Handling sign language transcription system with the computer-friendly numerical multilabels

Sylwia Majchrowska^{1,2} , **Marta Plantykow**^{3} , **Milena Olech**^{3} ,
^{1}Wroclaw University of Science and Technology, Wrocław, Poland
^{2}AI Sweden, Göteborg, Sweden
^{3}Intel Technology Poland, Gdańsk, Poland
sylwia.majchrowska@ai.se, m.plantykow@gmail.com, milena.w.olech@gmail.com

Abstract

This paper presents our recent developments in the automatic processing of sign language corpora using the Hamburg Sign Language Annotation System (HamNoSys). We designed an automated tool to convert HamNoSys annotations into numerical labels for defined initial features of body and hand positions. Our proposed numerical multilabels greatly simplify annotations' structure without significant loss of gloss meaning. These numerical multilabels can potentially be used to feed the machine learning models, which would accelerate the development of vision-based sign language recognition. In addition, this tool can assist experts in the annotation process and help identify semantic errors. The code and sample annotations are publicly available at https://github.com/hearai/parse-hamnosys.

1 Introduction

People communicate to exchange information, feelings, and ideas. Verbal communication is the primary method of human interaction. Those who speak and listen must use the same type of code (language) to communicate effectively. Communication outside the Deaf community is a huge challenge, as sign languages (SLs) differ from spoken languages. The inability to speak makes Deaf people one of the excluded groups in society.

Moreover, different nationalities use different versions of SL, and there is no universal one. SLs are natural human languages with their own grammatical rules and lexicons. Therefore, developing an efficient system capable of translating spoken languages into sign languages and vice versa would significantly improve two-way communication. However, designing and implementing such a machine learning (ML) system requires a large dataset with appropriate annotations.

Traditionally, SL corpora intended for ML training have been annotated at the gloss level. In the simplest terms, a gloss is a label. Nevertheless, the same word in different SL can be represented by other gestures. Therefore, ML methods need large datasets for each SL separately to achieve good results. Although there is no standard notation system for SL, the Hamburg Sign Language Notation System (HamNoSys) defines a unified set of written symbols that the creators of different SL datasets can use. This system consists of transcriptions of non–manual and manual features that describe a given sign's shape, orientation, position, and hand movement.

Although HamNoSys is language-agnostic, there are not many ML solutions using this notation system. First, the grammar of HamNoSys is challenging to understand without specialized linguistic knowledge. Moreover, using HamNoSys in its raw form in ML-based applications would be extremely difficult. An extensive training database is needed to perform this task since using HamNoSys labels would change the classification technique from single-class to multi-class compared to gloss labels. Furthermore, the inconsistent order of symbols in HamNoSys annotations amongst different datasets makes it almost impossible to process the labels correctly.

To address the problem described above, we created a solution for the automatic processing of SL corpora that transforms HamNoSys labels into numerical labels for carefully defined classes representing the initial body and hand positions. The numerical labels thus created would support the training of language-agnostic ML models since such multilabel annotations are easier to interpret by users unfamiliar with HamNoSys annotations. Moreover, the proposed solution allows for further simplification of the annotation structure, making it easier to develop a suitable model. The presented solution is unique and a publicly available tool that allows arbitrary selection of appropriate classes to describe a specific part of a given gesture.

2 Related Works

In recent years, work around HamNoSys has focused on creating large multilingual lexicons and simplifying the labeling procedure. Since the grammar of the notation system is complex and known only to sign language experts, the main effort has been in annotation processing [Jakub and Zdeněk, 2008; Power *et al.*, 2019; Skobov and Lepage, 2020]. Although not all available SL corpora are annotated using HamNoSys, the unified annotation system has attracted the attention of many researchers [Koller *et al.*, 2016; Dhanjal and Singh, 2022] because it can significantly simplify multilingual research.

2.1 HamNoSys-annotated SL Corpora

Deep learning (DL) based approaches are a promising method for the process of sign language translation. However, they need a large amount of adequately labeled training data to perform well. As HamNoSys annotations are not language-dependent, they can facilitate cross-language research. Moreover, merging several multilingual corpora will increase the available data resources. Below is a comprehensive overview of the existing corpora supplied with HamNoSys labels.

