarios. Due to the large number of lexical items in the VGT dictionary, future querying systems may choose to take into account the five sign parameters, rather than searching in a single dense vector space.

7. REFERENCES

[1] M. Van Herreweghe, M. Vermeerbergen, K. De Weerdt, and K. Van Mulders, "Woordenboek Nederlands–Vlaamse Gebarentaal/Vlaamse Gebarentaal–Nederlands," online, (`https://woordenboek.vlaamsegebarentaal.be/`), accessed: 10 February 2023.

[2] Nederlands Gebarencentrum, "Gebarenwoordenboek," online, (`https://ow.gebarencentrum.nl/`), accessed: 10 February 2023.

[3] J. Kruskal, "An overview of sequence comparison: Time warps, string edits, and macromolecules," *SIAM review*, vol. 25.2, pp. 201–237, 1983.

[4] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of CVPR*, 2017.

[5] M. Fragkiadakis, V. Nyst, and P. van der Putten, "Signing as input for a dictionary query: matching signs based on joint positions of the dominant hand," in *Proceedings of the LREC 9th Workshop on the Representation and Processing of Sign Languages*, 2020, pp. 69–74.

[6] M. Fragkiadakis and P. van der Putten, "Sign and search: Sign search functionality for sign language lexica," in *Proceedings of Machine Translation Summit XVIII*, 2021.

[7] V. Athitsos et al., "Large lexicon project: American sign language video corpus and sign language indexing/retrieval algorithms," in *Proceedings of Sign-lang@LREC*, 2010, pp. 11–14.

[8] M. Wijkhuizen, O. Crasborn, and M. Larson, "A sign similarity approach to an information retrieval inspired visual dictionary for sign language learners," *Computational Linguistics in the Netherlands Journal*, vol. 12, pp. 287–309, 2022.

[9] J. Fink, P. Poitier, M. Andre, L. Meurant, B. Frenay, and A. Cleve, "Dictionnaire contextuel langue des signes belge francophone vers francais," online, (`https://dico.corpus-lsfb.be/`), accessed: 10 February 2023.

[10] M. Van Herreweghe, M. Vermeerbergen, E. Demey, H. De Durpel, H. Nyffels, and S. Verstraete, "Het corpus vgt. een digitaal open access corpus van videos and annotaties van vlaamse gebarentaal, ontwikkeld aan de universiteit gent ism ku leuven," 2015.

[11] W. Stokoe, *Sign language structure: An outline of the visual communication systems of the American deaf*, Studies in Linguistics, 1960.

[12] R. Battison, *Lexical borrowing in American sign language*, 1978.

[13] S. Hassan, O. Alonzo, A. Glasser, and M. Huenerfauth, "Effect of sign-recognition performance on the usability of sign-language dictionary search," in *ACM Transactions on Accessible Computing (TACCESS)*, 2021, vol. 14, pp. 1–33.

[14] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 10096–10106.

[15] S. Hassan, A. Al Amin, A. Gordon, S. Lee, and M. Huenerfauth, "Design and evaluation of hybrid search for american sign language to english dictionaries: Making the most of imperfect sign recognition," in *Proceedings of theCHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–13.

[16] M. De Coster, M. Van Herreweghe, and J. Dambre, "Towards automatic sign language corpus annotation using deep learning," in *6th Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, 2019.

[17] B. Shi, D. Brentari, G. Shakhnarovich, and K. Livescu, "Open-domain sign language translation learned from online video," in *arXiv:2205.12870*, 2022.

[18] V. Bazarevsky, I. Grischchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," in *arXiv:2006.10204*, 2020.

[19] S. K. Ko, J. G. Son, and H. Jung, "Sign language recognition with recurrent neural network using human keypoint detection," in *Proceedings of the conference on research in adaptive and convergent systems (RACS)*, 2018, pp. 326–328.

[20] M. De Coster, M. Van Herreweghe, and J. Dambre, "Sign language recognition with transformer networks," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020.

[21] A. Moryossef et al., "Evaluating the immediate applicability of pose estimation for sign language recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3434–3440.

[22] M. Vazquez-Enriquez, J. L. Alba-Castro, L. Docio-Fernandez, and E. Rodriguez-Banga, "Isolated sign language recognition with multi-scale spatial-temporal graph convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3462–3471.

[23] Google LLC, "Mediapipe holistic," online, `https://google.github.io/mediapipe/solutions/holistic.html`, accessed: 10 February, 2023.