Spatial and temporal coding of Nicaraguan Sign Language in MediaTagger: Documenting three dimensions with a two-dimensional tool.

Ann Senghas

Abstract: This presentation discusses some advantages and disadvantages encountered using a multimedia tool to analyze Nicaraguan Sign Language (NSL). NSL is a very young language, currently undergoing dramatic changes in both its syntax and morphology. Ongoing research explores the growth of NSL by comparing older and younger signers' use of certain syntactic and morphological elements. The syntactic elements include systematic word order patterns, including repetition; the morphological elements include the modulation of signs with respect to shared or contrasting spatial locations. Previously, this work has been conducted using glosses and coding spreadsheets that referred to video timecode. I have recently begun using MediaTagger, a software program designed for coding the gestures that accompany spoken language. MediaTagger offers many advantages over video and glossing: it allows immediate access to tagged video segments, and permits the linking of those segments directly to the relevant transcription and coding. However, it inherits some of the drawbacks of glossing, drawbacks that are shared with other tools available. In particular, our coding and transcribing tools seem to bias co-temporality (that is, morphemes or signs that occur at the same time) over co-spatiality (that is, morphemes or signs that occur in the same location). This imbalance in all of our tools may make us inadvertently attend to time more than space as we look for patterns in sign languages. Keeping this in mind, we have in MediaTagger a powerful and convenient tool for viewing and coding sign language data.

1. Introduction

MediaTagger is a software program for the Macintosh computer being developed at the Max Planck Institute for Psycholinguistics in the Netherlands. It is designed for transcribing and tagging spoken language and its accompanying gesture. For the past year I have been using MediaTagger for the analysis of Nicaraguan Sign Language (NSL) and found it to be well suited to sign language research. It offers the advantages found in many tools that operate on digital video: immediate access to tagged video segments, and the linking of video segments directly to the relevant text and coding. Click anywhere on a transcript, and see the associated signing; click on any segment of the signing, and see the associated transcription and coding. In addition, MediaTagger offers some particular features for managing and exporting data and video that are useful.

In the interest of comparing this tool with others that are currently available, I will describe some of the features of MediaTagger and how they are being used in research on NSL. I will also discuss some drawbacks that I have become aware of, drawbacks that are shared with other digital tools. These drawbacks have been inherited from glossing and still permeate modern software.

2. The search for morphological and syntactic regularities

The researcher's goal, in using a transcribing and tagging tool, is to find the patterns hidden within a stream of language. The signer onscreen has (unconsciously) used both morphological and syntactic devices to indicate the relationships among the signs that make up a stretch of discourse. A good tagging tool helps us identify those devices, by enabling a researcher to locate signs that are candidates for certain kinds of grammatical links, to tag them, to tally them, and to compare them, thus discovering the regularities among them.

There are certain relationships among signs and their elements that we know are important candidates for linking devices. For example, it may be grammatically relevant that sign elements are produced in the same location, at the same time, or with a shared component (such as the same handshape or facial expression).

We need to create and select our transcribing and coding tools with these patterns in mind. Raw data arrives as a stream of signing, covering the entire signing space, in a sequence of ordered events. Ideally, once we have identified the relevant domain of interest, we should be able to progress through a segment of video, tagging and coding the "relevant" elements; the underlying patterns should then emerge in our analyses. In our ideal software, then, spatial and temporal relationships should be taggable, as well as features of the sign elements themselves.

I have been using MediaTagger to code segments of Nicaraguan Sign Language, and will now turn to that research as a context for demonstrating the effectiveness of this tool for tagging and analyzing such data.

3. A brief history of Nicaraguan Sign Language

Nicaraguan Sign Language is a young language, less than two decades old. It began in the late 1970s, among children entering schools for special education in Nicaragua.

Children entering these schools were not taught to sign. Their teachers did not know anything about sign languages. Nevertheless, a group of deaf children entering by the early 1980s (the "first cohort"), quickly began to converge on a common signing system. By the mid-1980s, this group was entering adulthood. At that time, a "second cohort" of children was entering the schools and learning to sign from their older peers. This group continued to grammaticize the language into the richer system we see today. The grammatical differences between the two cohorts indicate the work, or reanalysis, that the second cohort did, as children.