GLEX: The Fachgebärden Lexicon – Gesundheit and Pflege [Hanke *et al.*, 2020] was developed at the German Sign Language Institute of the University of Hamburg between 2004 and 2007 and contains a total of 2330 signs. This dataset is dedicated to technical terms related to health and nursing care.

GALEX: The Fachgebärden Lexicon – Gärtnerei und Landschaftsbau [Hanke *et al.*, 2020] was also created at the Hamburg Institute of German Sign Language between 2006 and 2009. This dataset consists of 710 signs related to technical terms for landscaping and gardening.

BL: The Basic Lexicon is part of the multilingual[1] three-year research project DICTA-SIGN [Matthes *et al.*, 2012], carried out since 2009 by a Consortium of European Universities[2]. About 1k signs are provided for each SL. The shared list of glosses covers the topic of traveling.

CDPSL: The Corpus-based Dictionary of Polish Sign Language (PJM) [Linde-Usiekniewicz *et al.*, 2014] was created in 2016 at the Sign Linguistics Laboratory of the University of Warsaw. The dictionary was developed based on the PJM corpus, which collected the video data of 150 deaf signers who use PJM. CDPSL allows documenting and describing authentic everyday use of PJM.

GSLL: The Greek Sign Language Lemmas [Efthimiou *et al.*, 2009; Theodorakis *et al.*, 2014; Kratimenos *et al.*, 2021] is developed by the National Technical University of Athens and supported by the EU research project DICTA-SIGN. The dataset is dedicated to isolated SL recognition and contains 347 glosses signed by two participants, where each sign is repeated between 5 and 17 times.

2.2 HamNoSys Processing Tools

In the first decade of the 21st century, Kanis *et al.* [Jakub and Zdeněk, 2008] proposed a HamNoSys editor called an automatic signed speech synthesizer. This tool can generate a sign animation based on the input HamNoSys label. The created application can accept the input label in two modes: one designed for direct insertion of HamNoSys symbols and the other with more intuitive graphical interfaces. The main goal of this solution was to allow an inexperienced annotator to annotate signs correctly. Unfortunately, it has several limitations, as it contains no more than 300 signs in Czech Sign Language and is based on a domain-specific lexicon.

Power *et al.* [Power *et al.*, 2019] conducted a historical sign language analysis based on 284 multilingual[3] signs annotated with HamNoSys symbols. They developed an open-source Python library to facilitate statistical analysis for manipulating HamNoSys annotations. As a result, they created the first publicly available tool capable of parsing HamNoSys annotations. However, the proposed parser can only work with sign language data if the HamNoSys labels satisfy strict assumptions. The symbols must appear in the gloss notation in a specific order and be separated by spaces. Consequently, the tool cannot be used on existing SL corpora without initial preprocessing.

In 2020, an automatic annotation system from video-to-HamNoSys based on the HamNoSys grammar structure was presented [Skobov and Lepage, 2020]. The proposed solution is based on virtual avatar animations created using the JASigning platform[4]. This approach generated a HamNoSys label based on a given sign animation. However, the Skobov *et al.* claimed that the proposed methodology with some modifications could also be used to obtain better results on the real data.

2.3 ML-based approaches using HamNoSys

There is a visibly growing interest in sign language recognition from video recordings using ML methods. Some use HamNoSys labels in a reduced form, as the access to large databases is limited. The most popular approaches use classification networks.

In 2016, Koller *et al.* [Koller *et al.*, 2016] used part of HamNoSys annotations describing the hand orientation modality. They trained classifier on isolated signs in Swiss-German and Danish Sign Language. They applied it to the continuous sign language recognition task on the RWTH-PHOENIX-Weather corpus [Forster *et al.*, 2012] containing German Sign Language. This multilingual approach significantly reduced word errors.

Recently, Hidden Markov Models were also used to directly translate multilingual speech into Indian Sign Language [Dhanjal and Singh, 2022]. In this approach, HamNoSys annotations provided an intermediate step between spoken and sign language. Additional conversion of speech into human-readable text was not needed. Despite the successful creation of an Indian speech recognition system, it was pointed out that building an optimized model requires a sufficient amount of data in an adequately transcribed format. This limitation has yet to be overcome.

