HNS2CF: A Mapping Tool from HamNoSys to SL CatForm

Lisa Loy, Hope E. Morgan
Universität Hamburg, Universität Hamburg
lisa.loy@uni-hamburg.de, hope.morgan@uni-hamburg.de

Abstract

Over the past six decades, a variety of systems have been developed for representing sign language forms, from Stokoe Notation (Stokoe, 1960) to SignWriting (Sutton, 1999) and lexical database schemas. Each was designed with specific goals and applications, leading to a fragmented landscape of representations. To enable greater interoperability and data sharing among sign language users and researchers, we propose a robust approach to translating between notation systems. As a first step in this direction, we introduce a formal mapping framework between HamNoSys and the SL CatForm coding schema, describe its implementation, and present empirical evidence of its performance. An extensive evaluation of mapping mismatches revealed improvements to the mapping logic needed to further advance the HNS2CF mapping tool. However, the initial version of the system already achieves an overall accuracy of 76.7% and an in-depth analysis reveals that many apparent mismatches stem from annotator disagreement rather than mapping errors, indicating that the tool's actual accuracy is even higher. These results demonstrate the feasibility and promise of establishing mapping mechanisms across sign representation systems.

Keywords: phonological notation, HamNoSys, coding translation

1. Introduction

Many different ways of representing sign forms have been designed over the past 60+ years, from Stokoe Notation (Stokoe, 1960) to SignWriting (Sutton, 1999), to fields in lexical databases and much in between. Each of these systems was developed with certain goals and uses in mind. In order to promote flexibility for language users and researchers alike to fit a notation system to any given purpose, we argue there should be a robust effort to translate between systems so that data is not isolated. Indeed, this was the goal of the SignTyp conference and project over 15 years ago (Channon and van der Hulst, 2008), which sought to create a master dataset that creates a mapping between various notation systems, especially those used in research. Unfortunately, the momentum from that conference was lost and its goal was not followed up. In this paper, we pick up this thread again and introduce the first version of HNS2CF, an automated tool, mapping from the Hamburg Notation System (HamNoSys; Hanke, 2004) to the new SL CatForm coding schema (Morgan, 2026) to move towards interoperability in sign language corpora and resources.

This paper has three parts and main contributions: first, our HNS2CF mapping tool, including the formal mapping rules/relations and an implementation thereof; second, the Python code used to build the mapping tool[1]; and third, an empirical evaluation of the system's accuracy on German Sign Language (Deutsche Gebärdensprache, DGS) data.

[1] The code repository is vailable at https://codeberg.org/lisaloy/HNS2CF.git

2. Background

2.1. Hamburg Notation System

The Hamburg Notation System (HamNoSys) is a phonetic notation system designed to transcribe signs in a language-independent way. The notation system was originally developed in the 1980s and has undergone several improvements and extensions (see Prillwitz et al., 1987 for version 1 and Prillwitz et al., 1989 for version 2). In this paper, we use HamNoSys version 3 (Hanke, 2004). This notation system encodes signs as linear strings of iconic symbols that represent the manual part of a sign (handshape, orientation, location, movement) and optionally non-manual features (facial expression, mouth, gaze, etc.). The system was developed to be used in academic research rather than as a writing system for everyday use by Deaf communities. The HamNoSys alphabet encompasses roughly 200–210 characters (see Figure 1 for some examples). These symbols can be categorised into the following main groups: **handshape** (hand configuration, plus modifiers for finger selection, bending, and shape details), **orientation** (direction of palm and finger facing), **location** (where the sign is articulated), **movement** (path movements, in-place movements, repetitions, size, and manner), and **symmetry** markers (indicating how the hands relate to each other in two-handed signs). HamNoSys follows a specific syntax and is inherently combinatorial, mean-

Proceedings of the 12th Workshop on the Representation and Processing of Sign Languages, pages 287–296
16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

ing individual parameters of a sign are usually not encoded in only one character but rather a combination of multiple characters and diacritics. A basic handshape, for instance, is annotated using a main handshape character, combined with a character for finger specifications and relations and a character for palm orientation, with the ability to specify additional properties (e.g., bending, thumb position, etc.). An example of a HamNoSys notation for the sign PAIN3 is provided in Figure 1[2].

HamNoSys has been used extensively in areas such as: corpus annotation (e.g., Konrad et al., 2020), language documentation (e.g., Schmaling, 2000), sign language recognition (e.g., Koller et al., 2016), and sign animation/avatar technology (e.g., Elliott et al., 2004).

2.2. SL CatForm Coding Schema

SL CatForm (Morgan, 2026) is a more recent phonological coding schema, designed to represent minimal units of categorical form in sign languages for research purposes, particularly for computational analysis of large datasets. It derives from database fields for coding a range of phonological characteristics. SL CatForm builds on the descriptive analysis of one specific sign language, Kenyan Sign Language (KSL; as documented in Morgan, 2022). Version 1.0 includes only values from KSL (Morgan, 2026), but can be modified to include additional language-specific values. In fact, the same basic values have been used to encode phonological data of Israeli Sign Language in ISL-LEX (Morgan et al., 2022), and only a few new values were required for handshapes and path shapes. Thus, the basic categories (variables) are likely applicable in other sign languages.