Today members of the first cohort are in their mid- to late-twenties, and members of the second cohort are adolescents. We can chart the development of this language's grammar by comparing the signing of these first two cohorts. By studying early changes, we learn how new languages organize themselves.

One finding discovered in earlier work is that changes in the language occur first among the youngest members of the community. These developments then percolate up from them both to older signers, and to new younger members who enter the community from that time onward (Senghas, 1995).

In order to understand the details of these developments, we need to closely compare specific constructions, and quantify the differences that we note. For example, we might want to proceed through a collection of utterances produced by signers of both cohorts and pick out the occurrences of particular types of constructions (say, a potential classifier, or a potential type of agreement marking). We might then want to examine comparable utterances, side-by-side on the screen, produced by signers of different cohorts (say, a description, by different signers, of the same vehicles in a row, or of a giving event). Finally, we might want to summarize our data into tables that we can analyze statistically, to see if the differences we note represent consistent differences between cohorts, and hence, changes in the language.

One ongoing study in Nicaraguan Sign Language (Senghas, Coppola, Newport, and Supalla, 1997; Senghas, 1999) is designed to detect the emergence of devices for linking semantic arguments (like agent and patient) with their respective verbs. When signers describe a single event involving two participants, such as when a woman taps a man, how does a signer indicate who is the tapper, and whom gets tapped? The segment shown in clip 1 shows how a first-cohort signer expressed this event.

Insert clip 1 here

As you can imagine, there are both spatial and temporal ways that this information can be expressed, and the use of both means has been changing over time. In order to capture these changes, there are certain comparisons we need to make. I will discuss how we can use MediaTagger to assist us in making these comparisons. In doing so, I hope it will be clear how MediaTagger could be used for other kinds of signed data as well.

4. MediaTagger as a transcribing and tagging tool

4.1 Components

MediaTagger is an application written for the Macintosh computer for coding QuickTime digital video files. It allows users to link structured text and time references to video. MediaTagger requires the following components:

• software: You will need two programs, MediaTagger (at 1.1 MB, it will fit on a floppy diskette) and QuickTime 2.1 or higher. You can obtain MediaTagger by contacting Peter Wittenburg at the Max Planck Institute for Psycholinguistics in Nijmegen, the Netherlands. He is best reached by email at Peter.Wittenburg@mpi.nl. QuickTime is available commercially.

• hardware: MediaTagger requires a Macintosh PowerPC or better, with 16 MB of memory.

• MediaTagger files: These files are created automatically when you use the MediaTagger program. They are quite small (mine average 100K for a half-hour of coding). MediaTagger files consist predominantly of text, along with invisible reference information linking the text to specific moments in a QuickTime movie file.

• movie files: QuickTime movie files are the largest component. Currently one hour of video, digitized and compressed, is 1.8 GB. For this reason, you may choose to store video files on a server and download them as needed. Within the limits of your computer's capacity, movies can be as long or short as you like, from a single sign to a long conversation.

4.2 Tiers and Tags

The first step in coding is to set up a number of levels, or tiers, in which text tags can be entered. For example, one might have a tier for sign glosses, on which individual glosses would be entered, and another tier for eyegaze, in which each change in eyegaze direction would be entered. These tiers are user-defined, and can be selected for the task at hand. It is

not necessary to start with glosses. In fact, you could elect not to include glosses at all if they aren't relevant to your coding task.

Conceptually, the MediaTagger file resembles a large spreadsheet, in which tiers correspond to rows, and individual tags correspond to cells in those rows. An important difference is that rather than being arranged into columns, the tags are time-locked to a segment of video. That is, each tag on a tier is associated with a beginning and ending time. MediaTagger registers these times automatically, as timecode frame numbers, when you select a segment of video using a slider bar and buttons. Once you type in the contents of a tag, the tag is linked to the selected video segment.

