

[†]*This version of the article has been accepted for publication at “Language Resources and Evaluation” after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: <https://doi.org/10.1007/s10579-023-09694-9>. Use of this Accepted Version is subject to the publisher’s Accepted Manuscript terms of use <https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms>.*

Building the VisSE Corpus of Spanish SignWriting

Antonio F. G. Sevilla^{1,2*}, Alberto Díaz Esteban^{1,3} and José María Lahoz-Bengoechea²

¹Department of Software Engineering and Artificial Intelligence.

²Department of Spanish Linguistics and Literary Theory.

³Knowledge Engineering Institute.

Universidad Complutense de Madrid.

*Corresponding author(s). E-mail(s): afgs@ucm.es;

Contributing authors: albertodiaz@fdi.ucm.es; jmlahoz@ucm.es;

Abstract

SignWriting is a system for transcribing sign languages, using iconic depictions of the hands and other body parts, as well as exploiting the possibilities of the page as a two dimensional medium to capture the three-dimensional nature of signs. This goes beyond the usual line-oriented nature of oral writing systems, and thus requires a different approach to its processing. In this article we present a corpus of handwritten SignWriting, a collection of images which transcribe signs from Spanish Sign Language. We explain the annotation schema we have devised, and the decisions which have been necessary to deal with the challenges that both sign language and SignWriting present. These challenges include the transformational nature of symbols in SignWriting, which can rotate and otherwise transform to convey meaning, as well as how to properly codify location, a fundamental part of SignWriting which is completely different to oral writing systems. The data in the corpus is fully annotated, and can serve as a tool for computational training and evaluation of algorithms, as well as provide a window into the nature of SignWriting and the distribution of its features across a real vocabulary. The corpus is freely available online at <https://zenodo.org/record/6337885>.

Keywords: Sign Language, SignWriting, Corpus, Writing Systems, Graphical Languages

Acknowledgments

The VisSE corpus was created thanks to funding from the project “Visualizando la SignoEscritura” (VisSE, Visualizing SignWriting¹), reference number PR2014_19/01, funded by Indra and Fundación Universia, and development continues thanks to the project “Signario de LSE: Diccionario paramétrico de la lengua de signos española” (SSL Signary: A parametric dictionary of Spanish Sign Language²), reference number IN[21]_HMS_LIN_0070, supported by a 2021 Leonardo Grant for Researchers and Cultural Creators from the BBVA Foundation. The BBVA Foundation accepts no responsibility for the opinions, statements and contents included in the project and/or the results thereof, which are entirely the responsibility of the authors.

1 Introduction

Modern linguistics rely ever increasingly on digital data, source instances of language along with annotations of their origin, meaning, or features. These assets are often organized into datasets or corpora, collections of annotated linguistic data sharing a theme or object of study. The creation and sharing of datasets can help immensely in the research of a certain subject, allowing empirical investigation as well as providing a shared substrate on which to discuss and compare theories.

As an object of increasing linguistic scrutiny, sign languages have also seen the construction of diverse corpora in recent times, covering some of the more than a hundred different sign languages in use in the world. Sign languages, however, present unique challenges due to their viso-gestual nature and, especially for linguists, their lack of a standard and widespread form of writing.

Often, sign languages are recorded using video, and the meaning is annotated using glosses from the oral language in the same geographic region. This is, however, not a proper transcription, since sign languages are natural languages with a grammar and lexicon of their own, and, in order to properly capture them, a native system is needed.

There are a few existing proposals for transcribing sign languages into a written form. Unique among them, SignWriting (Sutton & Frost, 2008) transposes the three-dimensional nature of signs into a bi-dimensional arrangement of symbols, as can be seen in Figure 1. Different iconic symbols are used to represent the head, hands and other body parts, and their location and movement is recorded in an abstract and systematic manner.

However, the graphical nature of SignWriting means it is very different from the usual writing systems for oral languages, making it harder to process with existing tools and standards. Moreover, while there exist some computational representations, very often SignWriting is shared in the form of images,

¹<https://www.ucm.es/visse>

²<https://www.ucm.es/signariolse>



Fig. 1: SignWriting transcription of the Spanish Sign Language sign for “lie”. A video can be seen online at SpreadTheSign: <https://www.spreadthesign.com/es.es/word/22502/mentira/0/?q=mentira>

which do not require special fonts and software installed to be viewed, but are impossible to process as text.

Therefore, to be able to linguistically process SignWriting in its image form, tools and standards are required. If these are to be developed empirically, source data are also needed. A few collections of sign language transcribed using SignWriting exist, but are not research oriented and deal only with the digital representation. Additionally, SignWriting can also be handwritten, and there are no research corpora of handwritten SignWriting that we are aware of.

In this article, we present the VisSE corpus of Spanish SignWriting, a collection of handwritten SignWriting instances representing signs of Spanish Sign language. The instances have been graphically annotated, for which an extensive schema has been developed, recording both the lexical meaning of the different symbols involved as well as their spatial information, a fundamental part of SignWriting.

This corpus can be used to extract information on SignWriting, for research on the processing of graphical languages, for empirical study of the features of sign languages, for the training and evaluation of machine learning methods, or for other research purposes which have not occurred to us. We have used it in our previous and ongoing research, and therefore, believing it may be of use to the research community at large, we have freely released it online (Sevilla, Lahoz-Bengoechea, & Díaz Esteban, 2022). We will continue to expand and improve it, and this article relates its current state, how it has been built and the annotation schema used.

In section 2, related corpora and tools are discussed, as well as systems for computationally storing SignWriting. Section 3 explains the different objects in the corpus and how they have been annotated, while section 4 is centered on the concrete details and computational aspects of its construction. Section 5 gives an overview of the data and some statistics, and in sections 6 and 7 a few conclusions are drawn and future work is described.

Due to its complexity, explaining SignWriting is out of the scope of this article, but enough detail will be given to allow the reader to follow the discussion.

For more information, interested readers can see the extensive documentation available online at <https://signwriting.org>.

2 Related Work

Existing sign language corpora or datasets are usually comprised of videos of utterances, whether isolated signs or phrases. Many are not intended for research, but rather for regular use, and are structured as dictionaries.

Spread the Sign³ (Hilzensauer & Krammer, 2015) and DILSE⁴ (Moreno, 2012) are two of these dictionaries containing the Sign Language translations (as video) of words in one or many oral languages. These videos are provided without any phonetic annotation, though DILSE is interesting in that, additionally to the video, it includes static photographs of signers where movement is annotated using superposed arrows, and when needed, different instants of the sign are recorded as consecutive photographs. We see this as an approach halfway to SignWriting, which follows similar principles but in a more abstract and standardized manner.