The schema contains 42 variables representing aspects of sign form, with each variable consisting of values in single or double alphanumeric characters (see Figure 1 for an illustration). These variables are placed in a fixed order in the SL CatForm string for use in computational scripts; and they are grouped into six categories: **articulator** (variables *#0-6; overall sign type, handedness, and balance*), **handshape** (variables *#7-19*; whole handshape as well as individual features), **orientation** (variables *#20-21*; absolute palm and finger direction), **location** (variables *#22-25*; where the sign is articulated and lateral symmetry), **core movement** (variables *#26-34*; overall movement types and axes/shape of path movements), and **manner of movement** (variables *#35-42*; how movements are distributed in syllables; e.g., by repetition or a change in dominance). An example of a sign coded using the SL CatForm coding schema is given in Figure 1.

Compared to HamNoSys, the variables in the schema are fixed and independent, therefore the system does not employ rules for sequential ordering; i.e., a syntax. Yet, each variable requires a value, unlike HamNoSys which has the option of leaving out irrelevant or unspecified information. Also unlike HamNoSys, SL CatForm is not designed for ease of human readability since forms are represented by the fewest possible characters. It was also created to represent phonological types that are explicitly categorical — e.g., they occur in minimal pairs — rather than gradient phonetic phenomena. Relatedly, SL CatForm was developed in alignment with a particular model of sign language structure, described in Morgan (2022).[3] It also encodes information about syllable structure, which is not overtly encoded into HamNoSys.

SL CatForm has only recently been made publicly available, so it is not cited in previous literature; however a current study on phonological distance uses the SL CatForm string structure (Morgan et al., 2026).

3. Mapping Specifications

The goal of the HNS2CF mapping system is to convert HamNoSys strings (source domain) to SL CatForm strings (target domain). To achieve this, we started by working from the documentation in the current version of HamNoSys (Hanke, 2004) and the documentation for SL CatForm (Morgan, 2026), looking for overlaps in the information encoded in each system. For some phonological information, the mapping was straightforward; for example, path shape, the number of hands, or the finger/palm orientation. For these phonological aspects, the mapping system simply matches certain characters in the HamNoSys string (e.g., movement characters for path shape, symmetry markers for number of hands, or finger extension characters for absolute finger orientation) to the corresponding value in the SL CatForm system.

Other phonological information is only explicitly encoded in one system or the other. For instance, SL CatForm encodes lexical mouthing in variable *#6*, whereas there is no information on lexical mouthing encoded in HamNoSys. Due to its combinatorial nature, HamNoSys, in turn, offers a wider variety for possible handshapes, resulting in handshapes which do not (yet) exist in SL CatForm.

[2] All images used as examples of signs in this paper were taken from the Public DGS Corpus release 3 (Konrad et al., 2020).

[3] This model is based on the Dependency Model (van der Kooij, 2002), but incorporates observations about syllable structure from the Prosodic Model (Brentari, 1989).

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	37	38	39	40	41
M	2	S	B	A	U	0	12	0	12	H	S	N	S	0	H	S	N	S	0	C	U	T	5	0	S	P	C	M	0	O	0	0	0	0	M	E	0	0	0	U	0

Figure 1: The DGS sign PAIN3 (top) with the corresponding HamNoSys annotation (middle) and coding in SL CatForm (bottom)

Finally, there is some phonological information that is only implicitly encoded in one system or the other. For example, for path direction, HamNoSys uses its many arrow-based characters to unambiguously represent specific path directions, including diagonals. In contrast, SL CatForm uses a combination of three variables (*#28–30*) to encode path axis and directionality, but with less precision, and diagonals that move along three axes at once are not fully described. On the other hand, SL CatForm explicitly encodes the number of syllables of a sign, whereas this information is only implicitly encoded in the HamNoSys string. For such cases, the mapping required more complex logic; this is discussed in Section 3.1

After a comprehensive comparison of the two notation/coding systems and their inventories, we decided that two variables in the SL CatForm coding schema had to be excluded because the information needed to code these variables is not in HamNoSys: *#6 lexical mouthing* and *#42 secondary movement*. Specifically, the latter refers to how changes in the joints or palm orientation are distributed across syllables; e.g., the difference between finger flexion timed to the start and end of a path movement versus flexion repeating multiple times on a path. However, for full transparency, we keep these variables in their original position in the SL CatForm string, using the placeholder "xx", and consider them in the evaluation.