3 Hamburg Sign Language Notation System

The Hamburg Sign Language Notation System (HamNoSys) is a *phonetic* transcription system that has been widespread for more than 20 years [Hanke *et al.*, 2020]. HamNoSys does

[1] British Sign Language, German Sign Language, Greek Sign Language, and French Sign Language

[2] Institute for Language and Speech Processing, Universität Hamburg, University of East Anglia, University of Surrey, Laboratoire d'informatique pour la mécanique et les sciences de l'ingénieur, Université Paul Sabatier, National Technical University of Athens, WebSourd

[3] American Sign Language, Flemish Sign Language, Mexican Sign Language, French Sign Language, French Belgian Sign Language, Brazilian Sign Language

[4] https://vh.cmp.uea.ac.uk/index.php/JASigning

not refer to different national finger alphabets and can therefore be used internationally. HamNoSys font consists of more than 210 symbols, which encode the initial position of the signer and basic movement of the sign [Hanke, 2004]. It can be divided into six basic blocks, as presented in Fig. 1 (upper panel). The first two out of six blocks – symmetry operator and non–manual features – are optional. The remaining four components – handshape, hand position, hand location, and movement – are mandatory [Smith, 2013].

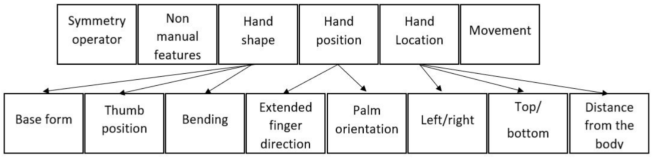

Figure 1: The structure of HamNoSys consists of six main blocks, four of which are mandatory and two are optional. Additionally, **Handshape, Hand position** and **Hand location** blocks are divided into components that specify particular parts of the HamNoSys notation.

Each HamNoSys part mentioned above has the following meaning [Hanke, 2004]:

- **Symmetry operator** (optional): denotes two-handed signs and determines how attributes should be mirrored to the non–dominant hand unless otherwise specified.
- **Non–manual features** (optional): represents non–manual features (e.g., puffed or sucked-in cheeks) that can be used to describe a given sign.
- **Handshape**: refers to the handshape description, which is composed of three subblocks – **Base form, Thumb position**, and **Bending**.
- **Hand position/orientation**: describes the orientation of the hand using two subblocks – **Extended finger direction**, which can specify two degrees of freedom seen from signer's, birds' or from-the-right view, and **Palm orientation**, as third degree of freedom, defined relative to the extended finger direction.
- **Hand location**: is split into three components – **Location left/right** specifies x coordinate, **Location top/bottom** for y coordinate, and **Distance** (that is skipped if natural) specifies the z coordinate.
- **Movement/Action**: represents a combination of path movements that can be specified as targeted/absolute (location) or relative (direction and size) movements.

4 Decision Tree-based HamNoSys parser

The main goal of the parser is to translate a label representing a SL gloss, written in HamNoSys format, into a form that can be used for DL-based classification. Since the structure of the HamNoSys grammar can be described as a decision tree [Skobov and Lepage, 2020], we used the method to decompose the notation into numerical multilabels for the defined classes. The parser logic implements the rules underlying data with the sequential structure to analyze a series of symbols. It matches each symbol with the class that describes it (while assigning it the appropriate number) or removes it. The fig. 2 shows a diagram of how a parser works.

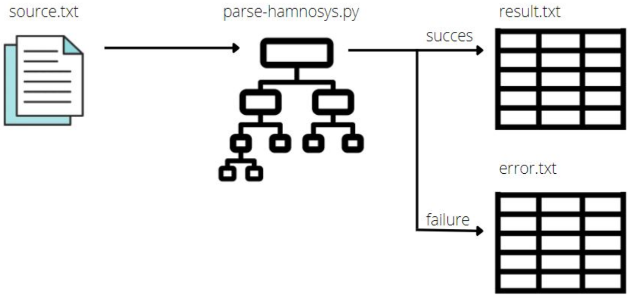

Figure 2: A schema of action of implemented parser. The source file containing HamNoSys notations is delivered to the parser which is a decision tree and transforms HamNoSys glosses into numeric labels. In case of success, the data are written to the result text file, otherwise error file is updated.

