Proceedings of the 10th Workshop on the Representation and Processing of Sign Languages (sign-lang@LREC 2022), pages 59–66
Language Resources and Evaluation Conference (LREC 2022), Marseille, 20-25 June 2022

Sign Language Phonetic Annotator-Analyzer: Open-Source Software for Form-Based Analysis of Sign Languages

Kathleen Currie Hall, Yurika Aonuki, Kaili Vesik, April Poy, and Nico Tolmie
University of British Columbia
2613 West Mall, Vancouver, BC V6T 1Z4 Canada
{kathleen.hall, yurika.aonuki, kaili.vesik}@ubc.ca
{aprilpoy, nhtolmie}@student.ubc.ca

Abstract

This paper provides an introduction to the Sign Language Phonetic Annotator-Analyzer (SLP-AA) software, a free and open-source tool currently under development, for facilitating detailed form-based transcription of signs. The software is designed to have a user-friendly interface that allows coders to transcribe a great deal of phonetic detail without being constrained to a particular phonetic annotation system or phonological framework. Here, we focus on the 'annotator' component of the software, outlining the functionality for transcribing movement, location, hand configuration, orientation, and contact, as well as the timing relations between them.

Keywords: sign language, phonetics, phonology, corpus, software

1. Background

Johnston (2010, 2014) made convincing arguments in favour of focusing resources on annotation rather than detailed phonetic transcription when it comes to building sign language corpora. Such advice makes sense when there is no widely agreed-upon system for transcription, video records are easily available, and priority needs to be placed on amassing larger datasets that are representative of individual sign languages (see also McEnery & Wilson, 2001; Gut & Voormann, 2014). At the same time, Johnston (2014: 157) also mentions that there can be a "value added" through transcription and that many teams have been interested in and invested in creating more detailed transcriptions. Indeed, our understanding of ways to represent the form of sign languages has been improved greatly by proposals for both phonological models (e.g., Padden & Perlmutter, 1987; Liddell & Johnson, 1989; Crasborn, 2001; Brentari, 1998; van der Kooij, 2002; Sandler & Lillo-Martin, 2006; Morgan, 2017) and phonetic transcription systems (e.g., Stokoe et al., 1965; Prillwitz et al., 1989; Hanke, 2004; Eccarius & Brentari, 2008; Johnson & Liddell, 2010, 2011a-b, 2012, 2021; Liddell & Johnson, 2019). Simultaneously, there has been increasing development of annotated corpora and databases of specific sign languages (e.g., the various SignBank corpora, including ASL-SignBank; ASL-LEX 2.0; the Corpus of Polish Sign Language (PJM); the German Sign Language corpus (DGS-Korpus); the database for Spanish Sign Language (LSE-Sign); the Swedish Sign Language Corpus; HandSpeak; Lifeprint.com).

As we have an increasing number of these 'broadbase' resources, it becomes more relevant to return to the idea of transcription in addition to annotation, to see whether there are new, more efficient ways to approach adding detailed phonetic transcriptions. Such transcriptions are necessary for doing more complete phonetic and phonological analyses of sign languages, as is illustrated by and discussed in Morgan (2017). Morgan's attention to detail in the documentation and description of the phonology of Kenyan Sign Language highlights both the previous lack of detail in many sign language descriptions and the new insights that can be gained by following more rigorous methods. As one example, Morgan (2017: §3.5-3.6) lays out a far more exacting method for deciding what 'counts' as a minimal pair in a signed language than many prior researchers have done and yet also points out that the oft-mentioned 'scarcity' of minimal pairs in sign languages is likely inaccurate. Instead, the way that minimal pairs are *distributed* in signed languages differs from that in spoken languages, such that a single parameter of contrast is simply likely to be the source of only a few pairs, rather than being re-used across many pairs as is common in spoken languages. As Morgan explains, however, "the process itself [of finding true minimal pairs] is painstaking and is probably impossible to do well without a digitized record of the formational content of signs that is easy to query on demand" (2017: 113).

It is in light of such observations that we are in the process of developing a new piece of software, Sign Language Phonetic Annotator-Analyzer (SLP-AA), a graphical user interface (GUI)-based software system for the form-based transcription and analysis of signs (see also Tkachman et al., 2016; Hall et al., 2017; and Lo & Hall, 2019 for descriptions of earlier versions). Three key components of this project that differentiate it from other similar endeavours are:

- the focus is on providing tools for facilitating form-based transcription and analysis, rather than being a more generalized linguistic annotation system or providing a database or corpus of already-coded forms;
- the system is intended to be relatively phonetic in nature and compatible with multiple phonological theories, enabling transcription of detailed variation across many sign languages, largely without relying on a specific notation system;
- the software and code are all completely free and open source (the Python code is available on GitHub), encouraging broad use and customization.

In this paper, we explain in more detail our vision and current implementation of the software in light of these features. Although our intention is for SLP-AA to be usable across all sign languages, most of our examples come from American Sign Language (ASL), as this is the sign language most in use in our local community and research setting.

2. Overview

There are two primary components to the SLP-AA software: the annotator portion, for which there is a version in development and which is the primary focus of this paper, and the analyzer portion, which is still pre-development. Both components make use of a relatively straightforward GUI that involves selecting from pre-set options written out in text in most cases, to maximize consistency and efficiency of coding.

2.1 Annotator

The annotator portion includes or will include modules for coding meta information and for transcribing sign type, movement, location, hand configuration, orientation, contact, and non-manual markers. Additionally, temporal relations among these modules can be encoded using a generic 'x-slot' framework (described in §3.8). There is also some ability for the software to auto-generate, auto-fill, and auto-link components. Details of all of these elements are described below in §3; see Figure 4 at the end of the paper for a visualization of how they all fit together.