3.1. Mapping Relations

In order to transform these observations about how information is encoded in each system into a translational mapping in Python code, a classification of mappings was created. Table 1 shows a summary of four overall types of mapping relations that were used to extract information for each SL CatForm variable from the HamNoSys string. The most common mapping relation (25/40 variables in Table 1) from HamNoSys to SL CatForm was `n:1`, meaning multiple characters in the HamNoSys string map to a single value within a variable in the SL CatForm string. Eighteen variables have the most straightforward `1:1` mapping; i.e., direct matches from one HamNoSys character to one value in SL CatForm. Additionally, we developed two more mapping relations, `combination` and `pattern`. The `combination` mapping relation describes instances where the matching value for a variable could not be extracted from a single character in the HamNoSys string but from a fixed combination of individual HamNoSys characters. Similarly, the mapping relation `pattern` looks for combinations of HamNoSys characters; however, these are not fixed characters but can be any character from a set of characters. Once a `pattern` is found, this triggers one of two scenarios: a) set a specific value for a specific variable, or b) set a slicing point and split the HamNoSys string into two (or more) parts, then look only in one of the parts for matches. We present one example per mapping relation for the sign PAIN3 (presented in Figure 1) in Table 2. The mapping relation `pattern → set slicing point` was not found and activated in the PAIN3 example, so Figure 2 shows how this mapping relation would work for a two-handed sign with two different handshapes.

3.2. Technical Specifications

Our HNS2CF mapping tool was written in Python and consists of six functions, one for

SL CatForm category	SL CatForm variable	1:1	n:1	combination	pattern	
					set value	set slicing point
ARTICULATOR	0		✓		✓	
	1		✓		✓	
	2				✓	✓
	3		✓			
	4	✓		✓		
	5	✓				
	6					
HANDSHAPE	7				✓	✓
	8				✓	✓
	9		✓		✓	✓
	10		✓	✓	✓	✓
	11	✓	✓			✓
	12	✓	✓		✓	
	13	✓				✓
	14		✓			✓
	15		✓	✓	✓	✓
	16	✓	✓			✓
	17	✓	✓		✓	
	18	✓				✓
	19		✓			✓
ORIEN-TATION	20	✓	✓	✓	✓	
	21	✓	✓		✓	
LOCATION	22		✓		✓	
	23	✓	✓	✓		
	24	✓	✓	✓		✓
	25	✓		✓	✓	
CORE MOVEMENT	26	✓	✓			
	27	✓	✓	✓	✓	
	28		✓		✓	
	29		✓			
	30	✓	✓	✓	✓	
	31				✓	✓
	32			✓	✓	✓
	33				✓	
	34			✓	✓	
MANNER OF MOVEMENT	35	✓	✓		✓	
	36		✓			
	37				✓	✓
	38	✓				
	39					✓
	40		✓			
	41					

Table 1: Summary table of HNS2CF mapping relations used per variable (#0-#41) in the SL CatForm string

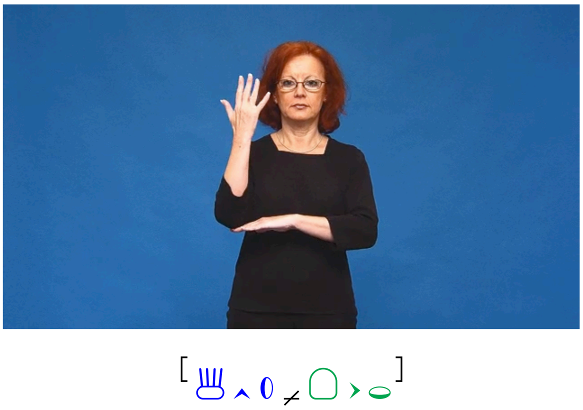
mapping relation	information encoded in HamNoSys	target variable + value in SL CatForm
1:1	~ hands move alternatingly	#4 hands move simult./altern. 'A' (alternating)
n:1	⸸ or ᵡ hands brush body or touch body	#28 path axis 1 'M' (midsagittal)
combination	○ +⸝ fist handshape with thumb out	#7 whole handshape H1 '12' (A-thumb)
pattern → set value	no ↣ followed by any location character location does not change	#24 2nd location '0' (no 2nd location)

Table 2: Examples for each mapping relation for the sign PAIN3 (see Figure 1)

the mapping of each of the six categories in the SL CatForm coding schema (see section 2.2.). Each function contains a set of rules and conditionalities employing the four mapping relations described above. There are seven lists in our code: `handshapechars` (containing all handshape characters), `locationchars` (containing all location characters), `movementchars` (containing all movement characters), `orientationchars` (containing all orientation characters), `palmchars` (containing only palm orientation characters), `fingerchars` (containing only finger orientation characters), and `symmetrychars` (containing all symmetry characters). These lists are accessed at various points throughout the code, especially during `n:1` mappings and to set up `patterns`. The HamNoSys strings were examined for combinations by indexing through a for-loop. To map the HamNoSys to the *whole handshape* variables in the SL CatForm string (variables *#7*, *#8*, and *#9*), we created a dictionary containing tuples of HamNoSys character combinations and their corresponding values in SL CatForm. This dictionary is accessed in the code to iterate through the handshape combinations and find matches in the HamNoSys annotation string. Similarly, to map the HamNoSys locations to the *major area* variable (*#22*) in the SL CatForm string, we created a dictionary containing HamNoSys location characters and their corresponding values for *major location* in SL CatForm. This dictionary is accessed to perform two operations: a) to find the corresponding value in the SL CatForm string for locations in the HamNoSys strings, and b) to set the value for variable *#22* to '2' only if there are two locations specified in the HamNoSys, which **do not** fall within the same major area. Finally, patterns were stored as Python variables, triggering TRUE if a pattern is found in the HamNoSys string or FALSE if a pattern is not found. Based on this boolean, a new set of mapping conditionalities was either activated or ignored.