Other datasets, even if often also intended to be usable as dictionaries, are structured to allow research by examining the data or searching by features instead of just by meaning. LSE-Sign (Gutierrez-Sigut, Costello, Baus, & Carreiras, 2016) is a web tool that contains 2400 signs from Spanish Sign Language, annotated with linguistic features to enable searching for concrete characteristics of signs. The signs are stored as videos and glosses, but the annotation is rich, with entries for hand shape and orientation, movement shape, and other features. Other such corpora of sign language videos exist, such as those for Australian Sign Language, British Sign Language, and others, based on the Signbank software (Cassidy et al., 2018).

A common necessity of sign language corpora is a relevant and meaningful annotation of the signs depicted, since video by itself is not computationally processable. To this end, phonological or phonetic transcriptions of signs can be used, but there is not a universally accepted way to represent the movements and gestures of sign language neither formally nor computationally. The forefather of sign language linguistics, William Stokoe, proposed a linear writing system consisting of abstract symbols to encode the different parameters of the language (Stokoe, 1960). The Hamburg Notation System (Hanke, 2004) uses a similar approach but with a different set of symbols, while Ángel Herrero Blanco, in his study of Spanish Sign Language, developed another featured writing system but using characters from the roman alphabet (Herrero Blanco, 2003).

A different approach to sign language transcription can be found in SignWriting (Sutton & Frost, 2008), a featural writing system (Galea, 2014, pp. 76-77) where abstract symbols are used for representing linguistic features. Their shape is chosen as iconic as possible, helping the reader and

³<https://www.spreadthesign.com>

⁴<https://fundacioncnse-dilse.org>

writer remember the actual physical articulator the symbol represents. The main difference that SignWriting introduces is that symbols are also arranged iconically, instead of in a linear fashion. The bi-dimensional page is used to represent three-dimensional sign space, and symbols are set on it according to their actual location in the realization of the sign. This enables the writer to capture the spatial richness of sign language almost directly, but is a radical departure from the main paradigm of oral writing systems.

One of the problems this presents is that the common computational representation of oral writing systems, as sequences of individual and mostly independent symbols, is insufficient for representing SignWriting. Nonetheless, there is ongoing effort to solve this problem, and some of SignWriting can be represented using Unicode.

Unicode is a “universal character encoding standard for written characters and text” (The Unicode Consortium, 2021) which assigns a number to each possible character in use in a documented human language, so that text can be computationally stored as a sequence of bytes. It includes character points and combinations for many of the symbols in SignWriting, and The International SignWriting Alphabet (Sutton & Slevinski, 2010) provides fonts which, when installed in the user’s computer, allow for the proper display of the symbols.

However, “the spatial arrangement of the symbols (...) constitutes a higher-level protocol beyond the scope of the Unicode Standard” (The Unicode Consortium, 2021, p. 831), meaning Unicode is not enough to fully codify SignWriting. To solve this problem, computational solutions such as Formal SignWriting or SignWriting Markup Language (da Rocha Costa & Dimuro, 2002; Slevinski, 2016; Verdú Perez, Pelayo García-Bustelo, Martínez Sánchez, González Crespo, et al., 2017) often store positional information as numerical coordinates alongside the Unicode bytes, indicating where to place them in bi-dimensional symbol space. Nonetheless, these systems are intended mainly for creation and display of SignWriting, not for its linguistic processing, and thus lack many annotations needed for the fully automatic understanding of the transcriptions.

Compared to video-based corpora of sign language, there are not many databases of sign language which use SignWriting as their representation form. SignPuddle⁵ is a dictionary and database of sign language which stores SignWriting using Unicode and storing symbol positions as coordinate pairs. It is a multilingual dictionary, containing entries for many different sign languages across the world, including Spanish Sign Language. The web interface allows searching by word, symbol or searching full signs, and data can be exported for offline processing. However, since it uses the computational systems mentioned before, it does not contain the “higher-level protocol” information identified by the Unicode Standard and needed for decoding the complete meaning of transcriptions.

⁵<https://www.signbank.org/signpuddle2.0/>

3 Annotation Schema

The VisSE corpus is a collection of handwritten SignWriting instances (images) representing signs or parts of signs from Spanish Sign Language. These instances are annotated both graphically, by demarcating the relevant regions of the image where meaning is codified, and more conventionally using textual tags to codify the meaning and attributes of the different symbols.

3.1 Logograms

Each SignWriting instance is called a “logogram” (Slevinski, 2016), since it represents units of meaning which are either words or word-like (not necessarily full signs⁶). Although the term logogram is commonly used to emphasize the non-phonetic nature of characters, it is worth noting that there are instances where sub-units may contain phonetic information (Liu, Vermeyleen, Wisniewski, & Brysbaert, 2020). In the case of SignWriting, all of the sub-units are phonetic, conveying the gestures (understood in the broad articulatory sense) necessary to articulate each sign. We call each of these sub-units of writing “graphemes”.

Graphemes codify most of the phonetic content of SignWriting, and so they require the most complex annotation, consisting of not a single label but a set of features for each grapheme. The list of graphemes present, along with their feature set, is the core of each logogram’s annotation.

However, the meaning of each grapheme is not only determined by its graphical properties, but also by its position relative to the other graphemes in the logogram. It is only after contemplating both each grapheme’s features and their holistic arrangement in the page that can the sign transcribed by a logogram be understood.



Therefore, logogram annotation consists of a list of graphemes with their relative locations, each annotated with their own independent feature set. The locations are annotated as ‘bounding boxes’, the geometrical regions within the logogram where each of the graphemes can be found. An example of this logogram annotation can be seen in Figure 2.

3.2 Graphemes





Unlike other writing systems, graphemes in SignWriting are not a one-to-one mapping from a shape or picture to a phoneme or phonemic feature, but rather encode complex meaning in their graphic form and visual properties. They have internal structure, both in their graphical properties (strokes, fill) and in their presentation (rotation, reflection). Each of these properties are encoded in a set of tags, a mapping from feature names to feature values that stores their independent meaning.

Some of this meaning is lexical, in that the shape of the grapheme must be looked up in a dictionary to understand what it represents. This is mostly the

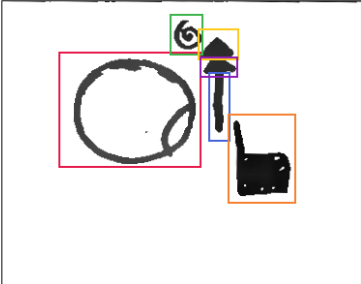
⁶Whether these are syllables, morphemes, or maybe something else is a question of linguistic research.