4.1 Methodology

As described in Section 3, HamNoSys labels can be represented by blocks. In our implementation, four blocks (symmetry operator, location left/right, location top/bottom, distance from the body) refer to the overall human posture In comparison, five blocks (handshape base form, handshape thumb position, handshape bending, hand position extended finger direction, and hand position palm orientation) relate to a single hand.

Furthermore, both hands (dominant and non–dominant) can be involved in each sign. Up to two HamNoSys symbols describing a single class can be assigned for each hand. To properly store all symbols, the classes related to a single hand are repeated four times as the primary and secondary descriptions for the dominant hand and the primary and secondary descriptions for the non–dominant hand. Moreover, we added one extra class that indicates if the sign description starts from a relaxed hand sign. As a result, together with classes describing overall human posture, the parser considers 25 classes when analyzing HamNoSys labels. The fig. 3 presents the numerical values and assigns HamNoSys symbols to them.

Class **Symmetry operator**, describes how the description of the dominant hands maps to the non–dominant hand using nine numbers. A value of class **NonDom first** equals 1 if the sign description starts from a relaxed hand sign, otherwise is set to 0.

The following 20 classes contain primary and secondary descriptions of five features assigned separately for dominant and non–dominant hands. Secondary classes will be assigned if the \ operator is used. For example, if A\B construct is used, symbol A will be assigned to the primary class, and symbol B will be assigned to the secondary class.

The **Handshape base form** class describes the base form of a handshape in the initial posture. This class value can be in the range of 0 to 11. The appearance of the class symbol is expected right after the symmetry operator. Only three symbols: relaxed hand, ˜ or [can occur in between. The first one, indicating a relaxed hand, will be assigned a numeric label, and the other two will be omitted. An error will occur if this symbol is not found in the expected place.

The **Handshape thumb position** class describes the position of a thumb in the initial posture and has four possibilities. This class symbol can only be used right after the *handshape base form* symbol or *bending* symbol. Otherwise, an error will occur. The **Handshape bending** class describes the hand bending in the initial posture. This class symbol is in the range 0 to 5 and can be used only right after the *handshape base form* symbol or *handshape thumb position* symbol. Otherwise, an error will occur. Fig. 3 presents the thumb position and bending combined with the handshape base form icon to increase its readability.

The **Hand position extended finger direction** class specifies two degrees of freedom. This class value can be in the range of 0 to 17. This class must occur at least once in the HamNoSys label. Otherwise, an error will occur since this class is mandatory. The **Hand position palm orientation** class is in the range 0 to 7 and specifies the third degree of freedom. Like the previous one, this class is mandatory. The symbol must be found in the HamNoSys label at least once - if not, an error will occur.

Three remaining classes, **Hand location L/R** (Left/Right) **Hand location T/B** (Top/Bottom) and **Hand location distance**, describe the hand location, using x, y and z coordinates respectively. The first is 0 to 4, the second uses 37 symbols, and the last class values are between 0 and 5. For **Hand location L/R** class, the symbol position is analyzed in relation to the **Hand location T/B** symbol (see Fig. 3).

As previously described, there are three mandatory symbol classes the parser shall find in the HamNoSys label: *hand position base form*, *hand extended finger direction*, and *hand palm orientation*. If any of those classes are not found, an error will occur.

As suggested in sources [Hanke, 2004], if the location information is missing, the default *neutral* values are assigned automatically, meaning 0 for distance, 0 for left/right, and 14 for top/bottom. All classes not found in the HamNoSys label are marked as *NaN* to distinguish them easily. An error when parsing a particular class will be indicated by a negative number assigned to the class.

4.2 Example usage

The open-source code is implemented in Python programming language, making it more flexible and user-friendly and allowing it to incorporate commonly used Python libraries such as Pandas.