3.3. Methodological Choices

Besides excluding the two variables (*#6 lexical mouthing* and *#42 secondary movement*) but keeping the placeholder 'xx' for readability, it was necessary to make certain methodological choices to account for ambiguities emerging during the creation of the HNS2CF mapping tool. For instance,



7	8	9	10	11	12	13	14	15	16	17	18	19
4	0	1	A	S	S	S	0	A	S	U	U	0

information encoded in HamNoSys	two-handed sign with two different handshapes
pattern → set slicing point	any orientation character ( ) followed by (plus in HamNoSys) followed by any handshape character ( )
target variable in SL CatForm	#7 + #10-14 for H1 and #9 + #15-19 for H2

Figure 2: Example for the mapping relation `pattern → set slicing point` for a two-handed sign (TREE2) with two different handshapes. Only the relevant handshape information from HamNoSys (top) and SL CatForm (bottom) are presented

the content of HamNoSys strings can vary from only the most basic information for a valid string to as much detail as the researcher/annotator can discern and chooses to annotate. For this reason, some information was encoded in HamNoSys that was not encoded in the SL CatForm coding schema. At the same time, there were also instances in which the HamNoSys strings for some signs lacked information; most often, this was for phonologically under-specified features. For example, the initial movement to a touch is typically unmarked in HamNoSys but encoded in SL CatForm. Another example is handshapes in which the thumb was not explicitly coded in HamNoSys due to its precise position being underspecified, such as in the V-handshape. However, thumb position is obligatorily encoded for all handshapes in SL CatForm. For the V-handshape, for instance, the thumb is marked as restraining/crossed. In these cases, we treated the thumb position as a match because it is a permissible thumb position for this handshape. Also, dispersed signs (i.e., including 'two touch' signs, like the sign DEAF1A[4], among others), are not uniformly encoded in HamNoSys. Therefore, we created a forced choice between dispersed signs and two-location signs: if there are two locations within the same major area (e.g., *forehead* and *chin* are both on the head), these are treated as dispersed signs within the same phonological location. When signs have two locations in different major areas (e.g., on the head and torso), these are encoded as two locations.

4. Data

There are four datasets in the current study: 1) the HamNoSys dataset, 2) a test dataset for evaluation, 3) an existing database of KSL signs coded in SL CatForm, and 4) the output dataset of DGS transformed into SL CatForm. First, the **HamNoSys dataset** is a list of types[5] in the Public DGS Corpus release 3 (Konrad et al., 2020) with a total of 14,476 types and their corresponding HamNoSys strings. This dataset was used for spot checks and to test intermediary steps in the creation of the HNS2CF tool, as well as being the source domain for the HNS2CF mapping. The total runtime for the HNS2CF mapping system on this dataset of 14,476 types was 1:05 minutes.

Second, to evaluate the accuracy of our HNS2CF mapping system, we created our **test dataset** by randomly selecting 100 sign types from the HamNoSys dataset. These were then manually coded according to the SL CatForm coding schema. This manual coding only used reference videos of the sign types and was therefore completely independent of the HamNoSys notations. During the analysis, however, we found two outliers, where the sign was coded as one-handed in the HamNoSys string but the corresponding video showed a signer producing the sign with two hands, or vice versa. The two outliers were removed from the test dataset and the final evaluation was performed on the 98 remaining signs in our test dataset. The total runtime for the HNS2CF mapping system on this test dataset of 98 types was 0.4 seconds.

The third dataset is **1,880 KSL signs coded in the SL CatForm coding schema**, contained in the KSL Lexical Database (Morgan, 2022). This dataset was used as a reference dataset at two points. First, it was used to address ambiguities arising from the theoretical mapping rules (section 3). Second, it was used in the evaluation step (section 5) to verify how certain phonological phenomena were coded in the past. This unpublished

[4] https://doi.org/10.25592/dgs.corpus-3.0-type-14040

[5] A type is equivalent to a citation form in a dictionary, distinguishable from individual tokens in a corpus.

dataset is part of a different project and the authors were granted access for the purpose of verifying mapping decisions.

The fourth dataset is the dataset of **DGS signs transformed into SL CatForm**. Currently, this dataset is not publicly available, pending further improvements to the mapping.

5. Evaluation Study

In order to find out how accurate the HNS2CF mapping tool worked, we created an evaluation study to compare the automated SL CatForm output with the hand-coded SL CatForm in the test dataset of 98 DGS signs. The results are divided into five categories: a) **matches** (i.e. the manual coding and the HNS2CF output were the same), b) **mismatches** (i.e. the manual annotator and the HNS2CF system assigned different values for a variable), c) **no match found** (i.e. the HNS2CF mapping tool did not find a match for this variable and kept the placeholder 'xx'), d) **no manual coding** (i.e. the manual coder was unsure and left the variable blank), and e) **mapping not possible** (i.e. variables *#6* and *#42*, which were not encoded in the HamNoSys strings).