2.2 Analyzer

The analyzer interface, intended to facilitate phonological searches and analyses, has not been developed yet, but is modelled on the *Phonological CorpusTools* software for spoken languages (Hall et al., 2019). It will allow users to search for any (combination of) specifications within the detailed coding, as well as to search for categories of signs that span specific combinations (e.g., searching for signs with three extended fingers, regardless of which fingers those are, or searching for multi-syllabic signs defined in a number of different ways). We also intend to have several pre-set search options (e.g., searching for dominance condition violations, cf. Battison (1978), or searching for typologically rare properties). In addition to the searches, other phonological analyses will be possible. For example, these might include finding minimal pairs, calculating the neighbourhood density of a given sign (Luce & Pisoni, 1998), calculating the functional load (Hockett, 1966) or informativity (Cohen Priva, 2015) of a particular phonological component, etc. Finally, we envision having a side-by-side comparison option, which will highlight form-based similarities or differences between signs selected by the user.

2.3 Integration

Currently, SLP-AA is stand-alone software that can be used to give detailed phonetic transcriptions of individual signs, one at a time. While these signs can come from any source, the software does not currently directly allow for the transcription of continuous signing. We are interested in seeing whether it can be integrated into other software that is equipped for this type of time-aligned transcription, e.g. iLex (Hanke & Storz, 2008), SignStream (Neidle et al., 2018), or ELAN (Crasborn & Sloetjes, 2008).

3. Annotator Functions

In this section, we briefly describe the specific functions available in the annotator component of the software. We preface this discussion, however, with the statement of an overarching general principle in our system design: our transcription system is intended to be phonetic in nature and as descriptive as possible. We try to cover all physically possible options and use physically based descriptions rather than relying on phonological categories that may be tied to specific languages or phonological theories. Our goal in doing so is to allow maximal flexibility and phonetic detail in the transcription of a variety of languages and dialects, registers, and phenomena (e.g., acquisition, poetic usage, slips of the hand). Of course, we recognize that *any* attempt to label components of signs is an act of categorization, and only the original production itself can be truly phonetic and maximally detailed. As others have pointed out, however, corpora are only useful for linguistic analysis insofar as they have been made machine-readable (e.g., Johnston, 2014; Crasborn, 2022), and our aim here is to help bridge the gap between broad annotations and original video recordings.

Another feature of our coding system is that it relies mostly on prose-based descriptions of phonetic characteristics (e.g., 'H1 [hand 1] and H2 [hand 2] maintain contact throughout the sign'). This bypasses the necessity for users to choose one of the multiple possible transcription systems listed in §1. As others have pointed out, no single system has yet achieved widespread acceptance (see discussion in, e.g., Hochgesang, 2014), and having to learn a notational system can itself be a barrier to both transcribing data in the first place and disseminating transcribed data (see discussion in Morgan, 2017: 60). In particular, using a text-based system means that the codings may be more accessible to a non-specialist audience—for example, a corpus or database that is coded using prose descriptions can be searched by teachers or learners of a sign language in order to find signs matching particular characteristics.

3.1 Metadata

Following guidelines set out for good data management practices (e.g., Crasborn, 2022; Kung, 2022; Mattern, 2022), SLP-AA includes built-in functions for managing metadata. For example, each signer, transcriber, source, and/or recording can be tagged with relevant demographic or reference information, and then each sign can be tagged with this source information. Each entry in the corpus is also automatically assigned a unique entry ID, according to information and formatting options selected by the user.

3.2 Sign-Level Information

Each entry has a variety of sign-level information that can be assigned to it. This includes the gloss and lemma / ID-gloss of the sign (see discussion in Johnston, 2008; Fenlon et al., 2015; Hochgesang et al., 2018), as well as the specific metadata for this token. Lemmas / ID-glosses have the potential to be imported from another source if desired, to maximize consistency. Labels can also be added here to flag additional information, e.g., compounds, finger-spelled items, or suboptimal video quality, with notes to explain the details. The list of these tags can also be modified by the user.

3.3 Sign Type

The sign type of a sign allows users to specify the overarching 'kind' of sign an entry is; at a basic level, this would cover types like one-handed (1H) and two-handed (2H). Within these larger types, additional phonetic detail can be added. For example, for 2H signs, users can specify the relation between the two hands in terms of shared or different handshapes, contact or lack thereof, and movement (see Figure 1). The sign type coding is a good example of how SLP-AA is more phonetically oriented than other systems. For instance, some existing corpora base their sign type distinctions on Battison's (1978) five sign types (e.g., ASL-Lex (Sehyr et al., 2021), LSE-Sign (Gutierrez-Sigut et al., 2016)). These sign types were developed with ASL in mind, and they combine logically separable features in ways that do not fully cover all possible combinations. For example, Battison's Type 1 signs are the only ones that involve both hands moving, and he further stipulates that in these signs, the movement must be identical or alternating. This precludes classifying signs like ASL RUN, which ASL-Lex 2.0 simply calls a "Symmetry Violation" sign. In SLP-AA, however, each component of the relation between the two hands is coded separately, so all signs can be described more exactly. Still, we recognize the utility of Battison's sign types, and plan to allow pre-defined searches in the Analyzer component that can find, e.g., all "Type 1" signs or all "Symmetry Violation" signs, regardless of the type of violation.