Thus, to enter data on a *sign gloss* tier, you might watch a segment of video corresponding to the sign MAN, click on the beginning and ending frames of the sign, and type the word "MAN" into the insert window provided. You could proceed through the sign gloss tier in this manner, typing in glosses, ignoring for the moment other features such as direction of movement, or eyegaze. You could later proceed through the video again, this time entering data on an *eyegaze* tier. Again, you would first select a segment of video, and this time enter the direction of gaze for that segment.

By coding different types of information on different tiers, you can reduce your notational complexity, and concentrate on only those factors relevant to the present research question. Other features can be coded at a later time. They can also be coded on separate tiers by different coders who can't see each others' tiers, if you want to calculate reliability.

It is possible to restrict allowed entries on a tier to a list of preset choices. For example, you might limit *eyegaze* tags to the set of choices *interlocutor*, *imitative*, *configurational*, and *ref-track*. When it is time to enter the contents of the tag, a pop-up menu will appear, allowing you to select among only these choices. It is always possible to add other choices to the preset list later.

It is possible to use specialist fonts within the tag window. Accordingly, it is possible to have a tier in which you transcribe in Hamburg Notation or other alternative fonts. (However, it may not be possible to use such fonts within the search and query commands mentioned below.)

4.3 Tier Dependencies

In the examples above, tiers are independent of each other. However, sets of tiers can be organized into a hierarchical structure of parent and dependent tiers. Tags on a dependent tier are constrained or locked to tags on a parent tier. Their are two types of dependencies these tags may have: they are either *included in* or *attributes of* the tags on the parent tier. These relationships are schematized in Figure 1.

---

insert tier schematic here.

4.3.1 Inclusion dependencies

Inclusion dependencies arise when a segment of video is divided up into shorter subsegments. For example, a parent *sentence gloss* tier could have a dependent *sign* tier. In this example a movie would be divided into sentences on the *sentence gloss* tier. Each sentence would be

further subdivided into signs on the *sign* tier. Every sequence of signs would be "included in" a parent sentence, and the frames linked to the first and last signs in the sequence could not extend past the initial and final frames of the sentence.

To take another example, another level down on the hierarchy, the *sign* tier could in turn be the parent tier to a dependent *segment* tier. Each sign on the *sign* tier would be divided into holds and movements on the *segment* tier. One sign might be ten frames long, its first four frames constituting a hold, the next three frames a movement, and the final three frames another hold. Thus, each movement tag on the *segment* tier would be "included in" a particular sign on the sign tier.

4.3.2 Attribute dependencies

In attribute dependencies, the beginning and ending of each tag on the dependent tier correspond exactly to those of a tag on the parent tier. For example, you might have a parent tier for *sign* with a dependent tier for *word class*. Each sign on the *sign* tier would be assigned a word class on the *word class* tier. For example, the sign MAN might be given the word class NOUN. Note that the tags on a dependent tier apply to the entire tag on the parent tier, and hence are an "attribute of" it.

A parent tier can have several dependent tiers, of both types. However, each dependent tier has only one parent tier. This results in a hierarchical structure of tiers. Adjustments made to tags on the parent tier will automatically be made to its time-locked dependent tiers. If you decide to change the end time for the sign MAN, the end time for its attribute NOUN will automatically be changed to match.

MediaTagger comes with no preset tiers; all of the tiers and their dependencies are set up to suit the research questions and the theoretical framework of the researcher. Analysis can occur at any level, from turn-taking patterns across a conversation, to the timing of sign-internal segments. In fact, the coding need not be limited to language data. You could tag when a mother enters and leaves a room, and a toddler's subsequent response. You could code when each shot was attempted at a basketball game, and by whom.

4.4 Viewing and navigating

Whenever a MediaTagger file is opened, the associated video is also opened and can be watched in a QuickTime movie window. Any number of tiers, selected by the user, can be displayed as the video plays. Text that pertains to a particular time segment will appear below the video window during the segment. Sentence tags will appear for the duration of the sentence, sign tags will appear for the duration of the sign. The tag text does not stream by; it appears and then disappears. This has the advantage of precisely indicating the frames each tag pertains to. It has the disadvantage that tags associated with very short segments (for example, a hold that is only a few frames in duration) will blink by so quickly the user won't be able to read them at normal speed. Of course, such tags will be readable in slow-motion or frame-by-frame viewing and during targeted searches. (Streaming is planned for future versions).

