Modeling French Sign Language: a proposal for a semantically compositional system

Mohamed Nassime Hadjadj, Michael Filhol, Annelies Braffort
LIMSI–CNRS B.P. 133, 91403 Orsay cedex, France
{mohamed.hadjadj, michael.filhol, annelies.braffort}@limsi.fr

Abstract

The recognition of French Sign Language (LSF) as a natural language in 2005 created an important need for the development of tools to make information accessible to the deaf public. With this prospect, the goal of this article is to propose a linguistic approach aimed at modeling the French sign language. We first present the models proposed in computer science to formalize the sign language (SL). We also show the difficulty of applying the grammars originally designed for spoken languages to model SL. In a second step, we propose an approach allowing to take into account the linguistic properties of the SL while respecting the constraints of a modelisation process. By studying the links between semantic functions and their observed forms in Corpus, we have identified several production rules that govern the functioning of the LSF. We finally present the rule functioning as a system capable of modeling an entire utterance in French sign language.

Keywords: French sign language, computer modelling, formal grammar

1. Introduction

Since the ‘60s, numerous efforts have proposed linguistic models to describe sign languages (Stokoe, 1960). However, none of the proposed studies has yet succeeded to fully describe the specificities of SL, such as the use of the signing space and the simultaneous articulation of multiple channels. Thus, researchers in Sign Language Processing (SLP) must model languages without fine linguistic descriptions, without a written form and with a limited amount of corpus (Cuxac and Dalle, 2007). In this context, most SL modeling approaches assume that the difference of modality has a minor impact on the organization of the utterance. The modeling of SLs is then carried out primarily using models originally designed for spoken languages. This operation implies linearity constraints that characterize other languages. The aim of this article is to propose a modeling approach to the French Sign Language (LSF), allowing to take into account the specificities relating to their visual-gestural modality.

In the section below (section 2), we present the most important approaches for the modeling of SL. Section 3 present our corpus study methodology to identify production rules that govern the functioning of the LSF. Finally, Section 4 proposes the combination of rules as a system capable of modeling an entire utterance.

2. Research in Sign Language Processing

Research in SLs modeling can be classified into two categories. In the first category, models were based on approaches initially proposed to model spoken languages. In the second category, models were designed specifically for SLs. In the following, we present some examples of approaches that love been developed with aim to translate from text to SL.

2.1 Models initially dedicated to spoken languages

The two projects **Team** (Zhao et al., 2000) and **ASL Workbench** (Speers, 2001) present a translation system from English to ASL using a syntactic transfer. To generate ASL, **Team Project** represents statements with the tree-adjoining grammar (STAGs) (Shieber, 1994). **Workbench**, for its part, use a LFG grammar (Kaplan and Bresnan, 1982). Under the translation project **Visicast** Marshall and Sàfàr (2003), propose a translation system from English to British Sign Language (BSL), this time using a semantic transfer. The semantic presentations in the form of a DRS (Discourse Representation Structure) structure are converted into an HPSG representation (Pollard and Sag., 1994), the utterances are generated using the HamNoSys phonetic model (Prillwitz et al., 1989).

The grammars initially developed to model spoken languages predict a systematic linear sequence of units. In SL models based on these grammars, a signed structure is considered the equivalent of a sentence and a gestural unit as the equivalent of a word. However, a considerable part of LSF contains structures that do not present a linear order (Garcia et al., 2010). The models based on spoken languages are not very efficient to fully describe SLs. To support this statement, we take the following structure as an example: "The city located next to Red Sea", its interpretation in LSF involves gestures that are specific to visual-gestural languages. Signed utterances do not only form a sequence of signs. We have rather a structure as described in figure 1.