3.4 Movement Module

For each of the primary phonological parameters—movement, location, hand configuration, orientation, and, eventually, non-manual markers—SLP-AA works on a 'module' system. There is a pre-existing interface for each module, such as the movement module, and a user can invoke the module as many times as is relevant for achieving the desired level of phonetic detail, with each resulting specification tied to either H1 or H2. For example, in a sign like ASL DESTROY, there is both what is traditionally thought of as a 'path' movement, in which the two hands cross over each other and back along a horizontal axis, and what is traditionally thought of as a 'local' movement, in which the fingers of the two hands close and then open. In coding this sign in SLP-AA, then, a user would invoke two instances of the movement module per hand, one to represent each type of movement.[1]

This modularity allows a great deal of flexibility in terms of the variety of signs that can be coded as well as the internal conventions for coding. For example, a user could choose instead to invoke the movement module four times per hand in DESTROY, once for each direction of each movement (ipsi → contra, contra → ipsi for the 'path' movement and open → close, close → open for the 'local' movement), rather than twice with each movement being marked as bidirectional. While we recognize that this places a certain burden on the users of the system to be explicit and consistent about their own internal conventions, we also believe that this allows for the widest usability of the system across theoretical frameworks.

Similarly, we do not actually require users to classify movements into the traditional categories of 'path' / 'major' / 'primary' vs. 'local' / 'minor' / 'secondary' movements and instead have classifications for 1) "perceptual movements" (e.g., straight, circle, arc), "joint-specific movements" (e.g., twisting, closing), and "handshape changes" (e.g., fingerspelling). As Napoli et al. (2011: 19) point out, "the actual distinction between primary and secondary movement is not uncontroversial and is far from simple." For example, while wrist movements are typically considered local movements according to articulatory definitions of path and local movement categories (e.g., Brentari, 1998), some of them have been categorized as path movements (van der Kooij, 2002: 229; Sehyr et al., 2021: 269). Furthermore, forcing the choice between path and local movements at the level of phonetic transcription could mask empirical phenomena such as proximalization and distalization (Brentari, 1998), in which both path and local movements can be articulated by non-canonical joints. In response to these issues, our system allows any movement in which the hand or arm draws a perceptual shape in space to be classified as perceptual movement, with optional manual specifications of the exact (combination of) joints executing the movement under a separate "joint activity" section. Traditional local movements (relating to particular joints) defined in the literature are listed under the joint-specific movement section, with the associated joint activities optionally auto-filled (e.g., the joint-specific movement of "closing" can auto-fill to flexion of finger joints in the "joint activity" section).

In addition to the overall movement type and joint activity involved, each movement module allows for specification of the axis/axes, direction(s), and, if relevant, plane(s) of movement involved, along with characteristics like repetition and bidirectionality. Options in the movement module can be selected manually through a clickable menu system or typed in using keywords to allow for faster coding.

3.5 Location Module

As with movement, locations are specified in a modular system, such that users can invoke multiple instances of the location module to capture the position(s) of the hand(s) in space during the course of a sign in as much detail as the user wishes. At its most basic, this could be used to code positions of the hand at different timepoints in a sign (e.g., before and after a location-changing movement), though of course individual users could choose to code only one of these and leave the other unspecified / inferred from movement direction. Another use for multiple location modules, however, is to code the location of H1 both in space and with respect to H2. For example, in coding the ASL sign ROCKING CHAIR, a user could specify that H1 is in the location of the back of the index and middle fingers of H2, but also that both hands are simultaneously in neutral space.

While one could also use the modularity to encode what are traditionally called 'major' and 'minor' locations, we do allow for a single instance of the location module to be tagged with hierarchically nested locations. For example, in ASL EVERYDAY, a single location module could be used to transcribe that the sign is articulated on the head and at the cheek. Although users can specify their own custom set of locations, SLP-AA comes with a large set of pre-specified options, based on both body locations and signing space options.

[1] Note that the temporal relations between instances of a module will be covered in §3.8 and illustrated in Figure 3.