The package consists of two main scripts: *parse-hamnosys*, which is the primary driver, and *hamnosys_dicts*, which is the dictionary file. The *parse-hamnosys* script can be used to convert a HamNoSys encoding into numerical labels. It requires arguments indicating the source file containing HamNoSys notation and destination files that separately contain successfully and not successfully parsed results. Moreover, two optional arguments specify the names of columns in the input and output text files.

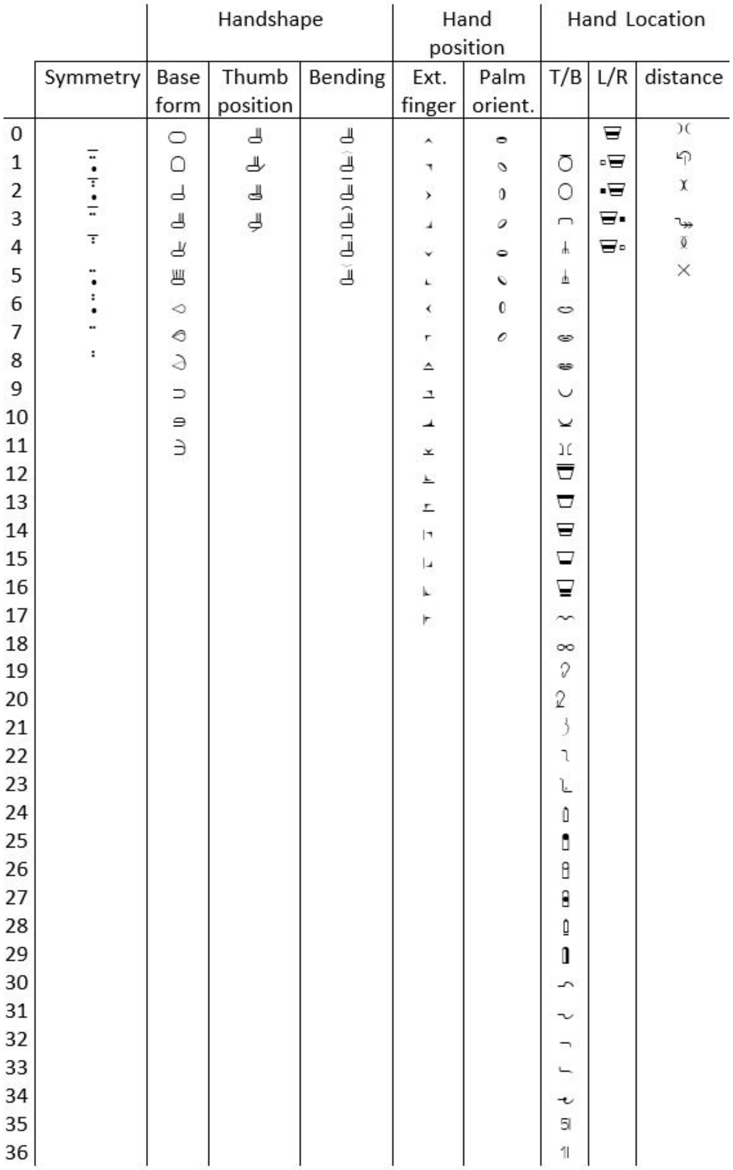
Handshape
Hand position
Hand Location
Symmetry
Base form
Thumb position
Bending
Ext. finger
Palm orient.
T/B
L/R
distance
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

Figure 3: Parser classes numerical values and its counterpart character in HamNoSys font. The thumb position and bending are presented in combination with the handshape base form symbol to increase its readability.

Listing 1 gives an example of the command to use the parser with the mentioned above parameters. As a result, two files, *source.txt* and *error.txt*, will be generated. The first of them contains correctly parsed glosses with their classes. The other one includes glosses that were not parsed correctly.

Listing 1: Basic example of calling *parse-hamnosys* script.

```
python3 parse-hamnosys.py \
    --src_file source.txt \
    --dst_file result.txt \
    --err_file error.txt
```

Table 1 presents a few examples from two different datasets. The example 1 stands for word $E\Pi I\Sigma KE\Upsilon AZ\Omega$ (encoded as *7 0 0 0 0 0 5 2 0 4 NaN 0 0 0 1 6 2 0 4 NaN NaN 0 13 0*, which each position of numerical label is related to specific HamNoSys block) and comes from GSLL dataset. The second example comes from Basic Lexicon dataset and stands for word *know*. The term NaN emphasizes that a given feature of a gloss was not specified in the HamNoSys string.