VisSE » logograms/A1_2 » 112  

Metadata gloss:

Annotation Bounding Boxes:















	CLASS	SHAPE	VAR	ROT	REF	
	HEAD	cheekr	VAR	ROT	REF	
	DIAC	rub	VAR	ROT	REF	
	ARRO	b	VAR	N	REF	
	STEM	s	VAR	N	REF	
	HAND	l	b	N	y	
	ARRO	b	VAR	N	REF	

Fig. 2: Visual annotation of the sign “lie/to lie”. Superposed over the logogram image, the bounding boxes of the different graphemes are drawn. The color codes serve to match each region to their grapheme, represented to the right as a table of feature names and values.

shape and outline of the grapheme’s form, which while often iconic and thus intuitive for humans to remember, is in the end conventional and abstract. The rest of the grapheme’s meaning is morphological, in that it is derived from the graphical transformation of the grapheme’s form. For example, the grapheme may be filled with different patterns of black and white, or drawn rotated around its center.

To properly annotate this meaning, a schema of five different features or tags is used. The first two, the ‘CLASS’ and ‘SHAPE’, codify the lexical part of the grapheme’s meaning, while the rest, namely ‘VAR’, ‘ROT’ and ‘REF’, encode the graphical transformations. As we will see, not all graphemes can be transformed in the same way or at all, so not all graphemes use the same set of features. Which features need to be used is determined by the first tag, the ‘CLASS’, which separates graphemes into different groups according to their visual characteristics (mostly size and variability) as well as their transformation possibilities. We have settled in six classes for our corpus: ‘HEAD’, ‘DIAC’, ‘HAND’, ‘ARRO’, ‘STEM’ and ‘ARC’.

Once the ‘CLASS’ is determined, the ‘SHAPE’ completes the lexical meaning by refining the classification down to what a user of SignWriting might actually identify as a ‘character’. For example, a concrete hand shape, a symbol for a head or a contact mark. The rationale for the grouping and annotation of



Fig. 3: Three HEAD graphemes, representing different parts of the head as place of articulation. The first is SHAPe=chin; the second SHAPe=mouth, and the third SHAPe=smile. This third grapheme represents not only the head as a body part, but also the “smiling” facial expression, which can be semantically relevant in sign languages.

the different classes, as well as any further tags needed for any of them, are covered in the following.

‘HEAD’ and ‘DIAC’ have the simplest annotation, so are explained together in 3.2.1, while ‘HAND’ graphemes are the most complex and section 3.2.2 is fully dedicated to them. ‘ARRO’, ‘STEM’ and ‘ARC’ are grapheme classes used to annotate the significant components of movement markers, and so are described together in section 3.2.3.

3.2.1 Invariant graphemes

Two first classes of graphemes are ‘HEAD’ and ‘DIAC’. These groups of graphemes do not transform, so are always presented with the same picture, and the ‘SHAPe’ feature is enough to discern their independent meaning. They are separated into two classes mostly due to their graphical characteristics. ‘HEAD’ graphemes are big and sparse, while ‘DIAC’ graphemes are small and compact. Nonetheless, they also have different characteristics in how they contribute to the meaning of a logogram.

‘HEAD’ graphemes, by depicting the head or some of its parts (eyes, nose, etc), establish a place of articulation, and the location of other graphemes is decided relative to them. Additionally, iconic representations of the eyes, mouth, and other elements can be used to transcribe facial expressions, an important non-manual parameter of sign languages. Some examples of ‘HEAD’ graphemes can be seen in Figure 3.

‘DIAC’ graphemes act more like diacritics, modifying the meaning of nearby graphemes in some predictable way—hence the name, though no profound thinking has been given to whether they actually count as traditional diacritics. Some examples of ‘DIAC’ graphemes are dynamic marks, which establish the coordination of the hands, or the velocity of the signing, thus affecting the whole logogram and having a mostly arbitrary place within it. Other marks, such as internal movements of the hand, or contact markers, must be placed nearby the graphemes which they modify, though there are no hard rules as to where exactly.

In Figure 4 some example ‘DIAC’ samples can be seen, and their use in combination with other graphemes in the logogram.



Fig. 4: In 4a, some DIAC graphemes from the corpus are shown. Clockwise from top left, the SHAPES are: `wiggle`, `flex_hook`, `touch` and `brush`. In 4b, the logogram for the sign ‘left-handed’ is shown. The `touch` grapheme marks that the hand should be in contact with the head, while the top two `flex_hook` graphemes specify that the little finger must bend twice in the “hooking” manner (they are above a `HAND` grapheme, explained in the following).

3.2.2 Hand graphemes

Hands are the most prominent articulators of Sign Language, and have many degrees of freedom and articulatory possibilities. Different authors assign different features to signs, but some commonalities can be found. The “hand shape” or configuration is a feature that accounts for the articulatory possibilities of the fingers, i.e. how are they bent to produce a unitary and meaningful shape; orientation is a feature specifying the rotation of the hands as 3D objects in sign space.

In SignWriting, each different hand shape is assigned a picture, a combination of strokes that iconically represents the hand and the fingers. This basic picture can have different filling patterns of black and white, and can also be rotated or reflected. These different attributes are annotated separately in our corpus, to represent their combinatory possibilities. Hands are grouped under the `CLASS=HAND`, and present the most complex annotation, requiring all the features available in the corpus for their annotation.

The outline of the character, which SignWriting uses to represent the hand shape, is annotated in the tag ‘`SHAPE`’. This roughly corresponds to the sign language parameter of hand configuration, and therefore a suitable linguistic notation system can be used to transcribe it. Some different notation systems exist for hand shapes, varying in their applicability to different sign languages and their ease of use. We use our own notation system, somewhat similar to that of [Eccarius and Brentari \(2008\)](#), but specific for Spanish Sign Language. Some examples of hand shapes can be seen in table 1.

These basic forms for hand graphemes can suffer a number of graphical alterations in SignWriting, used to transcribe the hand as a three dimensional object in the flat page. The hand grapheme can be filled with three different patterns: full white, full black or half and half. Then, the grapheme can be rotated using a set of eight possible different angles. These two transformations

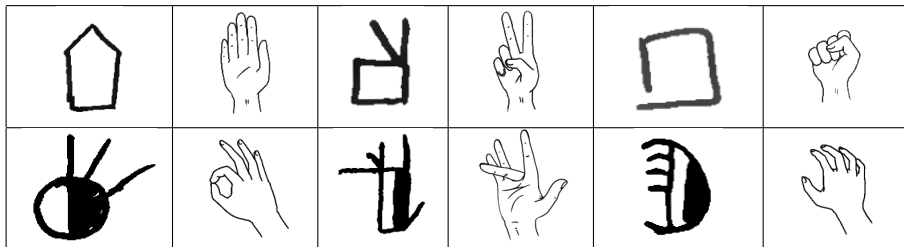


Table 1: Some SignWriting hand grapheme samples from the corpus along with the hand shape they represent.