5.1. General Results

Out of the 4,116 compared values (98 types * 42 variables), we found 76.7% (3,155 values; green) were matches, 16.9% (699 values) were mismatches 0.6% (24 values; red) were instances where the mapping tool could not find a suitable match; 1,1% (42 values; yellow) were empty manual annotation values because the manual coder was unsure of the correct value in the SL CatForm coding schema. These yellow values were due to the manual coder not being fluent in DGS; to avoid misinterpretations of the categorical form, it was left blank. Lastly, 4.8% (196 values; 98 types * 2 variables; grey) were not encoded in HamNoSys. Figure 3[6] shows an overview of the 98 signs in the test dataset and the result of the mapping for each of the 42 variables in the SL CatForm coding schema. As the first version of the HNS2CF mapping, these overall results were better than expected.

We further investigated if there was a difference in mapping performance by phonological parameter. When looking at the distribution according to the six categories in the SL CatForm coding schema (Table 3), one can see the mapping system performed least accurately for **orientation** with 52.1% of all values (91 values) being mismatches, whereas the system performed best for **articulator** with only 3.9% of all values (27 values) being mismatches. We attribute the poor performance for orientation to two factors. First, absolute orientation may be particularly sensitive to subjective differences between coders. Second, there is a notable difference in specificity between the two schemas; SL CatForm has many fewer categories than HamNoSys for orientation, so that edge cases may end up in a different category in the mapping.

It is interesting to note that instances where the mapping system could not find a suitable match (0.6%; 24 values) fall within only the **handshape** category (22 values) and **core movement** category (2 values). The empty manual annotation values (1.1%; 42 values), however, are spread across **orientation** (3 values), **location** (6 values), **core movement** (19 values), and **manner of movement** (14 values).

We also investigated which signs performed particularly well or poorly. The five signs with the fewest mismatches were EBB1[7] with 0 mismatches out of 42 possible matches, FOR2[8] with 1 mismatch, as well as END2[9], TO-PRAY1B[10], and WE1A[11] with 2 mismatches each. EBB1, the only sign with 100% accuracy, is a very simple two-handed sign with flat hands, symmetrical in neutral space, palms down, fingers outward, that moves straight downward once. The five signs with the most mismatches were PAIN3[12] and FEELING6[13] with 14 mismatches each, KINDERGARTEN1A[14] and CLOTHES1C[15] with 15 mismatches each, and PERFORMANCE3[16] with 17 mismatches. PAIN3 is shown in Figure 1. It is also two-handed and symmetrical, with two A-thumb handshapes, but with alternating arced or circular movement that have brushing contact on the upper torso. Therefore, the amount of phonological complexity may play a role in the mapping success.

5.2. Individual Variable Results

Furthermore, we were interested in how individual variables within each phonological parameter fared. These results are important because they pinpoint where improvements can be made in the mapping.

[6] This figure is intended to give an overview of the error/match distribution across the test dataset rather than a detailed analysis for each sign.

[7] https://doi.org/10.25592/dgs.corpus-3.0-type-14858
[8] https://doi.org/10.25592/dgs.corpus-3.0-type-13533
[9] https://doi.org/10.25592/dgs.corpus-3.0-type-15796
[10] https://doi.org/10.25592/dgs.corpus-3.0-type-14352
[11] https://doi.org/10.25592/dgs.corpus-3.0-type-14641
[12] https://doi.org/10.25592/dgs.corpus-3.0-type-13869
[13] https://doi.org/10.25592/dgs.corpus-3.0-type-13954
[14] https://doi.org/10.25592/dgs.corpus-3.0-type-15111
[15] https://doi.org/10.25592/dgs.corpus-3.0-type-18667
[16] https://doi.org/10.25592/dgs.corpus-3.0-type-17699