Class			Ex. 1	Ex. 2
	Symmetry		7	0
	NonDom first		0	0
Dom. 1	Shape	Base form	0	0
		Thumb	0	1
		Bending	0	0
	Position	Ext.finger dir.	0	0
		Palm	5	4
Dom. 2	Shape	Base form	2	NaN
		Thumb	0	NaN
		Bending	4	NaN
	Position	Ext.finger dir.	NaN	7
		Palm	NaN	2
Ndom. 1	Shape	Base form	0	NaN
		Thumb	0	NaN
		Bending	0	NaN
	Position	Ext.finger dir.	1	NaN
		Palm	6	NaN
Ndom 2	Shape	Base form	2	NaN
		Thumb	0	NaN
		Bending	4	NaN
	Position	Ext.finger dir.	NaN	NaN
		Palm	NaN	NaN
	Location	x	0	4
		y	13	3
		z	0	0

Table 1: The example of the final numerical labels created by the parser for two glosses (separate words) – $E\Pi I\Sigma KE\Upsilon AZ\Omega$ (Ex.1) and *know* (Ex.2) – from two databases. The NaN occurs when no symbol in the HamNoSys source string encodes the manual or non–manual feature (i.e., the Ex.2 is described only for the dominant hand).

4.3 Error handling

Decision tree-based parser implementation predefines possible HamNoSys label formats. If the parser does not recognize the order of symbols in the HamNoSys label, an error will be announced by filling the given column with a negative value.

We used the open part of each database mentioned in Section 2.1 to analyze the parser effectiveness. The gathered collection of datasets consists of around ten hours of videos accompanied by 11831 glosses, where 7095 are unique.

Table 2 presents the number of successfully parsed entries for each dataset analyzed. The *# glosses* column represents the number of all glosses passed to the parser as an input. The *# correct* column contains the number of correctly parsed glosses. In contrast, the *# errors* column shows the number of HamNoSys labels in which the order of symbols was not properly recognized. The parser effectiveness η_{p} was calculated as the percentage of entries that were correctly decomposed and parsed by the parser.

Table 2: Number of successfully parsed entries for each dataset.

Dataset	**#**	**#**	**#**	η_{p}
name	**glosses**	**parsed**	**errors**	**(%)**
GALEX	568	561	7	98.77
GLEX	829	778	51	93.85
CDPSL	2835	2828	7	99.75
BL	4123	3907	216	94.76
GSLL	3476	3316	160	95.40

For each of the analyzed datasets parser successfully decomposed over 93% of entries, reaching the lowest effectiveness when analyzing the GLEX dataset that consists of 829 entries (93.85%) and the best result for CDPSL (99.75%) analyzing 2835 notations.

As previously mentioned, the parser's main logic is based on a concept of a predefined decision tree, so the possible order of symbols in HamNoSyS notation is predefined. Additional manual analysis of notations that were not correctly processed by the parser demonstrated that in most cases, a label was not processed correctly as the predefined order of symbols was not applied. In some cases, such as duplicated hand shape bending symbol, additional logic was implemented to handle exceptions (remove excessive symbol). Unfortunately, it was impossible to forecast and serve all possible deviations.

The highest processing efficiency was reached for the CDPSL dataset, which can be connected with the time that the authors of the parser have spent analyzing this particular dataset and all possible workarounds that will help preserve the predefined order of symbols. The difference in obtained results proves that the expected order of symbols is not equally preserved in all of the datasets, meaning that all of the datasets were annotated in a different manner. If an error occurs, it is highly advised to manually analyze the order of symbols in the HamNoSys string placed in *error.txt* file.

5 Data reduction influence analysis

As a final analysis, we performed backward decoding from the assigned numeric multilabels to the correct glosses. In this study, we evaluated the impact of information reduction caused by the proposed HamNoSys label encoding methodology.