1. The sign unit: Red Sea.
2. A gaze direction activates a part of the signing space. The placement of the flat weak hand refers to the Red Sea or what is defined in SL linguistics as "Classifier (proform)".
3. The pointing sign (sign that points to a referent), the gaze direction is always maintained.
4. The sign unit: City

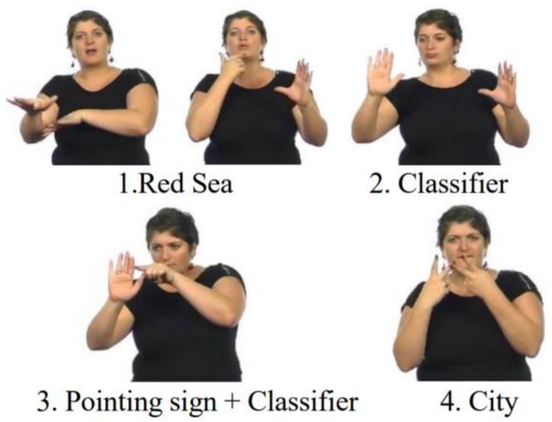

Figure 1: form of the structure “The city located next to Red Sea” in LSF

2.2 Models dedicated to sign languages

The model of Huenerfauth (2006), then taken up by López-Colino (2012) proposes an approach to take into account the multilinearity observed in SL. His model named "P/C" defines two parts to describe a signed structure: The "C-node" corresponds to the linear order of the constituents. The "P-node" allows to segment a node in a structure to create new parallel axes. This configuration allows to simultaneously specify several articulations. However, the different axes are constrained by the boundaries of a partition. Consequently, if this approach offers a way to represent multilinearity, it does not allow to deal with complex synchronization patterns.

3. Presentation of our approach

To describe the linguistic organization of SL, we adopt an approach that does not assume the formal requirements of linear models. In other words, no hypothesis or prediction of a linguistic functioning is advanced beforehand. We rely on a more general descriptive approach to integrate all of the linguistic phenomena observed in SL corpus.

3.1 The key concepts

Our approach is to define, based on corpus studies, a systematic link between **"observable forms"** and what we call **"semantic functions"**. By observable form, we mean any gestural articulation, including the different synchronizations that take place between them. For example, the following articulations:

A: "Eyebrow raising"

B: "Move the index finger down"

Each of these two articulations can be considered as a form. Also, an observable form may include an overlap of A and B. This type of synchronization is also a form.

The notion of semantic function designates an interpretation of such observable form. Contrary to the notion of "signified" in general linguistics, reserved only for concepts of lexical signs, the notion of semantic function in our approach can be linked to different levels of interpretation. In this sense, the examples below are considered as possible functions:

C: the concept of "pen"

D: expression of doubt on a variable element

E: location of an object (*obj1*) in relation to another object (*obj2*)

SLs are, like all languages, considered as a system governed by a set of rules shared by a linguistic community. Our goal is to identify these rules with an experimental study. Any systematic association between an observable form and a semantic function gives rise to a rule, which participates in the linguistic organization of the language. As part of a project to generate the LSF, we identify **production rules** (links from function to form). Once a production rule is defined, it is formalized with the AZee language (Filhol, 2014). Each rule consists of three elements:

An identifier: usually an abbreviated name for its semantic function, e.g. “pen”.

The arguments of the rule: this is the set of parameters on which the rule may depend, e.g. “obj1”&”obj2”.

Associated form: these are the invariant forms of the rule and their possible dependencies to the parameters. This includes all the joints and the necessary synchronization constraints between them.

Box diagrams (figure 2), more explanatory than AZee code, can illustrate a production rule. In those diagrams:

- The horizontal axis represents the production time.
- The boxes represent time intervals in which an articulation must take place. The articulators are set in bold; their positions are given in italics
- The blue boxes are invariant specifications. The boxes in red represent the time intervals during which an argument is to be produced.

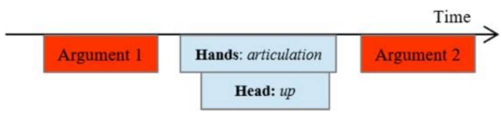

Figure 2: Example of a production rule

3.2 Methodology

The methodology we propose to define production rules consists in performing function-to-form iterations to identify a group of identical parameterized forms associated with the same semantic function. We present the different steps of this approach in the following sequence:

1. The first step is to begin with a starting criterion X to be specified. It can be either a form or a function.
2. Locate and list all occurrences of X in a selected corpus.