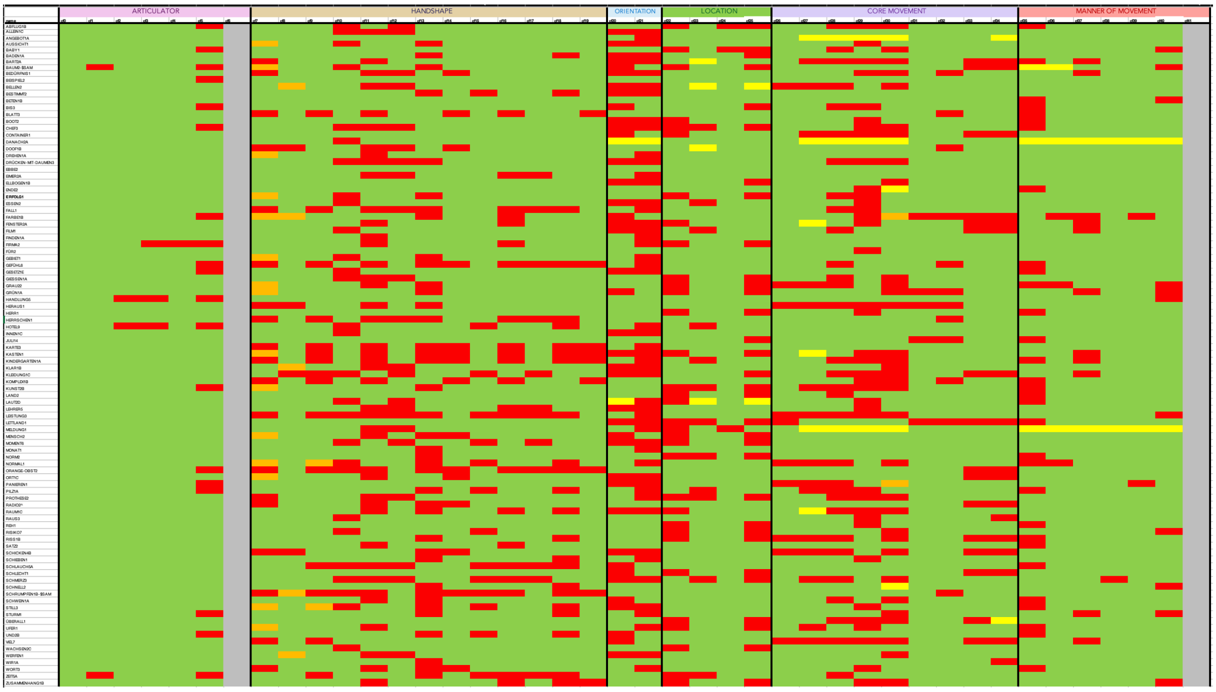


Figure 3: Colour-coded visualisation of the full evaluation results (matches = green, mismatches = red, no match found = orange, no manual coding = yellow, impossible mappings = grey). They x-axis represents the 42 SL CatForm variables, the y-axis represents the 98 signs in our test dataset

	matches	mismatches	no match found	no manual coding	not possible
articulator	561 (81.8%)	27 (3.9%)	0 (0%)	0 (0%)	98 (14.3%)
handshape	987 (77.5%)	265 (20.8%)	22 (1.7%)	0 (0%)	0 (0%)
orientation	102 (52.1%)	91 (46.4%)	0 (0%)	3 (1.5%)	0 (0%)
location	306 (78.1%)	80 (20.4%)	0 (0%)	6 (1.5%)	0 (0%)
core movement	684 (77.6%)	177 (20.1%)	2 (0.2%)	19 (2.2%)	0 (0%)
manner of movement	515 (75.1%)	59 (8.6%)	0 (0%)	14 (2.1%)	98 (14.3%)
total	3155 (76.7%)	699 (16.9%)	24 (0.6%)	42 (1.1%)	196 (4.8%)

Table 3: Match distribution according to the six main categories in the SL CatForm coding schema

We found that the HNS2CF mapping system performs the most accurately for variables *#0* (*overall sign type*; 100%), *#4* (*hands moving simult./alt.*; 98.9%), *#1* (*number of hands*; 97.9%), *#38* (*switch dominance*; 96.9%), *#24* (*second location*; 96.9%), *#2* (*same shape*; 96.2%), and *#39* (*switch orientation*; 95.9%).

In contrast, the mapping system performs least accurately for these five variables: *#21* (*direction fingers facing*; 46.9%), #20 (*direction palms facing*; 57.1%), *#11* (*H1 joint config.*; 57,1%), *#29* (*path axis 2*; 58.2%), and *#30* (*path directionality*; 59.2%). We analyse each of these in more detail below.

For **variable *#21, direction fingers facing***, most mismatches occurred when the manual coder assigned the value ‘Y’ (dynamic), meaning that the finger orientation changed during the sign, but the HNS2CF mapping system assigned a different value. However, we did not observe a predictable tendency or pattern; i.e. the mapping system did not confuse ‘Y’ for a single other value but assigned various other values instead of ‘Y’. This is due to the complex logic needed to extract information about finger orientation change in HamNoSys. Our HNS2CF mapping logic did find multiple instances of finger orientation change — even 9 instances the manual coder did not recognise as such — however, the system missed 8 instances of finger orientation change the manual coder did recognise and coded. Overall, within the 51 mismatches for *#21*, there were 31 mismatches where the mapping system actually worked correctly but the HamNoSys did not match the manual coder; these can be taken as successes. The other 19 mismatches were due to errors in the code for the HNS2CF tool, and can be addressed in future revisions of the tool.

For **variable *#20, direction palms facing***, most mismatches were caused by the mapping system assigning the value 'G' (diagonal) when the manual coder assigned a different value. However, there was no systematic patterning with a different value chosen by the manual coder. We found our HNS2CF logic still needs improvement in extracting the correct information from the HamNoSys due to the complexities of how palm orientation is encoded in HamNoSys (dependent also on the finger orientation). However, the 40 mismatches for variable *#20* are mostly not due to coding errors (only 12 mismatches) but due to differences in what the HamNoSys encodes and the values our manual coder assigned (25 mismatches). They should therefore be considered more of an annotator disagreement – that is, two annotators perceived certain aspects of a sign differently, leading to different annotations, which are not necessarily correct nor incorrect – than a mapping error.