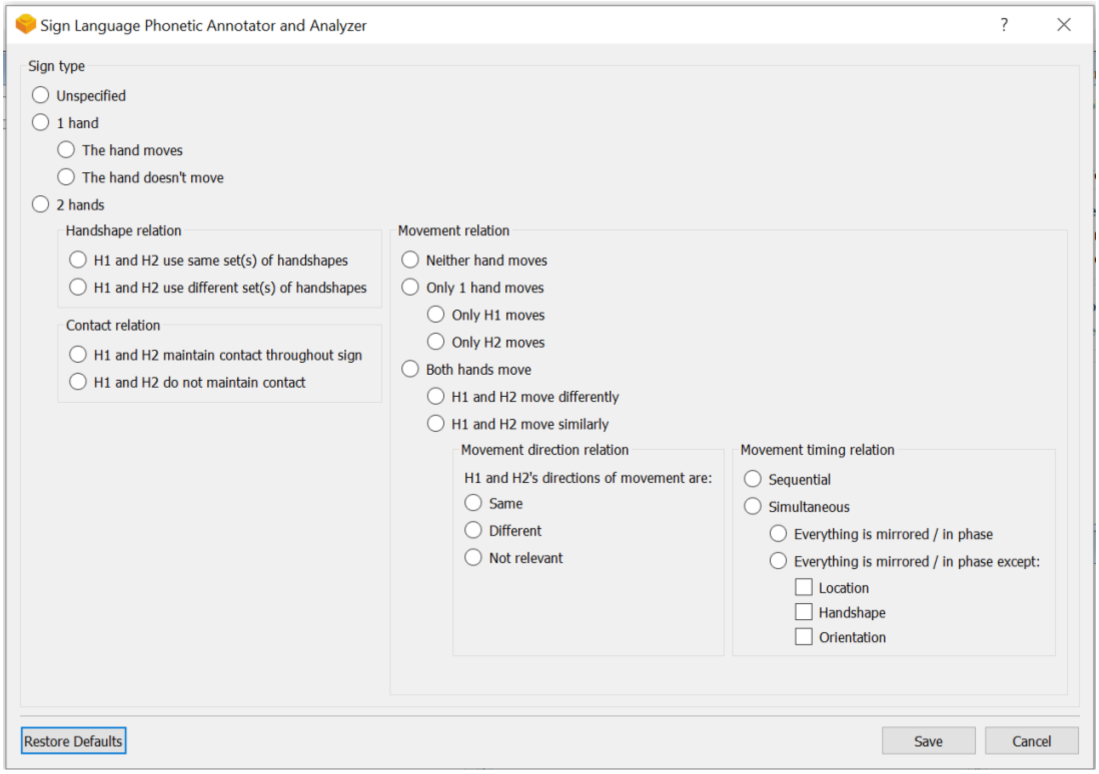

Figure 1: Screenshot of the "sign type" selector interface

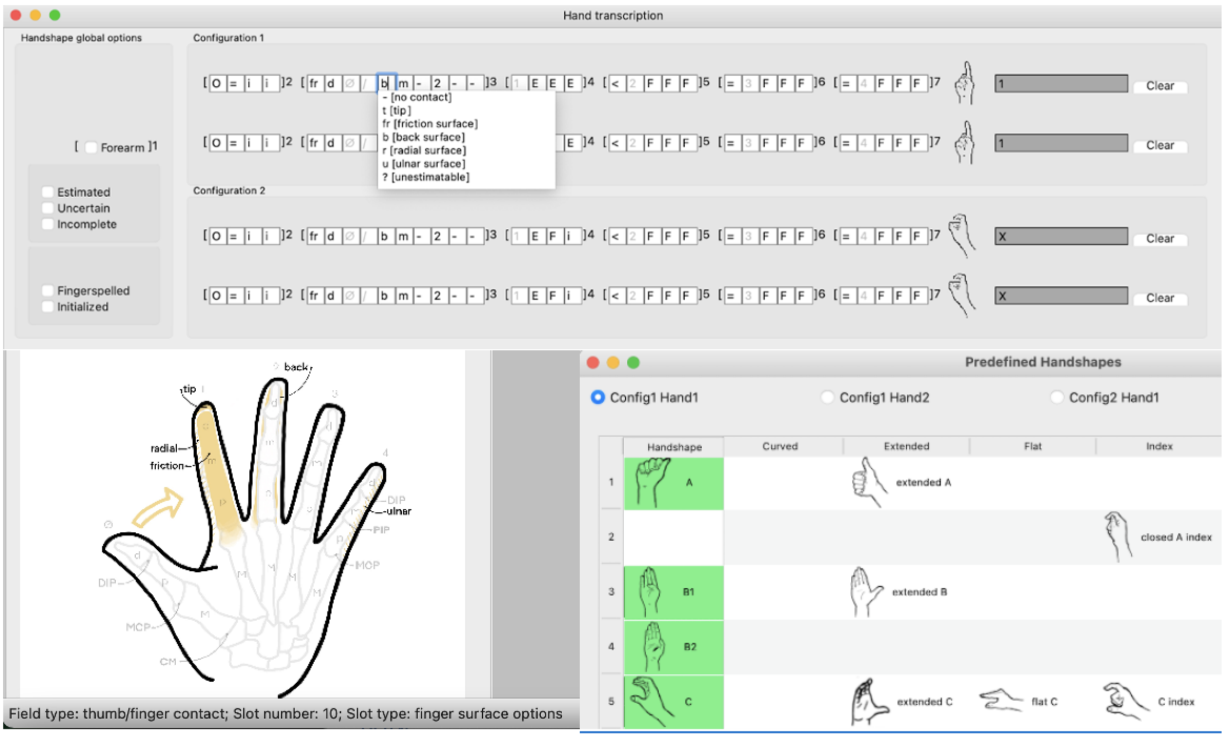

Figure 2: Elements from the hand configuration module for transcribing the handshapes in the ASL sign TEST. Top: Detailed phonetic coding following Johnson and Liddell (2011b, 2012), with the options for slot 10 ("thumb / finger contact") shown in a drop-down menu. Bottom left: Example drawing illustrating what type of information is expected in slot 10. Bottom right: Subset of the pre-defined handshapes selector, with phonologically unmarked base handshapes highlighted in green on the left and their variations in subsequent columns.

3.6 Hand Configuration Module

Each instance of the hand configuration module is used to transcribe a single configuration for one hand; a sign may consist of multiple sequential configurations. The transcription system for hand configuration is based closely on that proposed by Johnson and Liddell (2011b, 2012), and is also relatively thoroughly described elsewhere (Tkachman et al., 2016; Hall et al., 2017; Lo & Hall, 2019). There are 34 'slots' divided across seven 'fields' for transcribing a single hand configuration. Each slot is occupied by a letter, number, or symbol indicating characteristics like the degree of joint flexion, the degree of abduction / adduction from the adjacent finger, or points of contact within the hand configuration. Images and notes in the software remind the user what kind of information and choices are relevant for each slot (see Figure 2).