   Nocc refers to the number of occurrences found.
3. For each occurrence of criterion X, associate description elements that are:

- Elements of interpretation if X is a criterion of form.
- Observable forms if X is a semantic function criterion

4. Identify groups of occurrences sharing identical elements.
    - *Ngp* is the number of groups identified, groups are numbered from 1 to *Ngp*
    - For each $k \in [1..Ngp]$: D_k is the set of description elements that have been associated with all occurrences of the group *k*
    - *Nout* is the number of occurrences that are not included in any group.
5. **If** X is a function, which may depend on arguments {*A1*, *A2*, …, *An*}, <u>and</u> *Ngp* = 1, <u>and</u> *Nout* < 15% of all occurrences in the only group

    Then: we define a production rule for the function X and its associated form. It is specified by the triplet:

    - Identifier = X
    - Arguments = $\{A_i\}$
    - Form = D_1

    Else: **for each** group k, $k \in [1 ... Ngp]$: proceed to step 1 with the new criterion X.k, specified by D_k

3.3 AZee formalism

To describe forms of the last D_k, we use the AZee language, which is a functional language first described in (Filhol et al., 2014), where a more detailed specification can be found. It defines a specific type "score", whose values are full specifications of a signed articulation (the name of this type is an analogy to a music score on which several voices are specified in parallel). It allows to define "AZops", which are equivalent to functions in a programming language. To define a production rule, one defines an AZop, with named arguments for parameters and with a score as a return value. Applying a rule is like applying the function, with a value for each parameter.

Figure 3 gives an AZop example for the rule placing a weak hand classifier in the signing space, which depends on a classifier "C" (a set of additional constraints to apply to the hand), and on a target point "loc". It produces the arrangement given in the box diagram of figure 4, where boxes (1) and (2) are manual specifications of a small downward movement establishing the weak hand at the argument location. Box (3) specifies the eye gaze, which must also be directed to the same target point. In the code, indentation denotes argument nesting under their header lines. This makes code more readable, but a bracketed notation is also possible.

Lines 3 and 5 declare the AZop's arguments, each followed with "nodefault", i.e. they are mandatory when applying the AZop. The "sync" operator on line 7 takes a list of named boxes to arrange on a timeline and returns a score, which is the return value of the AZop. The three boxes are: "classmvt" (box 1, from line 8) specifying the downward movement," classcfg" (box 2, line 35) configuring the hand with the classifier constraints, and "eyegaze" (box 3, line 45), specifying the eye gaze direction. Lines 42-44 synchronise the start and end boundaries of box 2, and lines 50-52 those of box 3. References to the arguments are prefixed with an '@'. Dependencies on the target location of the classifier are visible on lines 17, 34 and 49. The dependency on the classifier itself appears on line 39.

```
1  "place-classifier-in-space"
2  azop
3      'classifier  # type: AZOP (h:SIDE -> CSTR)
4      'nodefault
5      'loc         # type: POINT
6      'nodefault
7      sync
8          'classmvt  # BOX 1
9          sequence
10             key
11                 0
12                 place
13                     translate
14                         site
15                             'PA
16                             w
17                         @loc
18                     scalevect
19                         up
20                         small
21             transition
22                 1
23                 path
24                     'straight/accel
25                     site
26                         'PA
27                         w
28             key
29                 1
30                 place
31                     site
32                         'PA
33                         w
34                     @loc
35         'classcfg  # BOX 2
36         key
37             1
38             apply
39                 @classifier
40                 'h
41                 w
42         'start/end
43         'classmvt:0:0
44         'classmvt:-1:0
45         'eyegaze  # BOX 3
46         key
47             1
48             lookat
49                 @loc
50         'start/end
51         'classmvt:0:-.5
52         'classmvt:-1:-1
```

Figure 3: AZop example for the rule placing a weak hand classifier in the signing space.

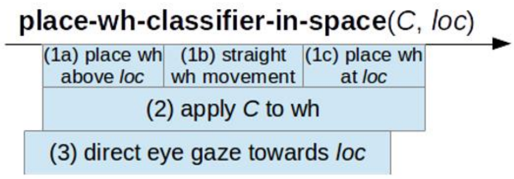

Figure 4: Box diagram example for the rule placing a weak hand classifier in the signing space

4. Juxtaposition as a form

As presented in Section 2, most work in automatic processing of sign languages follows the same process of analysis applied to spoken languages. Thus, the application of derivational grammars on SL requires a systematic sequentiality between the signed units. In our approach, no production rule is defined to satisfy constraints assumed by a formal grammar. Any production rule allowing juxtaposition of signed units must be motivated by a link from function to form.