VAR	white	white	white	half	half	black
ROT	North	South	NorthEast	North	North	North
REF	no	no	yes	no	yes	yes

Table 2: Hand graphemes and their transformational annotation. The label values have been expanded from the actual abbreviations used in the corpus to be more informative.

encode the orientation of the hand, and are annotated in our corpus in the ‘VAR’ and ‘ROT’ features.

A final graphical transformation allows the hand graphemes to be reflected across their longitudinal axis, turning them into their mirror image. This transformation is used by SignWriting to better iconically depict the hand as would be seen by the signer, representing the fingers in their correct position across the hand and also being useful to represent left hands (which are mirror images of the right hand). This reflection is coded in our corpus in the ‘REF’ feature, using a *yes/no* value.

To decide whether a hand grapheme is reflected or not, however, is not as straightforward as it may seem. Without the wider context, it is impossible to predict whether a grapheme is an unreflected left hand or a reflected right one. For example, in table 2, the fourth grapheme could be either a right or a left hand, we only know that the palm is looking left. Conversely, the fifth grapheme does not tell us, without wider context, whether it is a left or right hand, but only that the palm is oriented to the right. This wider context might be two hands, side by side, or a nearby body part, allowing humans to deduce which hand is depicted. This is inherently ambiguous, however, and requires understanding of the human body and sign language phonotactics.

We choose to optimize graphical stability, meaning graphemes which are similar in features should also be similar graphically. To this effect we choose the ‘half’ variant as the guide for whether graphemes are reflected or not

(choosing as non-reflected the one with the palm to the left), and base the ‘REF’ feature for white or black variants on their graphical similarity to the half one.

More examples of hand grapheme alterations can also be seen in table 2. For the complete enumeration and explanation of tag values, please refer to the annotation guide that can be found in the corpus (in English and Spanish, available inside the corpus distribution file or directly at <https://zenodo.org/record/6337885>).

3.2.3 Movement marks

Hand movements are an integral part of sign language, and therefore a substantial part of SignWriting. They are codified with paths and arrows depicting the 3D movements of the hands in the page, describing the shape of the movement by drawing it in an intuitive way. To properly encode 3D space in 2D writing, they use graphical attributes to distinguish between planes of movement, similar to how ‘HAND’ graphemes have variations to represent different palm orientations. Some examples of movements in the corpus can be seen in Figure 5.

What constitutes a grapheme in movement markers, however, is not an easy decision. As with other elements of SignWriting, movement symbols use both symbolic graphical properties (strokes, filling, shapes) as well as location in the page to record information. The shape of sign language movement is directly and iconically converted into bi-dimensional trajectories, with a few graphical attributes to bridge the gap to the extra dimension.

One approach would be to understand the full trajectory and associated symbols of the movement as a unit. This is the approach used by the Unicode standard and the International SignWriting Alphabet, even if some of the features, such as rotation, are encoded as different codepoints forming combining characters. Indeed, there are tens of thousands of glyphs in the International SignWriting Alphabet fonts to try and account for as many possible movements as possible, a small sample of which can be seen in Figure 6. Encoding movements holistically is, however, problematic for annotation due to the sheer number of them, and there is arguably a loss of information incurred when transforming a visual, meaningful, spatial representation into an index in a table (the lost information must be looked up in the table, instead of being directly available). Additionally, dealing with handwritten SignWriting renders this approach even more difficult, since it is not guaranteed that transcriptions will use only the abstractions that are collected in the Unicode representation.

Instead, we have opted to characterize the different elements of movement markers as individual graphemes. From the lexical annotation point of view, this makes sense because there are repeated elements, with identifiable semantics of their own, which can be used in different context. While these elements could be transcribed as different tags for the same grapheme, as is done for hands, their independent spatial characteristics make a subdividing approach more useful.

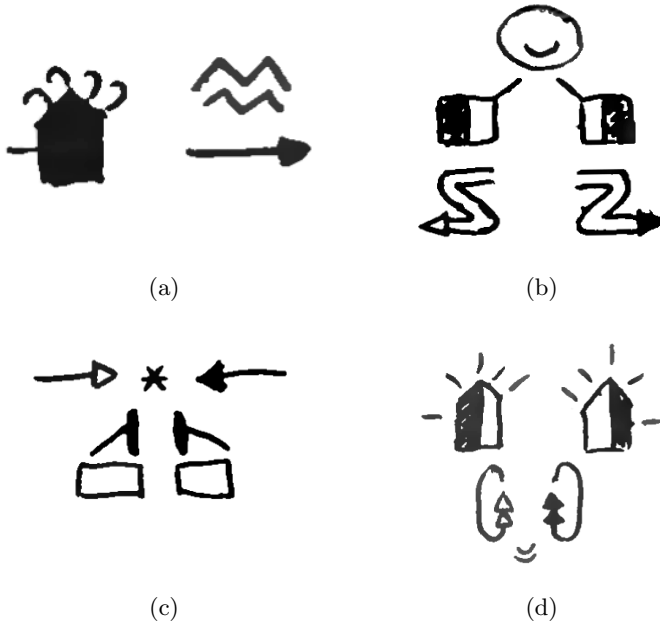


Fig. 5: Some logograms in the corpus which include movements of the hand. In 5a (“fingerspelling”), the hand moves to the right while the fingers “wiggle” (a ‘DIAC’). In 5b (“happy”), the hands simultaneously do parallel zigzagging downward movements. In 5c (“together”), the hands move from the sides to the center until they touch. Finally, in 5d (“Sign Language”), the hands make repeated circular movements towards and away from the body in the vertical plane.

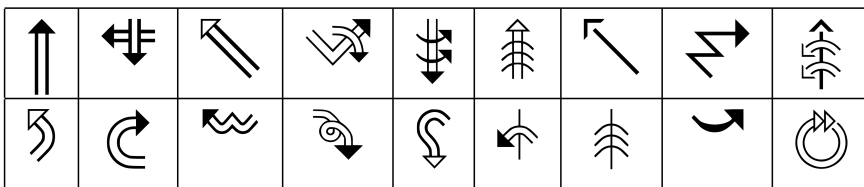


Fig. 6: Small sample of characters that can be found in the Sutton SignWriting fonts to represent different movements of the hand.

In Figure 6 we can see that arrow heads are repeating elements, as well as straight path segments, arcs and circles. Since each of them has a distinctive meaning (black arrow heads represent right hand movements, white ones left hand movements, double stemmed movements occur in the vertical plane, etc.) and there are movement marks which are distinguished only by the presence or

absence of one of them, it is reasonable to think of each independent segment as individual graphemes.