It was assumed that the decoding process was successful if the parser assigned one HamNoSys label to reach the tested gloss. If it assigned more than one label, the verification was considered unsuccessful. Due to database diversity and significant data differences, the results were verified on each of them separately. The lowest decoding efficiency η_{d} was observed for Basic Lexicon, which is a set that consists of four SLs. For this database, the number of glosses assigned to a single HamNoSys label (and thus misidentified) reached the highest value of 12. It is the same as in the context of spoken language, where there are similar words, there may be similar gestures that describe different glosses, especially in multilingual databases such as BL. Moreover, the primary purpose of creating BL was to identify common signs between the four sign languages.

Table 3 summarizes the influence of data reduction on each dataset. The *# unique glosses* column indicates the total number of distinctive glosses processed by the parser and assigned to HamNoSys labels. The *# singly labeled* and *# repeated* columns show the number of singly and multiply decoded glosses, respectively. Finally, the calculated high decoding efficiency $\eta_{\mathrm{d}} > 83\%$ proves that the parser can correctly recognize an individual gloss from different SLs. It is worth noting that some level of similarity (when the signatures of different glosses are similar) is inevitable due to the homonymy of language. We conclude that data reduction has no significant negative impact on possible invert decoding from numerical multilabels into glosses.

Table 3: A summary of the impact of data reduction on individual datasets that presents a total number of unique glosses in each dataset, number of glosses singly assigned to a unique HamNoSys label (consequently gloss), and the number of repeated assignments. The last column shows the parser's capability to recognize a single isolated gloss.

Dataset name	**# unique glosses**	**# singly labelled**	**# repeated**	η_{d} **(%)**
GALEX	514	484	30	94.16
GLEX	723	684	39	94.61
CDPSL	2480	2259	221	91.09
BL	3078	2580	498	83.82
GSLL	300	283	17	94.33

6 Conclusion

Nowadays, several lexical corpus collections include HamNoSys annotations. However, the way signs are annotated is not standardized. We decided to implement the HamNoSys parser as universally as possible to leverage and combine existing annotation efforts from different corpora. The parser focuses on converting HamNoSys symbols into a spatial-positional representation encoded by numerical multilabels to make existing corpora in this format useful for ML-based tasks and more straightforward to understand for non–linguistic researchers.

The proposed parser reduces the amount of data stored in the original HamNoSys character since it omits some data, such as movement (analyzing only the initial gloss position) or finger-related details. Nevertheless, the essential characteristic of the sign is preserved. We have also proved that backward decoding of the gloss is possible with efficiency above 83%.

The main benefit of this solution is that the created numerical labels can be used as input for multi-headed classification networks of the signer's initial position and then for recognition of targeted words in spoken language. This way of representing HamNoSys also simplifies its structure even further, e.g., encoding only the shape, orientation, and position of the dominant hand. We believe that the developed tool will contribute to future research efforts to create a fully functional sign language-agnostic translator.

Acknowledgements

The five-month non–profit educational project HearAI was organized by Natalia Czerep, Sylwia Majchrowska, Agnieszka Mikołajczyk, Milena Olech, Marta Plantykow, Żaneta Lucka Tomczyk. The main goals were to work on sign language recognition using HamNoSys and increase public awareness of the Deaf community. The authors acknowledge other team members: Natalia Czerep, Maria Ferlin, Wiktor Filipiuk, Grzegorz Goryl, Agnieszka Kamińska, Alicja Krzemińska, Adrian Lachowicz, Agnieszka Mikołajczyk, Patryk Radzki, Krzysztof Bork-Ceszlak, Alicja Kwasniewska, Karol Majek, Jakub Nalepa, Marek Sowa who contributed to the project. The authors acknowledge the infrastructure and support of Voicelab.ai in Gdańsk and the financial support of the Polish Development Fund (PFR) Foundation.

References

[Dhanjal and Singh, 2022] Amandeep Singh Dhanjal and Williamjeet Singh. An optimized machine translation technique for multi-lingual speech to sign language notation. *Multimedia Tools and Applications*, 2022.