In order to explain linear structure nonetheless, we begin this study with a criterion of form J: the juxtaposition of two successive and interpretable items. This juxtaposition can relate to the succession of two interpretable units such as "country" and "Brazil" but also between “country Brazil" and its geographical location. We present in the following sub-sections the corpus on which we conducted this study as well as the results obtained.

4.1 Corpus

The corpus we relied on for this study consisted of 40-news item, each is signed by 3 professional signers, totaling 120 videos, or one hour of journalistic signing (Filhol and Tannier, 2014).

4.2 Iteration J

Iteration J: juxtaposition of two units, item1 and item 2

- Nocc = 321
- Ngp = 7
- Nout =21

Group J.1: item 2 gives the status of item 1. This may concern its name, status, property...

Group J.2: item 1 is located in relation to item 2

Group J.3: Item 1 presents the context of item 2

Group J.4: Finger spelling

Group J.5: Chronological sequence of two events

Group J.6: Negation of the first item and assertion of the second item

Group J.7: Enumeration

4.3 New iterations from group J.1

In the framework of this article, we present mainly the study of occurrences from group J.1. Functions from other groups require a finer analysis to be considered as production rules.

Function Criterion J.1: « item2 describes the State of item1

- Nocc = 153
- Ngp = 3
- Nout = 11

Group J.1.1: movement of the chin upwards on the beginning of item2; transition time of 2~3 frames between item1 and item2

Group J.1.2: movement of the chin upwards on the beginning of item1; transition time of 2~3 frames

Group J.1.3: longer transition time (8~9 frames) between the two items Iteration

Form Criterion J.1.1: Juxtaposition of item1 and item2; chin moves up on item2; 2~3 frames between the two items

- Nocc = 70
- Ngp = 1
- Nout = 5

Single group 1: item2 gives additional information on the item1.

Example: "Pierre, aged 25, is taken hostage"; Item1 = Pierre, Item2 = 25 years old

Function Criterion J.1.1.1: side information added

- Nocc = 65
- Ngp = 1
- Nout = 0

Methodological condition (5a) is verified. This therefore raises a **production rule**, specified as:

- identifier: item1 is given the additional side information item2 (**add-info**)
- arguments: base_item (item1), add_info (item2)
- form: see figure 5.

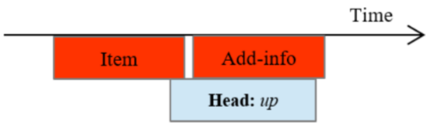

Figure 5: Form of add info rule

Form Criterion J.1.2: Juxtaposition of item1 and item2; chin moves up on item1; 2~3 frames between the two items

- Nocc = 33
- Ngp = 1
- Nout = 6

Single group 1: item2 is to be understood as the category of item1

Example: item1 = “country”; item2 = “Brasil”; combined interpretation = “Brasil”

Function Criterion J.12.1: item2 is to be understood as the category of item1

- Nocc = 27
- Ngp = 1
- Nout = 0

Production rule J.1.2.1

- identifier: item2 is to be understood as the category of item1**(cat)**
- arguments: base item (item1), category (item2)
- form: see figure 6.

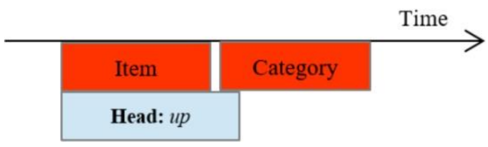

Figure 6: Form of cat rule

Form Criterion J.1.3: Juxtaposition of item1 and item2; longer transition time (8~9 frames) between the two items

- Nocc = 39
- Ngp = 1
- Nout = 6

Single group 1: item2 is the point being made about item1

Example: item1 = "power"; item2 = "tourism"; combined interpretation = "the strength/power is tourism"

Function Criterion J.1.3.1 : item2 is the point being made about item1

- Nocc = 33
- Ngp = 1
- Nout = 0

Production rule J.1.3.1

- identifier: item2 is the point being made about item1 **(info-focus)**
- Arguments : base_item (item1), focus (item2)
- Form: see figure 7.

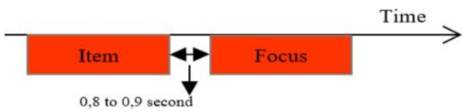

Figure 7: Form of info-focus rule

We present in the table below (table 1) a summary of all the iterations carried out as well as the production rules that were identified from this study.