Therefore, we have settled on three different classes of movement graphemes: arrow heads ‘ARRO’, straight segments ‘STEM’ and curved segments ‘ARC’. The ‘SHAPE’ tag for each of them distinguishes between the few different variants in the CLASS: in the case of ‘ARRO’, the color of the arrow head marks which hands move, and is encoded in the ‘SHAPE’. ‘STEM’ and ‘ARC’ can occur in different planes of movement, distinguished by their stems, so this is annotated in the ‘SHAPE’. For ARCs, the ‘SHAPE’ additionally stores the amplitude of the movement, since curved movements can be shorter or longer arcs or even full circumferences.

Additionally to the ‘SHAPE’, all movement graphemes can be rotated to convey orientation in 3D space. This is annotated in the ‘ROT’ tag, as is done for hands, and following the same notation using cardinal directions⁷. The ‘REF’ tag is not needed thanks to subdividing movements into segments, since each individual segment is symmetric. If movement markers were annotated as a whole, annotating reflection would be necessary for ‘ARC’s to distinguish clockwise and anti-clockwise orientations, but with our approach it is not necessary (it is marked by the orientation of the arrow head).

Some examples of this path subdivision can be seen in Figure 7.

The subdivision approach also helps with marking the bounding boxes for movement graphemes. Movement paths are very graphically sparse, the actual strokes often occupying a small part of the whole rectangular region they occupy. The bounding box of the full path is not very informative either, not being able to distinguish the actual shape of the paths or their direction. This could be annotated a posteriori, but it seems a waste of effort to try to reconstruct a geometric meaning by abstract terms when the description is there, on the page, using geometric features which we can spatially annotate.

Another instance where the segmenting approach helps is with crossings and overlapping graphemes. It is very common for movement markers to overlap each other. Sometimes this is not a problem, for example a small ARC crossing over a long STEM, as is done to indicate forearm rotation. Their geometric properties are distinct enough that spatial annotation can be reasonably performed, their bounding boxes in a distinguishable and characteristic arrangement even if there is much overlap (see Figure 7b). However, when there is a diagonal crossing of STEMs, as in Figure 7d, the bounding boxes overlap so much that they become meaningless, impeding grapheme discrimination. In this case, the paths are subdivided into consecutive sub-segments, forming a meaningful arrangement of movement graphemes which can be annotated and processed.

Despite the advantages we have explained, subdividing the movement markers present one small problem. With this schema, some markers used in SignWriting to represent body parts like shoulders, the waistline, or forearms become indistinguishable from STEMs. Indeed, in the case of forearms,

⁷For the specific details please see the annotation guide of the corpus.

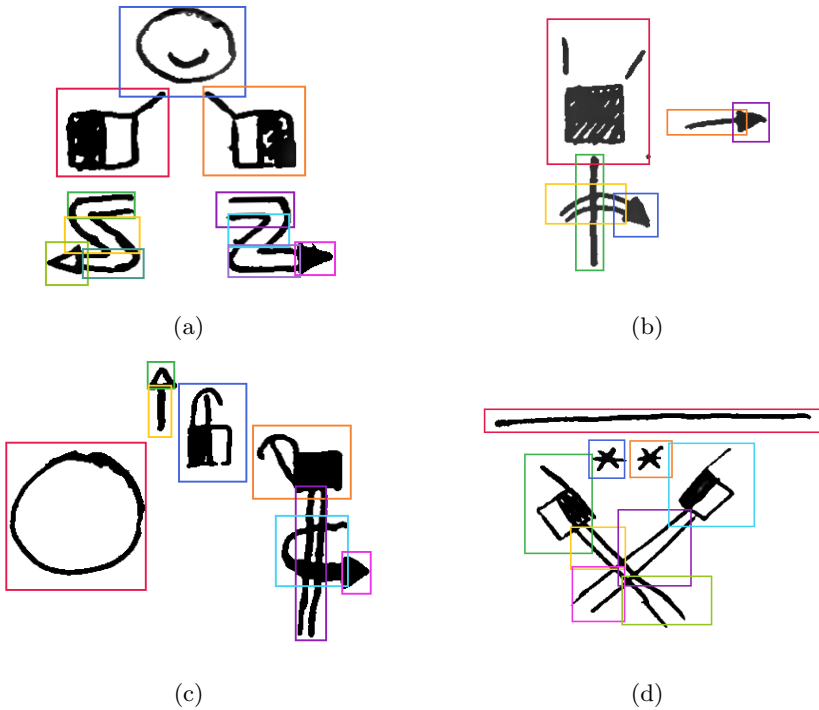


Fig. 7: Annotation of sub-segment bounding boxes for movement and forearm markers in some logograms from the corpus. The signs are “happy” in 7a, “long weekend” in 7b, “Zaragoza” (a city name) in 7c and “November” in 7d.

this similarity is not a coincidence, since they have the same features and behaviours of movement paths. They can be double or single, depending on whether the forearm is vertical or horizontal, and they can have overlapping ARCs to represent forearm rotation. The only difference is that these “STEM”s are not topped by an arrow head. Fortunately, this is also the solution to our problem. Shoulders, forearms etc. are annotated as STEMs, and their actual meaning is contextual, depending on the presence or absence of actual ARRO graphemes in their extremes. While this pushes interpretation of graphemes’ meaning to a further layer of processing which can take context into account, this is already a feature of SignWriting and our annotation, so this decision maintains coherence and makes the corpus consistent.

4 Corpus Construction

The VisSE corpus was built as part of the equally named VisSE project⁸ (“Visualizando la SignoEscritura”, “Visualizing SignWriting” in Spanish)

⁸<https://www.ucm.es/visse>

which had the goal of improving the use and access of SignWriting in digital contexts. To form the base for further study as well as training samples for machine learning development, around 1950 instances of raw SignWriting were collected. These samples had been produced by Dr. José María Lahoz-Bengoechea during a span of years while learning Spanish Sign Language at Universidad Complutense de Madrid, and were originally a tool for his private study.

The samples were handwritten with pen or pencil, and collected in vocabulary sheets. As part of the project, they were digitized, separated into different files for each entry, and graphically enhanced to reduce scanning artifacts and other noise. During this process, a reference to the original vocabulary entry was kept, and remains as a tentative gloss (in Spanish) for logograms in the corpus.

The logograms are collected in subsets according to the academic level at which they were collected. This has no further intended meaning than being a way to organize the corpus. Nonetheless, due to the temporal separation of the records, this organisation results in some greater graphical and usage consistency within each of the subsets. This has let us do annotation incrementally, learning from each phase and improving the annotation schema each time a new set was added. In the current release of the corpus, not all of the original logograms are present, but only those which have been annotated and revised. These are sets A1_1, A1_2, A1_3, A2 and B2_2, which include 1146 annotations.