For **variable *#11, H1 joint config.***, we observed that the HNS2CF mapping system tended to assign the value 'U' (curved) when the manual coder assigned the value 'H' (hooked). This was the case in 13 mismatches out of 42 (30.9%), but is not an error in the way the HNS2CF tool was coded but is either due to perception differences of what is considered curved or hooked, or it may also be slight phonetic differences between DGS and KSL handshape categories. There were, however, also 21 mismatches caused by an error in the HNS2CF logic: the mapping system incorrectly assigned the value 'C' (closed) when the manual coder assigned other values, such as the value 'S' (straight; 8 mismatches), the value 'U' (8 mismatches), or the value 'B' (bent; 5 mismatches). This coding error will be fixed in an updated version of the HNS2CF mapping system.

For **variable *#29, path axis 2***, most mismatches were caused by the mapping system assigning the value 'M' (midsagittal) when the manual coder assigned the value '0' (no second axis; 11 mismatches out of 37). Overall, the mapping system was more likely to confuse '0' values for other values (20 mismatches; 54.1%), meaning our HNS2CF mapping tool tends to code a second axis when, according to the manual coder, there is no second axis. Because this information is not directly encoded in the HamNoSys string, the mismatches are most likely caused by the mapping logic not being refined enough yet.

Finally, for **variable *#30, path directionality***, a tendency of the mapping system was to assign a different value when the manual coder assigned the value 'A' (away from body; 6 mismatches). However, there was no systematic patterning with a different value assigned by the mapping system. In general, we found the logic in our HNS2CF tool needs further refinement for diagonals and back-and-forth movements, as these caused 7 out of the 33 total mismatches within variable *#30*. However, it is important to note that variable *#30* was the variable with the highest number of missing manual annotations (5 values; 5.1%), suggesting that path directionality might be more complex to encode and various coders perceive path directionality differently. Indeed, out of the 33 mismatches for this variable, 13 were due to differences in what the HamNoSys encodes and the values our manual coder assigned rather than due to issues with the mapping tool and its implementation.

6. Discussion

This first version of the HNS2CF mapping tool already performs fairly well in the evaluation study, with 76.7% of variable matches in 98 random DGS signs. Further, as we unpacked the reasons for the non-matched values — i.e., everything that is not green in Figure 3 and Table 3 — we find that many mismatches do not point to problems with the HNS2CF mapping itself, but have other sources, such as the manual coder not feeling confident in coding the phonological form from one video alone (yellow cells in Figure 3 and Table 3), or the manual coder making mismatches for finger orientation that were later shown to be mapped correctly from the HamNoSys.

We also identified several ways to improve the mapping. One of the most straightforward to implement is to add language-specific values in certain phonological inventories in SL CatForm. The evaluation study had two handshape values that were not in SL CatForm. Handshape, location, and path shape inventories are known to differ across languages, and values like these can easily be expanded with new SL CatForm values. Other targets for improving the actual mapping tool were identified in the individual variable results in the previous section; e.g., refining the mapping logic for *path axis 2*.

We also found some indication that the degree of sign complexity may lead to more mismatches. We reached this conclusion by looking at the five signs with the highest and lowest accuracies. Those with the highest accuracy all have the following characteristics in common: they are signs in neutral space that have only path movement (no handshape or orientation movement; i.e., no secondary movement) and are marked as symmetrical for lateral symmetry. In other words, they were very simple signs. In contrast, the signs with the lowest accuracy were all two-handed and contained mismatches in orientation and in path axis 1; and many had secondary movements (handshape and/or orientation changes in the sign).

7. Conclusion & Future Directions

In this paper we have introduced the HNS2CF mapping system and tool. We presented the formal mapping relations, have outlined the system's implementation, and have shown empirical evidence of the system's quality. Despite it being only version 1 of this mapping system, we can already report an overall accuracy of 76.7%. Furthermore, the tool's total runtime of only 1:05 minutes on the DGS dataset with 14,476 types shows that the tool works efficiently, even with large datasets. However, a detailed analysis of mismatches in the evaluation study identified several places where the mapping logic should be revisited. Yet, the analysis also showed that the actual accuracy of the HNS2CF mapping tool is even higher than 76.7% as we have found multiple mismatches which were not due to errors with the tool but correctly mapped what was encoded in the HamNoSys to values in the SL CatForm coding schema. These mismatches therefore signal annotator disagreement rather than mapping errors.

We will continue to improve the current version of the HNS2CF mapping system, adding additional values to the SL CatForm coding schema where needed and adjusting the schema further to better fit DGS. Additionally, the efficiency of our mapping tool could potentially be further optimised by building a HNS2CF compiler. In line with this, we aim to build a reversed system, mapping from SL CatForm to HamNoSys, which could also serve as a validation tool and offer valuable insights into the kind and amount of information lost when mapping between those two systems. With some further adjustments, the mapping system has the potential to be extended not only to include a wider variety of sign languages and dialects, but its basic scaffolding could also be extended to other notation systems, for example, SignWriting. Further, the mapping system could be integrated into annotation pipelines to create even richer resources and larger datasets annotated in parallel in multiple notation systems and coding schemata.