5. Production rules as a system

The recursive aspect of rules, the embedding of one structure within another, is a feature that generates an infinite set of utterances from a finite set of rules. Each of the rules identified in this study depends on a nesting of arguments to generate a language. In other words, their production requires at least a level of nesting. We hypothesize that this nesting can reach several levels in order to model structures in LSF from a finite set of rules. In order to evaluate this hypothesis, we have described 14 news items of the corpus of the 40. We used production rules observed in this study but also other rules identified in previous studies.

To formalize the complete structure of a news item, we have first described combinations of forms by rules carrying an appropriate semantic function. For example, the description of the succession of the COUNTRY and BRAZIL units, if the observed form allows, is done from the production rule **cat** (J.1.2.1). Once first-level production rules are defined, we try to find production rules that can combine them. For example, the following form in LSF "tourist town Dahab" (figure 8) is described from the rules of production by the nesting presented in figure 9.

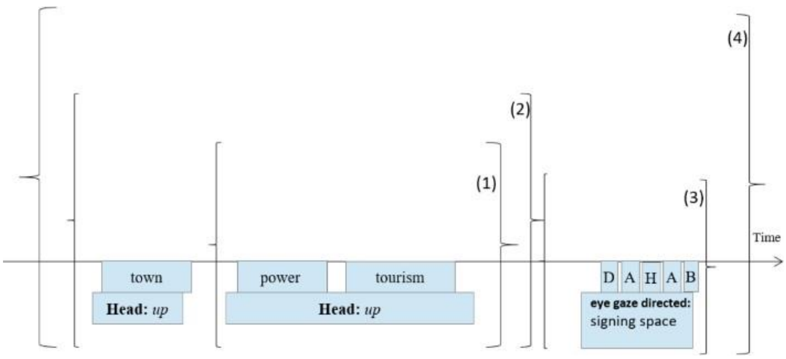

Figure 8: Form of the structure "tourist town Dahab" in LSF

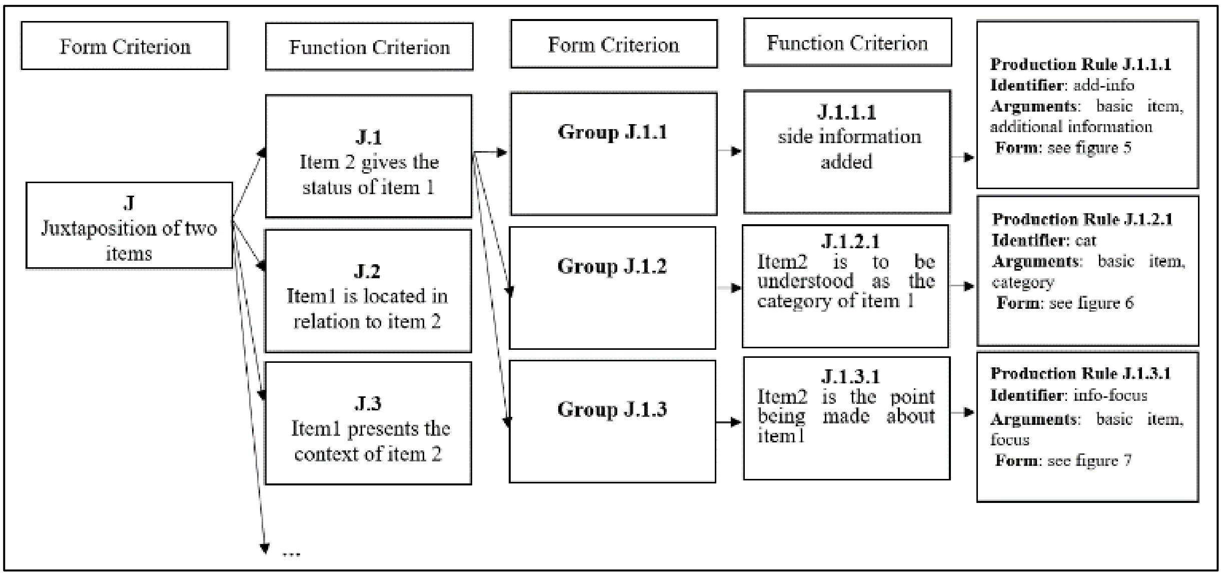

Table 1: synthesis of the study

1. The partition inside brackets 1 corresponds to the form generated by the **info-focus** rule (J.1.3.1). The second argument "tourism" is a focus on the first argument "power"

2. The partition inside brackets 2 corresponds to the form generated by the **add-info** rule. It takes as the first argument "city", its second argument is additional information "power tourism" (the accolade 1)

3. The partition inside brackets 3 corresponds to the form generated by the spelling rule (J.4). It takes a succession of letters as argument.