Separate windows, to the right of the movie window, display lists of tag contents with their associated time codes. There is a separate window for each tier, and these windows can be opened and closed independently. If you have created a tier for sentence glosses, the *sentence gloss* tier window will look very much like a transcript. If you have created a tier for sign glosses, the *sign gloss* tier window will look very much like a vocabulary list. The user can

browse through the tier windows, click on text of interest, and immediately watch the associated video segment.

Alternatively, a search command is available to locate particular text within any tier window. With this feature, a user could take a movie of coded dialogue and easily examine every production of the sign MAN, every example of eyegaze on the interlocutor, or all the signs produced with a "B" handshape. An experimenter could jump immediately to the response to item #27 for each subject. Such features make working with digital video extremely appealing.

(A waveform of the audio signal can also be displayed in a separate window. This feature is most useful for coding temporal aspects of spoken language, but might be useful to a sign language researcher who wants to know the nature or timing of sounds that occurred during the filming.)

4.5 Integration with other software

After viewing and tagging has been completed, the researcher essentially has a table of text that is time-linked to video. There are three ways to work with this data: one can conduct queries to locate relevant subsets of the data, export text with timecode for quantitative analysis, or export text with video for presentation.

4.5.1 Queries

If your files are part of a larger database (such as the European Distributed Corpus (EUDICO) database) you can do query searches across MediaTagger files. MediaTagger is designed to seamlessly accommodate the query software Oracle. Using this software's simple Boolean interface, you can create queries that refer to the contents of multiple tiers and their relationships. Some example queries are listed below:

- List all examples of two consecutive signs that are both two-handed.
- List all examples of PUSH in which the goal is to the right or far right.
- List all examples of nouns that are executed with eyegaze on the interlocutor at the beginning of the sign or within ten seconds before it.
- List all examples of verbs that incorporate a handling classifier and that move from center to left or far left, that are immediately followed by eyegaze to the right.

The output of such a query would include a list of files and the relevant tags. This information could then be subjected to frequency counts, or one could return to the original clips for further analysis. The most powerful aspect of these queries is the ability to find complex temporal relationships among features, in addition to simple co-occurrence.

4.5.2 Exporting text as a table

The contents of the tier windows can be exported as a table of text. This is not a fancy feature with bells and whistles, but it is nevertheless extremely useful. In the current version of MediaTagger all of the contents of all of the tier windows are exported into a single table with three columns, listing the starting timecode, the ending timecode, and the tag content. This table can be somewhat unwieldy, but if you import it into a spreadsheet program such as Microsoft Excel you can trim it and organize it into something more manageable. With Excel you can eliminate all but the tiers of interest, sort the data by timecode such that co-temporal features are listed together, convert columns to rows, and perform other manipulations to prepare the data for importation into statistical software. You can do basic analyses in Excel, as

well as summarize the data in simple charts and figures. Simple Boolean queries are also possible (although not ones involving complex temporal relationships).

4.5.3 Exporting selected videoclips with text

After you have watched and coded a videoclip, any segment of that video, along with whichever tiers you have displayed, can easily be exported as a QuickTime movie. This movie can then be imported into digital documents, such as webpages, Powerpoint slideshows, and electronic journals. The video segment displayed above in clip 1 was exported in this manner. To view the segment, users simply click a button and watch the movie, along with its accompanying tags. This is very useful for displaying your coding, synchronized with the video examples you provide. It can also be used to subtitle video segments with a translation.

If you export a single frame from the video, you can import it into documents as a videograbbed picture. This is particularly useful for including pictures of signs in printed documents.