Apart from learning from the annotation process and improving it for the following subset, this incremental approach allowed us to use a bootstrapping approach to annotation. Once the first set had been fully manually annotated, machine learning algorithms were trained on it and used to perform a preliminary annotation of the next subset. The resulting annotations had to be checked and corrected manually, but the process was somewhat faster. Some graphemes are easy to detect for the machine learning algorithms, meaning the human annotators could focus on the more difficult parts. As the algorithms improved, the speedup was evident, and some of the tasks could be somewhat automatized, like the drawing of bounding boxes for each grapheme. Readers interested in the machine learning aspect of our research can find more information in [Sevilla, Díaz Esteban, and Lahoz-Bengoechea \(2023\)](#).

4.1 Quevedo

The process of collecting, organizing, annotating, and performing machine learning on the data was a complex one, compounded by the fact that we were developing the annotation schema in parallel to the actual annotation of the data. Moreover, our annotations are complex and very specific, including both visual annotation of logograms and a multi-feature annotation of graphemes. To deal with this complexity and the specific requirements of our task, a specialized tool was developed as part of the VisSE project, named Quevedo⁹.

⁹ Available at <https://github.com/agarsev/quevedo>. An article detailing its features and internal working can be found in [Sevilla, Díaz Esteban, and Lahoz-Bengoechea \(2022\)](#).

Therefore, computational access to the corpus and its features is easiest when using Quevedo, and the on-disk format and organization of the corpus is as a Quevedo dataset.

Quevedo is available on the Python Package Index, so installing it can be done with the command `python3 -m pip install quevedo[web]` if Python and Pip are available. This will also install the web interface, which can be launched with `quevedo web` at the corpus root directory, allowing visual inspection of the logograms and their annotation.

The features of Quevedo and the format on disk are all explained in the online documentation, but are also briefly detailed in the following for parties who want to use the corpus with other tools or need access to the low-level details. All formats are open and standard, so all the data and features in the corpus can be thus accessed.

4.2 Computational representation and access

Logograms in the corpus are stored in the `logograms` directory, in subdirectories representing each of the subsets. Files are sequentially numbered, starting from 1, and each instance consists of two files. The source image is named with the index of the annotation plus file extension `.png`, and the annotation data uses the same filename (the index) but with `.json` extension. For example, the annotation data corresponding to image `logograms/A1_1/1.png` can be found in the file `logograms/A1_1/1.json`.

The json annotation file is a dictionary of attributes, among which there is a `graphemes` key containing an array of the different graphemes found in the logogram. Each of them is a dictionary as well, having a `box` key with the coordinates of the bounding box, and a `tags` key which is another dictionary representing the mapping from feature names to feature values.

The coordinates of the bounding boxes are 4-tuples of floating point numbers, in the format (cx, cy, w, h) . (cx, cy) are the coordinates of the center of the box relative to the logogram, which range from 0 to 1, $(0, 0)$ being the top left corner, and $(1, 1)$ the bottom right one. (w, h) are the width and height of the grapheme region, again relative to the width and height of the logogram, so ranging from 0 to 1. The grapheme tags are stored as strings of characters both for feature names and value.

Aside from the `graphemes` key, logogram annotation files include some other information used by Quevedo. The `meta` key stores a dictionary of additional metadata keys for the logogram, where the original gloss for the logogram can be found, as well as some boolean ‘`flags`’ which we have used to mark and exclude a few problematic graphemes.

There is also a `fold` key which stores a number, the index of the fold to which the annotation belongs. Folds can be used to split the data in the corpus along a different dimension from that of the subsets, which can be useful, for example, for logically partitioning the data into training and evaluation sets. Storing this split into the annotations as a fold number helps make experiments reproducible and sound and results comparable. Each logogram is assigned

a number ranging from 0 to 9, and this numbers split the corpus data into 10 approximately equally-sized folds. In our experiments we use folds 0-7 for training, and 8-9 for testing.

An example annotation file can be seen in appendix A.

4.3 Other files

Inside the corpus root directory there are a number of other files and directories not mentioned above. Especially relevant is the ‘**networks**’ directory, where the weights of neural networks trained with the corpus data are stored. These are included with the corpus so that interested parties can reproduce some of our experiments and pipelines without having to train the algorithms themselves. Visual testing of the networks and pipelines is also possible using the Quevedo web interface.

In the ‘**scripts**’ directory, some utility Python scripts are also included, and some researchers might find them useful. A ‘**dvc.yaml**’ file can also be found in the root directory, for use with DVC (Kuprieiev et al., 2021). This file can be used to run some common tasks with the data from the corpus. Relevant here may be the ‘**extract**’ step, which will extract all graphemes from the logograms and turn them into their own annotation files in a ‘**graphemes**’ directory, allowing them to be processed independently of the rest of the logogram.

For more information on the dataset structure or other files, please refer to Quevedo’s documentation at <https://agarsev.github.io/quevedo>, or to our other articles.

5 Data Description

There are 1146 annotations in the corpus, 982 of which are fully annotated logograms. The rest of the annotations are marked using the ‘**exclude**’ flag, most of them being long transcriptions of polysyllabic signs rather than single logograms. These have been split into two (or more) independent logograms, but the original annotations are also included for reference. Some other transcriptions, marked with the flag ‘**problem**’, present some kind of graphical or representational problem, which has led us to exclude them for now from annotation, but are kept in the corpus.

Within the 982 fully annotated logograms, 6060 different graphemes can be found. Of these, 330 belong to the **HEAD** class, 1047 to **DIAC**, 1649 to **HAND**, 1369 to **ARRO**, 1292 to **STEM** and 373 to **ARC**. In table 3, these numbers can be compared to the number of different **SHAPES** that can be found for each **CLASS**. As can be seen, the proportions are very different, meaning that some classes, like **ARRO** or **DIAC** have only a few different possible grapheme **SHAPES** but are abundantly represented in the data, while other classes like **HAND** or **ARC** are less abundant compared to the variability within the class.

Examining the rate of appearance of grapheme classes per logogram, we can also make some interesting observations. The average amount of graphemes

CLASS	Graphemes	Unique Observations	SHAPEs	Appearance Rate
HAND	1649	560	72	1.68
ARRO	1369	23	3	1.39
STEM	1292	15	2	1.32
DIAC	1047	19	19	1.07
ARC	373	37	6	0.38
HEAD	330	20	20	0.34
total	6060	674	122	6.17

Table 3: Counts of observations in the corpus by CLASS. Unique observations refer to those which share the same set of features. The rate of occurrence is the number of observations divided by the total number of logograms, measuring how likely a grapheme class is to appear in a logogram.