4. The partition inside brackets 4 corresponds to the form generated by the rule **cat**. It takes as its first argument the rule **add-info** (the accolade 2); its second argument gives the category of the first argument, its name "Dahab" (the brackets 3).

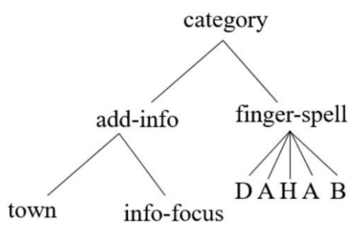

Figure 9: Functional tree "tourist town Dahab" in LSF

From this example, we observe that production rules create a system that allows to produce complete structures from multilevel nesting while respecting the form of the rules and their appropriate functions. Following the same process, we managed to describe most of the structures of the 14 news item. On 321 juxtaposition link, only 21 remain unexplained, often groups of forms with non-identifiable semantic functions. With this experiment, we have moved from a local function-form link that characterizes a single rule to the application of the same principle to describe this time a semantic composition of a structure that requires the nesting of several rules.

6. Conclusion et perspective

This article proposed a linguistic approach to formally describe LSF from corpus data. Using an LSF corpus, we have identified several production rules (semantic function to form links) that allow to juxtapose signed units. Then, we described how these rules could be used as a system to model an entire utterance in LSF. In future work, our goal is to study the possible combinations between the identified production rules. The definition of these patterns should allow to develop a grammar able to evaluate the acceptability of a rule structure in LSF. It should yield a linguistic model more flexible than a syntagmatic grammar.

7. Bibliographical References

Cuxac, C and Dalle, P (2007) Problématique des chercheurs en traitement automatique des langues des signes, volume 48 of Traitement Automatique des Langues. Lavoisier.

Filhol, M ., Hadjadj, MN and Annick. Choisier (2014) Non-manual features: the right to indifference. In Language resource and evaluation conference (LREC), 6th workshop on the representation and processing of Sign Language: beyond the manual channel, Reykjavik, Islande.

Filhol, M and Tannier, X (2014) Construction of a french- lsf corpus, building and using comparable corpora. In Proceedings of LREC 2014

Garcia, B. L'Huillier ., MT, Fremeaux, A ., and Makouke,D. (2010). Discursive and morphological processes in the formation and the stabilisation of lexematic units in french sign language. In Theoretical Issues in Sign Language Research (TISLR), UCL, London

Huenerfauth, M (2006) Generating American Sign Language Classifier Predicates for English-to-ASL Machine Translation. PhD thesis, University of Pennsylvania, Philadelphia, PA

Kaplan, R M and Bresnan J (1982) LexicalFunctional Grammar: A Formal System for Grammatical Representation. In Bresnan 1982b, 173–281. Reprinted in Dalrymple et al. (1995: 29– 135).

López-Colino, F & Colas, J. (2012) Spanish Sign Language synthesis system. Journal of visual languages and computing 23 121-136

Marshall, E and. Sáfár, I (2003) A prototype text to British Sign Language (BSL) translation system. In 41st Annual Meeting of the Association of Computational Linguistics, Sapporo, Japan, pp. 113–116.

Pollard, C and Sag, I (1994) Head-Driven Phrase Structure Grammar, Chicago: University of Chicago Press

Prilllwitz S ., Leven, R ., Zienert, H ., Hanke, T and Henning, J (1989) Hamburg Notation System for Sign Languages: an introductory guide - HamNoSys version 2.0., Hamburg (D), Signum Press

Shieber, S (1994) Restricting the Weak-Generative Capability of Synchronous Tree Adjoining Grammars. Computational Intelligence, 10(4)

Zhao, L., Wiliam, K ., Schuler, C ., Vogler, N ., and. Palmer, M (2000). ìA Machine Translation System from English to American Sign Language. î Association for Machine Translation in the Americas