[Efthimiou *et al.*, 2009] Eleni Efthimiou, Stavroula-Evita Fotinea, Christian Vogler, Thomas Hanke, John Glauert, Richard Bowden, Annelies Braffort, Christophe Collet, Petros Maragos, and Jérémie Segouat. Sign language recognition, generation, and modelling: A research effort with applications in deaf communication. In *Lecture Notes in Computer Science book series (LNISA,volume 5614)*, pages 21–30, 07 2009.

[Forster *et al.*, 2012] Jens Forster, Christoph Schmidt, Thomas Hoyoux, Oscar Koller, Uwe Zelle, Justus Piater, and Hermann Ney. Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3785–3789, Istanbul, Turkey, 05 2012. European Language Resources Association (ELRA).

[Hanke *et al.*, 2020] Thomas Hanke, Susanne König, Reiner Konrad, Gabriele Langer, Patricia Barbeito Rey-Geißler, Dolly Blanck, Stefan Goldschmidt, Ilona Hofmann, Sung-Eun Hong, Olga Jeziorski, Thimo Kleyboldt, Lutz König, Silke Matthes, Rie Nishio, Christian Rathmann, Uta Salden, Sven Wagner, and Satu Worseck. MEINE DGS. Öffentliches Korpus der Deutschen Gebärdensprache, 3. Release, 2020.

[Hanke, 2004] Thomas Hanke. Hamnosys-representing sign language data in language resources and language processing contexts. In *Proceedings of the LREC2004 4th International Conference on Language Resources and Evaluation*, volume 4, pages 1–6, 2004.

[Jakub and Zdeněk, 2008] Kanis Jakub and Krňoul Zdeněk. Interactive hamnosys notation editor for signed speech annotation. In *Proceedings of the 3rd Workshop on the Representation and Processing of Signed Languages: Construction and Exploitation of Sign Language Corpora*, 6 2008.

[Koller *et al.*, 2016] Oscar Koller, Hermann Ney, and Richard Bowden. Automatic alignment of hamnosys subunits for continuous sign language recognition. In *Proceedings of the LREC2016 10th edition of the Language Resources and Evaluation Conference*, 05 2016.

[Kratimenos *et al.*, 2021] Agelos Kratimenos, Georgios Pavlakos, and Petros Maragos. Independent sign language recognition with 3d body, hands, and face reconstruction. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 4270–4274. IEEE, 2021.

[Linde-Usiekniewicz *et al.*, 2014] Jadwiga Linde-Usiekniewicz, Małgorzata Czajkowska-Kisil, Joanna Łacheta, and Paweł Rutkowski. A corpus-based dictionary of polish sign language (pjm). In Andrea Abel, Chiara Vettori, and Natascia Ralli, editors, *Proceedings of the 16th EURALEX International Congress*, pages 365–376, Bolzano, Italy, jul 2014. EURAC research.

[Matthes *et al.*, 2012] Silke Matthes, Thomas Hanke, Anja Regen, Jakob Storz, Satu Worseck, Eleni Efthimiou, Athanasia-Lida Dimou, Annelies Braffort, John Glauert, and Eva Safar. Dicta-Sign – building a multilingual sign language corpus. In Onno Crasborn, Eleni Efthimiou, Stavroula-Evita Fotinea, Thomas Hanke, Jette Kristoffersen, and Johanna Mesch, editors, *Proceedings of the LREC2012 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon*, pages 117–122, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[Power *et al.*, 2019] Justin Power, David Quinto-Pozos, and Danny Law. Can the comparative method be used for signed language historical analyses?, 09 2019.

[Skobov and Lepage, 2020] Victor Skobov and Yves Lepage. Video-to-HamNoSys automated annotation system. In *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, pages 209–216, Marseille, France, May 2020. European Language Resources Association (ELRA).

[Smith, 2013] Robert Smith. *HamNoSys 4.0 User Guide*. Institute of technology Blanchardstown, 2013. Accessed: 2022-03-15.

[Theodorakis *et al.*, 2014] Stavros Theodorakis, Vassilis Pitsikalis, and Petros Maragos. Dynamic–static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition. *Image and Vision Computing*, 32(8):533–549, 2014.