Recognizing that such a detailed coding can be tedious, we have also implemented a set of 'pre-defined' handshape options. For any given instance of the module, a pre-defined handshape can be selected (e.g., the ASL-based label 'extended-A,' associated with the following handshape:). Selecting the handshape then fills in the complete transcription, though any element may be modified to better represent a specific token if needed.

Finally, a user will be able to specify any given fingers within a hand configuration as 'selected.' This must be done manually, as it is a phonological rather than a phonetic characteristic; as van der Kooij and Crasborn (2016: 277) point out, the same phonetic handshape can have different 'selected fingers' based on the rest of the sign.

3.7 Additional Modules: Absolute Orientation, Handpart, and Contact

There are several additional characteristics that are coded with separate modules in SLP-AA. First, if absolute orientation is to be used (cf. Sandler, 1989), an orientation-specific module can be invoked. Each instance of the module involves specifying the absolute direction (e.g., up, distal, right) of both the palmar surface of the hand and the finger roots. Alternatively, if relative orientation is to be used (cf. Crasborn & van der Kooij, 1997), there is a handpart module that can be invoked to indicate which part of the hand is approaching a location or leading a movement, and the specified handpart can then be associated with a specific location or movement module.

Contact is also indicated with a separate module in SLP-AA. Each instance of the contact module is used to code the presence or absence of contact between an articulator and its target location, and, if relevant, the distance (e.g., close or far) and/or the relation (e.g., holding or continuous, cf. Friedman, 1976: 46-47) between them. Each instance of this module can be associated with a specific location module so that, e.g., in a phonetic description of a sign like ASL TIE, the brief initial contact between the two hands and the continuous contact between H1 and the torso can each be coded. The timing relations are coded through the linking of these contact modules to the x-slots, as will be described in §3.8.

3.8 Timing Relations

In order to fully represent a sign, it may also be necessary to specify the timing relations among the various instances of modules that have been coded. For example, in ASL DESTROY, the joint-specific finger closing-opening movements happen *simultaneously* with the *second* half of the perceptual straight movement of the hands (i.e., as each hand moves from contra to ipsi). In order to represent this timing, we make use of a generic 'x-slot' system. While the user can use these slots however they see fit, the system is built with the following choices in mind. In most cases, the basic timing structure of a sign is based on the single module that codes the most proximal movement. In ASL DESTROY, this 'foundational' movement would be the perceptual straight movement of the hands. Assuming this is treated as a bidirectional movement, the sign is associated with two 'x-slots.' Then, the bidirectional closing-opening movement is linked to the second x-slot.[2] This relation is shown in Figure 3.

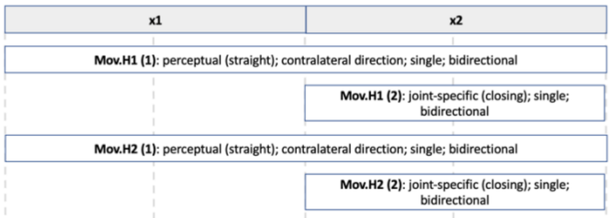

Figure 3: The movement components of the ASL sign DESTROY within an x-slot representation. Hand 1 ("H1") has two movements, with one being a bidirectional, unrepeated ("single"), straight perceptual movement that starts in a contralateral direction, and the second being a bidirectional joint-specific movement that starts with the fingers closing. The second movement starts halfway through the first movement, i.e., at the beginning of x2. Hand 2 ("H2") has two similar movements with the same timing.

In the full program, all of the modules are linked in this fashion, so that a fairly precise representation of the simultaneous and sequential components is attained, as is shown in Figure 4. This figure shows an example of the 'main' interface for SLP-AA, showing the summary of the transcription for a particular token of the ASL sign APPLE in the centre. Glosses of all signs in the corpus are shown on the left, and additional modules can be added and linked by clicking on the buttons on the right-hand side. Note that individual transcription modules can be linked as either intervals or timepoints to the x-slots (e.g., the movement, location, contact, and hand configuration modules are linked to intervals in Figure 4, while the orientations are linked to timepoints). The x-slots easily accommodate compounds or otherwise multisyllabic signs because any module, including the movement modules, can be sequentially ordered over any number of x-slots.

3.9 Auto-Generation/Filling/Linking

In many cases, the number of x-slots required for a given sign is predictable from the specified movements, as are the

[2] Recall that there are alternative ways of representing the bidirectionality in this sign (§3.4), which could lead to even more precise alignments of the closing-opening movement within the second x-slot.

number of instances of the location, hand configuration, and orientation modules, and how each instance would link to the x-slots. In these cases, SLP-AA will be able to generate x-slots and (un-filled) instances of the relevant modules and link the two to facilitate efficient and consistent transcription. For example, in ASL THROW-AWAY, the straight movement makes the hand travel in space, and the opening movement of the fingers changes the hand configuration. Once these movements are coded, auto-generation can begin. SLP-AA generates only one x-slot in this case, because the straight movement is unidirectional and unrepeated, and both the straight and opening movements are linked to the entirety of that single x-slot. Under the most detailed global transcription setting (see §3.10), two instances of the location module are auto-generated and auto-linked to the beginning and ending points of the x-slot, respectively, and the user is prompted to manually fill them in. Similarly, based on the existence of an opening movement, two instances of the hand configuration module are auto-generated and auto-linked to the beginning and ending points of the x-slot, respectively. Finally, given the lack of orientation-changing movement, the system auto-generates only one instance of the orientation module and links it to the entirety of the x-slot.