8. Limitations

A test dataset of 100 signs is only a snapshot and may not be sufficiently representative of the DGS data; yet, annotating signs manually is time-consuming. After all, there are 4,100 separate data points within even this small dataset. Another limitation we encountered in the manual coding is grounded in the data itself. The SL CatForm coding schema is based on phonological categories. Without an existing phonological analysis of DGS describing its categorical forms, it was difficult to force some articulatory types into SL CatForm values. Finally, the HamNoSys data was annotated by multiple annotators whereas the SL CatForm data was annotated by a single annotator. In order to be able to fully account for annotator disagreement, this imbalance should be resolved and analysed as its own variable in future studies.

9. Acknowledgements

We would like to thank the DGS corpus team at the University of Hamburg for their input and continued support.

Research in this paper was funded by the European Union (ERC, *SemaSign*, No.101117395). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

10. Bibliographical References

Diane Brentari. 1989. *A prosodic model of sign language phonology*. MIT Press, Cambridge, MA.

Rachel Channon and Harry van der Hulst. 2008. The Phonetics and Phonology of Sign Languages First SignTyp Conference June 26-28, 2008. [Accessed 03-04-2026].

Ralph Elliott, John Glauert, Vince Jennings, and Richard Kennaway. 2004. An overview of the SiGML notation and SiGMLSigning software system. In *Proceedings of the LREC2004 Workshop on the Representation and Processing of Sign Languages: From SignWriting to Image Processing. Information techniques and their implications for teaching, documentation and communication*, pages 98–104, Lisbon, Portugal. European Language Resources Association (ELRA).

Thomas Hanke. 2004. HamNoSys – representing sign language data in language resources and language processing contexts. In *Proceedings of the LREC2004 Workshop on the Representation and Processing of Sign Languages: From SignWriting to Image Processing. Information techniques and their implications for teaching, documentation and communication*, pages 1–6, Lisbon, Portugal. European Language Resources Association (ELRA).

Oscar Koller, Hermann Ney, and Richard Bowden. 2016. Automatic alignment of HamNoSys subunits for continuous sign language recognition. In *Proceedings of the LREC2016 7th Workshop on the Representation and Processing of Sign Languages: Corpus Mining*, pages 121–128, Portorož, Slovenia. European Language Resources Association (ELRA).

Reiner Konrad, Thomas Hanke, Gabriele Langer, Dolly Blanck, Julian Bleicken, Ilona Hofmann, Olga Jeziorski, Lutz König, Susanne König, Rie Nishio, Anja Regen, Uta Salden, Sven Wagner, Satu Worseck, Oliver Böse, Elena Jahn, and Marc Schulder. 2020. MEINE DGS – annotiert. Öffentliches Korpus der Deutschen Gebärdensprache, 3. Release / MY DGS – annotated. Public Corpus of German Sign Language, 3rd release.

Hope Morgan, Amy Isard, and Ahn Dang. 2026. Improving phonological distance measures for signs: the CatFormCompare tool. In *Proceedings of the LREC2026 12th Workshop on the Representation and Processing of Sign Languages: Language in Motion*, Palma, Mallorca (Spain). ELRA Language Resources Association (ELRA).

Hope E. Morgan. 2022. *A Phonological Grammar of Kenyan Sign Language*. De Gruyter Mouton, Berlin, Boston.

Hope E. Morgan. 2026. SL CatForm coding schema v.1: A field-based phonological coding for sign languages (Version 1.0). Technical report, Universität Hamburg.

Hope E. Morgan, Wendy Sandler, Rose Stamp, and Rama Novogrodsky. 2022. ISL-LEX v.1: An online lexical resource of Israeli Sign Language. In *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources*, pages 148–153, Marseille, France. European Language Resources Association (ELRA).

Siegmund Prillwitz, Regina Leven, Heiko Zienert, Thomas Hanke, Jan Henning, et al. 1987. *HamNoSys. Hamburg Notation System for Sign Language. An Introduction*. Zentrum für deutsche Gebärdensprache, Hamburg, Germany.

Siegmund Prillwitz, Regina Leven, Heiko Zienert, Thomas Hanke, Jan Henning, et al. 1989. *HamNoSys. Version 2.0. Hamburg Notation System for Sign Languages. An Introductory Guide*. Signum, Hamburg, Germany.

Constanze Schmaling. 2000. *Maganar Hannu, language of the hands: A descriptive analysis of Hausa Sign Language*. Hamburg: Signum.

William C. Stokoe. 1960. *Sign language structure: An outline of the visual communications systems of the American Deaf*. Silver Springs, MD: Linstock Press.

Valerie Sutton. 1999. SignWriting: On the occasion of its 25th anniversary November 1999. *Sign Language & Linguistics 2.2*.

Els van der Kooij. 2002. *Phonological categories in Sign Language of the Netherlands: The role of phonetic implementation and iconicity*. LOT Publications, Leiden, Netherlands.