4.6. Summary of features

To summarize, MediaTagger offers the following features:

4.6.1 MediaTagger-specific features

- ∑ automatic registration of time data
- ∑ time-aligned, hierarchical coding structure
- ∑ synchronized tagging (not linear text transcription)
- ∑ coding categories tailored by the user to the task at hand
- ∑ use of alternative fonts in tag windows
- ∑ immediate links from text to video, and from video to text
- ∑ text search engine
- ∑ multiple video windows possible, to compare segments across files
- ∑ multi-user coding of the same video data (for reliability or blind coding)
- ∑ complex analysis by database queries possible (using Oracle query software)
- ∑ output of coding text to spreadsheet software
- ∑ output of video segments, with selected coding, as QuickTime movies

4.6.2 Digital video features

- ∑ no video decay with time
- ∑ no video decay with multiple generations of copying
- ∑ immediate, nonlinear access to any frame
- ∑ playback at any speed, including frame-by-frame, forward or backward

5. Implications

Using digital video coding software like MediaTagger has clear advantages over glossing, and the experience of coding with MediaTagger is very different from glossing and coding by hand. Nevertheless, there is one feature both systems share: text is organized primarily along a temporal axis. As in glossing, the space on the screen, from left to right, is used to represent time. Throughout the software, this temporal priority holds: data are synchronized temporally, queries refer to temporal relationships among tiers, tiers are displayed with co-temporal events appearing above each other, in a stack. For this reason, there is a considerable level of flexibility and power available particularly for analyses examining temporal factors.

After the coding is completed, the researcher can consider possible interesting temporal relationships, and conduct a query to see if such relationships hold. Do certain features usually occur at the same time? Does one generally follow the other within some short window of time? Does one consistently begin after the other has begun, but before it is completed?

Even during analyses, the timeline can be divided and redivided without losing the original temporal information. A segment of video is carved into utterances, and then phrases, and then words, but still the continuous aspect of time is never lost. It is still retrievable in the representation.

No other feature can be automated or redivided this way. If several signs use the same handshape, for example, the pattern won't emerge unless you decide ahead which are the relevant handshapes are to code, and then code for them. This is the only conceivable option for features like handshape that aren't continuous in nature.

Space, on the other hand, is continuous in nature. Nevertheless, like handshape, spatial location must be coded categorically, into groupings decided ahead of time. To analyze spatial uses of signs, a coder must code each sign with respect to a set of discrete spatial values, such as L, LL, R, and RR. Since shared spaces aren't lined up on any one axis, patterns that aren't predicted and searched for will never serendipitously emerge.

This system is clearly better than what we've had before, but be alert that it doesn't offer the same jump in power in the spatial realm that it offers in our ability to analyze temporal patterns. We may, as a result, inadvertently attend to time more than space as we look for patterns in sign languages, like the man who keeps looking for his keys under the lamp post, only because that's where the light is best.

We will use this tool most effectively by keeping this temporal advantage in mind. To discover spatial relationships among signs will still require a set of clear hypotheses and a good eye. But you can throw your clock away, at least.

Acknowledgments
The author is grateful to Hennie Brugman, Sotaro Kita, Asli Ozyurek, Mandana Seyfeddinipur, Carlien de Witte, and David Wilkins for their assistance in preparing this presentation, and for introducing her to MediaTagger. Information for this manuscript was drawn from the *MediaTagger Software Manual*, written by Hennie Brugman and Henk-Jan van Soest of the Max Planck Institute. Research and writing was supported by the Max Planck Institute for Psycholinguistics, Nijmegen, the Netherlands.

References

Senghas, A. (1995). Children's contribution to the birth of Nicaraguan Sign Language. Ph.D. dissertation, Massachusetts Institute of Technology. Distributed by MIT Working Papers in Linguistics.

Senghas, A. (1999). The Development of Early Spatial Morphology in Nicaraguan Sign Language. Delivered at the Boston University Conference on Language Development, November, 1999.

Senghas, A., M. Coppola, E. L. Newport, and T. Supalla. (1997). Argument structure in Nicaraguan Sign Language: The emergence of grammatical devices. In E. Hughes, M. Hughes, and A. Greenhill (Eds.), *Proceedings of the Boston University Conference on Language Development,* 21. Boston: Cascadilla Press.