Moreover, for two-handed signs, some parameters of H2 may be predictable from those of H1 and the specification of the sign type. In those cases, some instances of modules for H2 can be auto-generated and auto-filled. For example, in two-handed signs with a shared handshape, like ASL MEET, once the user codes the hand configuration of H1 and links it to the x-slot, an identical instance of the hand configuration module is auto-generated for H2, which stretches for the same duration as its original H1 counterpart.

Any auto-generated information can be overwritten in the case of unusual sign types, but this feature helps streamline the process of doing such a detailed coding, while also facilitating inter-coder reliability.

3.10 Customization

To increase the flexibility of using the SLP-AA software with multiple different theoretical frameworks in mind, there are a number of customizations that can be done. Most menus will be modifiable so that individual menu items match those needed by a particular research team. Additionally, there are global settings that can be selected. These include whether to include timing relations (x-slots) at all; whether auto-generation/filling/linking should be done and if so how (e.g., should locations be coded as occurring before a location-changing movement, after, or both?); and how movement should be defined on the horizontal axis (i.e., in terms of ipsi / contra directions or in terms of right / left directions).

3.11 Additional features

There are a few other features of SLP-AA that are worth noting. First, individual transcriptions can be marked as being 'estimates,' 'uncertain,' or 'incomplete,' to capture and manage the realities of transcription. Additionally, each instance of each module can also be specified as being either a 'target' transcription or a 'token' transcription. In other words, coding may be done simultaneously for canonical productions and utterance forms. While a similar effect can also be achieved by linking individual signs to their lemma forms in a separately coded database, this feature allows for a fine-grained level of detail to be included. For instance, a particular utterance might be associated to a given lemma (e.g., it could be the ASL sign ADVERTISE), but then coded as having, say, a particular target handshape ("5") and phonetically realized as having the fingers more curved / relaxed than a canonical 5 handshape. Searches and analyses can be performed over targets, tokens, or both.

Finally, all transcriptions in the system will also be exportable to a plaintext format for use outside of the SLP-AA software itself, and the software is accompanied by detailed documentation.

While prompts within the software are currently all in English, the open-source nature of the project means that the prompts could be translated to another language (including e.g. having pop-up tool-tip videos in a sign language).

4. Conclusion

In sum, SLP-AA will be a tool for facilitating consistent and descriptively transparent transcriptions, and helps to set a precedent for the level of phonetic detail to be documented in signed languages. We believe that it is flexible enough to be used with a variety of languages, frameworks, and projects. As the software is still under development, we welcome feedback from other research teams as to what features and functionality would be most useful, and would like to work to integrate SLP-AA into other annotation and corpus-building workflows.

5. Acknowledgements

This research is supported in part by funding from the Social Sciences and Humanities Research Council. We are also grateful to numerous contributors and advisors on this project, including: Leanne Gallant and the Canadian Cultural Society of the Deaf, Ashley Chand, Paris Gappmayr, Julie Hochgesang, Cristina Lee, Roger Yu-Hsiang Lo, J. Scott Mackie, Hope Morgan, & Oksana Tkachman. All errors are our own. The handshape fonts are created by CSLDS, CUHK.

6. Bibliographical References

Battison, R. (1978). Lexical borrowing in American Sign Language. Silver Spring, MD: Linstok.

Brentari, D. (1998). A Prosodic Model of Sign Language Phonology. Cambridge, MA: MIT Press.

Cohen Priva, U. (2015). Informativity affects consonant duration and deletion rates. Laboratory Phonology, 6(2), pp. 243-278.

Crasborn, O. (2001). Phonetic implementation of phonological categories in Sign Language of the Netherlands. Doctoral dissertation. Universiteit Leiden.