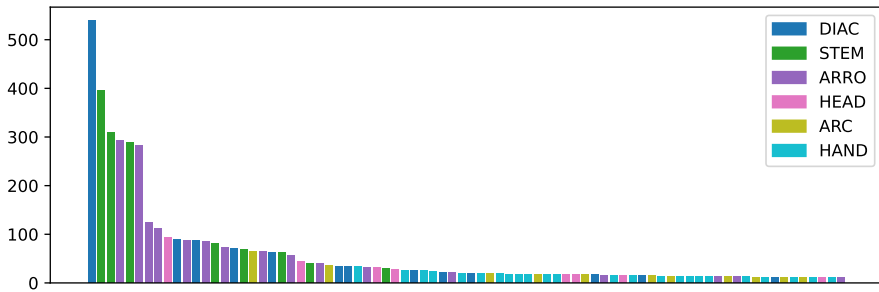


Fig. 8: Distribution of unique tag combinations in the corpus. The most common 80 are arranged in the horizontal axis (not labeled for clarity), while the vertical axis represents the number of times that unique combination appears in the corpus. The bars are also color-coded by CLASS. The plot would extend to the right more than 6 times, making the long tail even longer and thinner.

per logogram is about 6, 1.68 of which are hands. Unfortunately, we cannot distinguish between bimanual signs and transcriptions where a transformation in handshape is encoded, but further, semantic annotation would make this clear. Easier to compute is the complexity of movement paths. Since most movements are marked with an arrow head, the ratio of segments (STEM and ARC) to arrow heads (ARRO) can give us an approximate measure of the mean number of segments for paths: $\frac{1292+373}{1369} \approx 1.22$. This means that most movement markers are simple, with just one stem segment, but a non-trivial amount (approximately one every five) is more complex, having two or more segments.

If we examine the distribution of tag combinations, however, we can see a very skewed distribution, as is depicted in Figure 8. Some graphemes are very common, while many combinations are rare, forming a very long tail of infrequent graphemes. This also happens if we just look at the SHAPE feature, and across classes, as can be seen in Figure 9.

Since this is not a corpus of real utterances or texts, but rather a vocabulary, conclusions cannot be directly inferred about the frequency of elements in

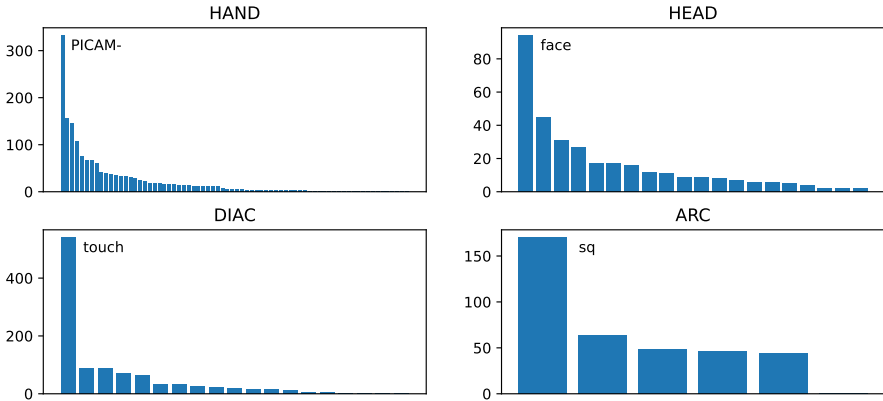


Fig. 9: Distribution of shapes for some classes. The horizontal axis represents the shapes (not labeled for clarity), the vertical axis is the number of times that shape appears in the corpus. The most common CLASS is labeled near its bar for reference.

SignWriting in use. However, we can make observations across the vocabulary of the transcribed sign language, seeing that some particular gestures (understood in the broad linguistic sense) are much more common than others. For example, the “touch” grapheme is extremely common, as well as the PICAM-hand shape (fingers extended but together, which acts as the “flat” object descriptor in Spanish Sign Language, the top left hand shape in Table 1).

6 Conclusion

SignWriting is a featureful writing system used to transcribe sign languages. Its power as a faithful phonetic representation of signs, both in the lexical and the spatial domains, means it can be a useful tool for empirical linguistic research. For this to work, examples and empirical processing methodologies need to be developed and tested.

To this end, we have built the VisSE Corpus of Spanish SignWriting, a collection of handwritten SignWriting logograms which can be used for automatic processing or linguistic analysis of the data. The corpus is freely available online, and in this article we have presented its annotation schema and construction, and we have performed a brief analysis of the data it includes. We have seen that processing SignWriting requires careful analysis, and believe our work may be useful for the research community both in how we have proceeded as well as in the end result, the published corpus.

As an example of the insight that can be gained from having data and annotating it, we have observed in our research some similarities between SignWriting and oral language writing systems, and some meaningful differences. This similarity and differences parallel the relation of sign languages with oral

languages: where sign languages are similar to oral ones, SignWriting resembles oral writing. Both differ to their oral counterparts in the same way: use of space to convey meaning and syntactic relations.

Another insight can be found in how to think of SignWriting. Due to the complexity of sign languages at the phonetic level, SignWriting can be a powerful and flexible tool for dealing with them, thanks to its very detailed and phonetically precise nature. However, it requires a mental and computational model more complex than that of other writing systems. To deal with this complexity, we think it is useful to think of SignWriting as a language in itself, even if a graphical one.

This mental model can be seen in our classification of graphemes in hierarchies, in which if we think of SignWriting as a language, the CLASS tags might be parallel to parts of speech, and SHAPE to particular words. The rest of the annotation, using sets of features, might somewhat resemble the morphological features of words. In the corpus and in this article not much attention has been paid to these semantic properties of graphemes, since we have focused on their descriptive annotation, but we want to note that this similarity of SignWriting elements to words in an oral language, with their syntactic and morphological properties, is not only found in their appearance, but also in their interpretation. Properly extracting this interpretation is a task which remains as future work.

On the other hand, just as space and movement are still unresolved issues for sign language theoretical description, annotation of spatial properties of SignWriting graphemes has been a challenge in itself. We could not base ourselves in any widespread linguistic standard or know-how, since these characteristics are not found in oral languages. We have seen that some of the graphical attributes of graphemes are similar to morphological derivational processes, manifesting as rotations and reflections of characters rather than their insertion or deletion. But others, intrinsically locative, require numerical annotation of positions and regions, and subdivision of paths into smaller elements.

We believe this insight might not be only useful for the computational treatment of SignWriting, but may also mirror some of the problems of computational treatment of sign language per se, and may inspire solutions or ideas in that space.

Moreover, as SignWriting elements map very well to the phonetic features of sign languages, especially when annotated in detail like is done in the VisSE corpus, we think that the study of a corpus of SignWriting can very well be useful to draw conclusions about the sign language it transcribes. This can help make data-based linguistic research on sign languages less costly, as collection of video corpora requires much time, face-to-face collaboration of native informants, and attention to issues such as privacy and distribution of the videos. Linguistic annotation of video is difficult and essentially a manual process, while annotating SignWriting can be faster with the right tools, or

can be even done with hand written samples as we show in this corpus, and using the machine learning algorithms trained on it.