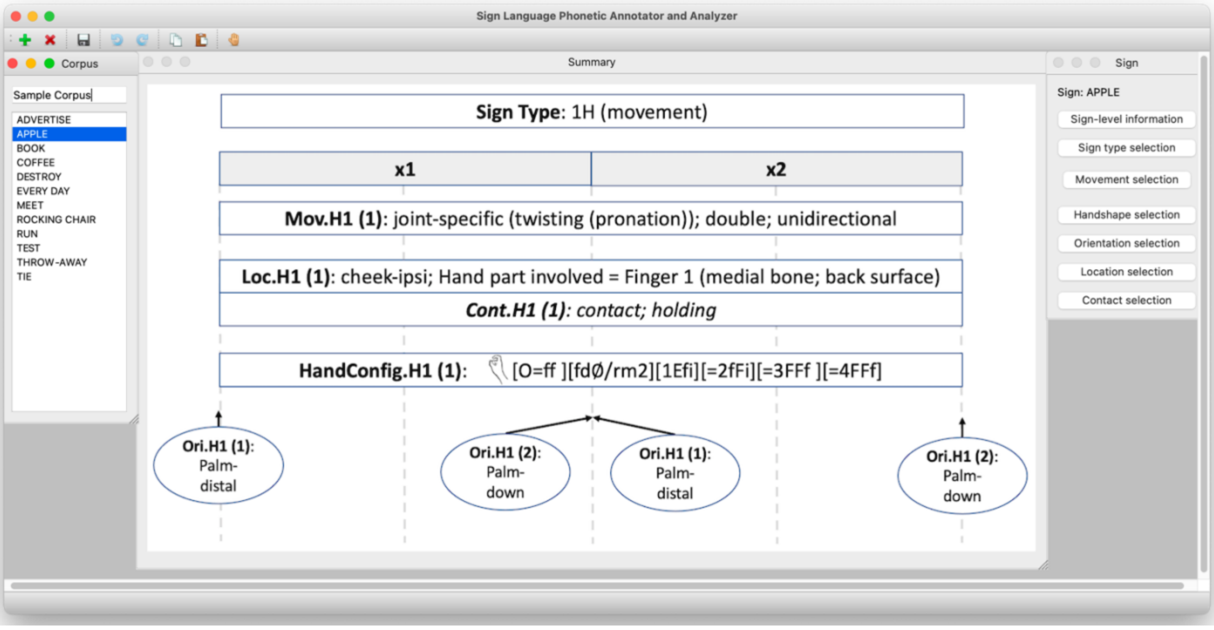

Figure 4: Mock-up of the SLP-AA interface, with a summary transcription for ASL APPLE. Note that this is a detailed phonetic transcription of this particular token of the sign; it is possible to link this to a lemma and/or do a less-detailed, more phonological transcription.

Crasborn, O. (2022). Managing data in sign language corpora. In A. L. Berez-Kroeker, B. McDonnell, E. Koller, L. B. Collister (Eds). The Open Handbook of Linguistic Data Management. Cambridge, MA pp. 463-470.

Crasborn, O., & van der Kooij, E. (1997). Relative orientation in sign language phonology. In J. Coerts, H. de Hoop (Eds.), Linguistics in the Netherlands. Amsterdam: John Benjamins, pp. 37-48.

Crasborn, O. & Sloetjes, H. (2008). Enhanced ELAN functionality for sign language corpora. In Proceedings of the 3rd LREC Workshop on the Representation and Processing of Sign Languages: Construction and exploitation of sign language corpora. Marrakesh: ELRA, pp. 39-43.

Eccarius, P. & Brentari, D. (2008). Handshape Coding Made Easier: A Theoreti-cally Based Notation for Phonological Transcription. Sign Language and Linguistics, 11(1), pp. 69-101.

Fenlon, J., Cormier, K. & Schembri, A. (2015). Building BSL SignBank: The lemma dilemma revisited. International Journal of Lexicography, 28, pp. 169-206.

Friedman, L. (1976). Phonology of a soundless language: Phonological structure of the American Sign Language. Doctoral dissertation. University of California, Berkeley.

Gut, U. & Voormann, H. (2014). Corpus Design. In J. Durand, U. Gut, G. Kristoffersen (Eds.), The Oxford Handbook of Corpus Phonology. Oxford: Oxford UP, pp. 13-26.

Gutiérrez-Sigut, E., Costello, B., Baus, C. & Carreiras, M. (2015). LSE-sign: A lexical database for Spanish Sign Language. Behavior Research Methods, 48, pp. 123-137.

Hall, K. C., Mackie, S., Fry, M. & Tkachman, O. (2017). SLPAnnotator: Tools for implementing Sign Language Phonetic Annotation. In Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017). Stockholm, Sweden: International Speech Communication Association, pp. 2083-2087.

Hall, K. C., Mackie, J. S. & Lo, R. Y-H. (2019). Phonological Corpus Tools: Software for doing phonological analysis on transcribed corpora. International Journal of Corpus Linguistics, 24(4), pp. 522-535.

Hanke, T. (2004). HamNoSys: Representing sign language data in language resources and language processing contexts. In Proceedings of the 4th LREC Workshop on the Representation and Processing of Sign Languages: From SignWriting to Image Processing. Paris: ELRA, pp. 1-6.

Hanke, T. & Storz, J. (2008). iLex: A database tool for integrating sign language corpus linguistics and sign language lexicography. In Proceedings of the 6th LREC Workshop on the Representation and Processing of Sign Languages: Construction and exploitation of sign language corpora. Paris: ELRA, pp. 64-67.

Hochgesang, J. A. (2014). Using Design Principles to Consider Representation of the Hand in Some Notation Systems. Sign Language Studies, 14(4), pp. 488-542.

Hochgesang, J., Crasborn, O. & Lillo-Martin, D. (2018). Building the ASL Signbank: Lemmatization Principles for ASL. In Proceedings of the 11th International Conference on Language Resources and Evaluation Workshop on the Representation and Processing of Sign Languages: Involving the Language Community. Paris: ELRA, pp. 69-74.

Hockett, C. F. (1966). The Quantification of functional load: A linguistic problem. U.S. Air Force Memorandum RM-5168-PR.

Johnson, R. E. & Liddell, S. K. (2010). Toward a phonetic representation of signs: sequentiality and contrast. Sign Language Studies, 11(2), pp. 271-74.