7 Future Work

While the presented corpus is currently usable (as demonstrated in our real use-case application described in [Sevilla et al. \(2023\)](#)), it represents only an initial step in its development.

There remains a significant portion of the corpus to be annotated, and this further annotation might uncover issues that require some revision of the annotation schema.

Additionally to expanding the coverage of the annotation, the data collected can be augmented along two axes. Collecting data from more informants will make the corpus more robust and representative of actual SignWriting. Collecting data from continuous, real-world usage, instead of isolated examples, may also further this goal.

On a different note, these last years have witnessed significant advancements in deep learning approaches. New, state-of-the-art algorithms may be able to tell us more about our collected data, expedite the annotation process, or maybe even open up avenues for exploitation which we had not thought about before.

Finally, it is our hope that the public availability of the corpus will invite other researchers to contribute to this future work or add their own insights. The involvement of a broader research community will surely lead to a richer understanding of SignWriting or even sign languages themselves.

Appendix A Example annotation file

Listing 1: Simplified JSON annotation file for the transcription in [Figure 2](#)

```
{
  "meta": {
    "glosa": "Mentira"
  },
  "fold": 8,
  "graphemes": [
    {
      "tags": {
        "CLASS": "HEAD", "SHAPE": "cheekr"
      },
      "box": [ 0.3561, 0.3867, 0.3992, 0.4156 ]
    },
    {
      "tags": {
        "CLASS": "DIAC", "SHAPE": "rub"
      },
      "box": [ 0.5134, 0.1207, 0.0944, 0.1507 ]
    },
    {
      "tags": {
```

```

    "CLASS": "ARRO", "SHAPE": "b",
    "ROT": "N"
  },
  "box": [ 0.6032, 0.1532, 0.1159, 0.1145 ]
},
{
  "tags": {
    "CLASS": "STEM", "SHAPE": "s",
    "ROT": "N"
  },
  "box": [ 0.6038, 0.3787, 0.0619, 0.25 ]
},
{
  "tags": {
    "CLASS": "HAND", "SHAPE": "I", "VAR": "b",
    "REF": "y", "ROT": "N"
  },
  "box": [ 0.7224, 0.5644, 0.1913, 0.3229 ]
},
{
  "tags": {
    "CLASS": "ARRO", "SHAPE": "b",
    "ROT": "N"
  },
  "box": [ 0.6025, 0.2363, 0.1078, 0.0763 ]
}
]
}

```

References

- Cassidy, S., Crasborn, O., Nieminen, H., Stoop, W., Hulsbosch, M., Even, S., ... Johnston, T. (2018). Signbank: Software to Support Web Based Dictionaries of Sign Language. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- da Rocha Costa, A.C., & Dimuro, G.P. (2002). SignWriting-based sign language processing. *Gesture and sign language in human-computer interaction* (pp. 202–205).
- Eccarius, P., & Brentari, D. (2008). Handshape coding made easier: A theoretically based notation for phonological transcription. *Sign Language & Linguistics*, 11, 69–101.
- Galea, M. (2014). *SignWriting (SW) of Maltese Sign Language (LSM) and its development into an orthography: Linguistic considerations* (PhD Dissertation). University of Malta.
- Gutierrez-Sigut, E., Costello, B., Baus, C., Carreiras, M. (2016). LSE-Sign: A lexical database for Spanish Sign Language. *Behavior Research Methods*, 48, 123–137.

- Hanke, T. (2004). HamNoSys – Representing Sign Language Data in Language Resources and Language Processing Contexts. *Proceedings of the Workshop on Representation and Processing of Sign Language, Workshop to the forth International Conference on Language Resources and Evaluation (LREC'04)* (pp. 1–6).
- Herrero Blanco, Á. (2003). *Escritura alfabética de la lengua de signos española: once lecciones*. Publicaciones de la Universidad de Alicante.
- Hilzensauer, M., & Krammer, K. (2015). A multilingual dictionary for sign languages: “SpreadTheSign”. *ICERI2015, the 8th annual International Conference of Education, Research and Innovation*. Sevilla.
- Kuprieiev, R., Petrov, D., Pachhai, S., Redzyński, P., Costa-Luis, C.d., Schepanovski, A., ... Vera (2021). *DVC: Data Version Control - Git for Data & Models*. <https://zenodo.org/record/5037865>. Zenodo.
- Liu, X., Vermeylen, L., Wisniewski, D., Brysbaert, M. (2020). The contribution of phonological information to visual word recognition: Evidence from chinese phonetic radicals. *Cortex*, 133, 48-64. <https://doi.org/10.1016/j.cortex.2020.09.010>.
- Moreno, D.S. (2012). Proyecto DILSE III: primer diccionario normativo de la lengua de signos española. *Estudios sobre la lengua de signos española: III congreso nacional de lengua de signos española: hacia la normalización de un derecho lingüístico y cultural, madrid 2009* (pp. 297–310).
- Sevilla, A.F.G., Díaz Esteban, A., Lahoz-Bengoechea, J.M. (2023). Automatic signwriting recognition: Combining machine learning and expert knowledge to solve a novel problem. *IEEE Access*, 11, 13211–13222. <https://doi.org/10.1109/ACCESS.2023.3242203>.
- Sevilla, A.F.G., Díaz Esteban, A., Lahoz-Bengoechea, J.M. (2022, June). Quevedo: Annotation and processing of graphical languages. *Proceedings of the language resources and evaluation conference*. Marseille, France: European Language Resources Association.
- Sevilla, A.F.G., Lahoz-Bengoechea, J.M., Díaz Esteban, A. (2022). *VisSE corpus of Spanish SignWriting*. <https://zenodo.org/record/6337885>. Zenodo.

- Slevinski, S. (2016). *The SignPuddle standard for SignWriting text* [Internet-Draft]. <https://datatracker.ietf.org/doc/html/draft-slevinski-signwriting-text>.
- Stokoe, W.C. (1960). Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf. *Studies in linguistics: Occasional papers*, 8.
- Sutton, V., & Frost, A. (2008). *SignWriting: sign languages are written languages!* Center for Sutton Movement Writing.
- Sutton, V., & Slevinski, S. (2010). *International SignWriting alphabet, HTML reference*. <http://www.signbank.org/iswa/>.
- The Unicode Consortium (2021). *The Unicode Standard: Version 14.0.0*. Mountain View, CA: Unicode Consortium.
- Verdú Perez, E., Pelayo García-Bustelo, B.C., Martínez Sánchez, M.Á., González Crespo, R., et al. (2017). A system to generate signwriting for video tracks enhancing accessibility of deaf people. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4 (6), 109–115.