Johnson, R. E. & Liddell, S. K. (2011a). A segmental framework for representing signs phonetically. Sign Language Studies, 11(3), pp. 408-63.
Johnson, R. E. & Liddell, S. K. (2011b). Toward a phonetic representation: The fingers. Sign Language Studies, 12(1), pp. 4-45.
Johnson, R. E. & Liddell, S. K. (2012). Toward a phonetic representation of hand configuration: The thumb. Sign Language Studies, 12(2), pp. 316-33.
Johnson, R. E. & Liddell, S. K. (2012). Toward a phonetic description of hand placement on bearings. Sign Language Studies, 22(1), pp. 131-80.
Johnston, T. (2008). Corpus linguistics and signed languages: No lemmata, no corpus. In Proceedings of the 6th International Conference on Language Resources and Evaluation Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora. Paris: ELRA. pp. 82-87.
Johnston, T. (2010). From archive to corpus: Transcription and annotation in the creation of signed language corpora. International Journal of Corpus Linguistics, 15(1), pp. 106–131.
Johnston, T. (2014). The reluctant oracle: Using strategic annotations to add value to, and extract value from, a signed language corpus. Corpora, 9(2), pp. 155-189.
Kooij, E. van der (2002). Phonological categories in Sign Language of the Netherlands: The role of phonetic implementation and iconicity. Doctoral dissertation. Netherlands Graduate School of Linguistics.
Kooij, E. van der and Crasborn, O. (2016). Phonology. In A. Baker, B. van den Bogaerde, R. Pfau, T. Schermer (Eds.), The linguistics of sign languages: An introduction. Amsterdam: John Benjamins, pp. 251-278.
Kung, S. S. (2022). Developing a data management plan. In A. L. Berez-Kroeker, B. McDonnell, E. Koller, L. B. Collister (Eds.), The Open Handbook of Linguistic Data Management. Cambridge, MA: MIT Press, pp.101-115.
Liddell, S. K. & Johnson, R. E. (1989). American Sign Language: The phonological base, Sign Language Studies, 64, pp. 195-277.
Liddell, S. K. & Johnson, R. E. (2019). Sign Language Articulators on Phonetic Bearings. Sign Language Studies, 20(1), pp. 132-172.
Lo, R. Y.-H. & Hall, K. C. (2019). SLP-AA: Tools for Sign Language Phonetic and Phonological Research. In Proceedings of the 20th Annual Conference of the International Speech Communication Association (INTERSPEECH 2019): Show & Tell Contribution, Graz, Austria: International Speech Communication Association, pp. 3679-3680.
Luce, P. A. & Pisoni, D. B. (1998). Recognizing spoken words: The neighborhood activation model. Ear and Hearing, 19, pp. 1-36.
Mattern, E. (2022). The Linguistic Data Life Cycle, Sustainability of Data, and Principles of Solid Data Management. In A. L. Berez-Kroeker, B. McDonnell, E. Koller, L. B. Collister (Eds.), The Open Handbook of Linguistic Data Management. Cambridge, MA: MIT Press, pp. 61-71.
Morgan, H. E. (2017). The phonology of Kenyan Sign Language (southwestern dialect). Doctoral dissertation. University of California, San Diego.
McEnery, A. M. & Wilson, A. (2001). Corpus Linguistics. Edinburgh: Edinburgh UP.
Napoli, D. J., Mai, M. & Gaw, N. (2011). Primary movement in sign languages: A study of six languages. Washington D.C.: Gallaudet UP.
Neidle, C., Opoku, A., Dimitriadis, G. & Metaxas, D. (2018). New shared & interconnected ASL resources: SignStream® 3 software; DAI 2 for web access to linguistically annotated video corpora; and a sign bank. In Proceedings of the 11th LREC Workshop on the Representation and Processing of Sign Languages: Involving the Language Community, Paris: ELRA.
Prillwitz, S., Leven, R., Zienert, H., Hanke, T. & Henning, J. (1989). HamNoSys. Version 2.0. International Studies on Sign Language and Communication of the Deaf, 7, pp. 225-231.
Padden, C. A. & Perlmutter, D. M. (1987). American Sign Language and the architecture of phonological theory. Natural Language & Linguistic Theory, 5(3), pp. 335-375.
Sandler, W. (1989). Phonological representation of the sign: Linearity and nonlinearity in American Sign Language (Vol. 32). Berlin: Walter de Gruyter.
Sandler, W. & Lillo-Martin, D. (2006). Sign language and linguistic universals. Cambridge: Cambridge UP.
Sehyr, Z. S., Caselli, N., Cohen-Goldberg, A. M. & Emmorey, K. (2021). The ASL-LEX 2.0 Project: A database of lexical and phonological properties for 2,723 signs in American Sign Language. The Journal of Deaf Studies and Deaf Education, 26(2), pp. 263-277.
Stokoe, W. C., Casterline, D. C. & Croneberg, C. G. (1965). A Dictionary of American Sign Language on Linguistic Principles. Silver Spring, MD: Linstok.
Tkachman, O., Hall, K. C., Xavier A. & Gick, B. (2016). Sign Language Phonetic Annotation meets Phonological CorpusTools: Towards a Sign Language Toolset for Phonetic Notation and Phonological Analysis. In Proceedings of the 2015 Annual Meeting on Phonology, Washington, DC: Linguistic Society of America, pp.1-9.

7. Language Resource References

ASL-LEX 2.0. (2020). Available at https://asl-lex.org/index.html
ASL-SignBank (2021). Available at https://aslsignbank.haskins.yale.edu/
Corpus of Polish Sign Language (PJM). (2013). Section for Sign Linguistics, University of Warsaw. Available at https://ses.library.usyd.edu.au/handle/2123/9841
DGS-Korpus. (2009). The Institute for German Sign Language and Communication of the Deaf at Hamburg University. Available at https://www.sign-lang.uni-hamburg.de/dgs-korpus/index.php/welcome.html
HandSpeak. (1995). American Sign Language dictionary. Available at www.handspeak.com
Lifeprint.com: ASL University. (n.d.). ASLU: English to ASL Dictionary. Available at https://www.lifeprint.com/
LSE-Sign. (2015). BCBL. Available on Portal LSE at http://lse-sign.bcbl.eu/web-busqueda/
SignBank. (n.d.). SignBank: Home for the various spinoffs of the Auslan Signbank. Available at https://github.com/Signbank
Swedish Sign Language Corpus (2020). Department of Sign Language, Department of Linguistics, Stockholm University. Available at teckenspraakskorpus.su.se