ELAN as a search engine for hierarchically structured, tagged corpora

Joshua Wilbur
Albert-Ludwigs-Universität Freiburg
Department of Scandinavian Studies
Freiburg Research Group in Saami Studies
joshua.wilbur@skandinavistik.uni-freiburg.de

2018-12-20

Abstract

The main goal of this paper is to outline and explore the usefulness of the corpus search functionalities provided in the ELAN annotation application when annotations are provided in hierarchically organized tiers. A general overview of ELAN's search functions is provided first, highlighting the program's usefulness as a corpus search engine for corpus and computational linguists. To illustrate this, the updated hierarchical tier structure for ELAN developed by the Freiburg Research Group in Saami Studies for the group's projects on both Saamic and Komi languages is presented as an example template. The suitability of hierarchical structures for annotations and the ELAN search interfaces for doing corpus linguistics is explored critically, including the description of a fundamental flaw in the "Multiple Layer Search" mode which likely prevents ELAN from being used as a search engine for complex corpus studies.

Kokkuvõte

Artikli peamine eesmärk on kirjeldada ning uurida, millised on programmi ELAN korpusepäringu võimalused, kui materjal on annoteeritud hierarhiliste kihtidena. Selleks antakse kõigepealt ülevaade programmi otsimootori üldistest võimalustest, tuues välja selle kasulikud omadused korpus- ja arvutilingvistide jaoks. Näitena tutvustatakse ELAN-i uuendatud hierarhilist kihistruktuuri, mille on välja arendanud Freiburg Research Group in Saami Studies töötades nii saami keelte kui ka komi keele teadusprojektidega. Artiklis arutletakse selle üle, kuivõrd hierarhiline kihistruktuur ja ELAN-i otsinguliides sobivad korpuslingvistilise uurimistöö jaoks. Ilmneb, et ELAN-i otsimootor võimaldab teha lihtsamaid päringuid, kuid keerulisemad otsingud on raskendatud. Programmi „Multiple Layer Search" töörežiimis esineb fundamentaalne puudus, mistõttu komplekssete korpusuuringute jaoks seda tõenäoliselt kasutada ei saa.

Proceedings of the fifth Workshop on Computational Linguistics for Uralic Languages, pages 90–103,
Tartu, Estonia, January 7 - January 8, 2019.

1 Overview of ELAN search functionalities

ELAN is a multimedia language annotation program which enables textual annotation of audio and/or video media files within a single application. It is free software developed by the Technical Group of the Max Planck Institute for Psycholinguistics, and can be downloaded from `https://tla.mpi.nl/tools/tla-tools/elan`. ELAN was created with linguists who work with non-text-based linguistic data as the main target user group, and continues to be developed with them in mind. ELAN annotation files are plain text files in xml format with the file extension `.eaf`. They are fully compatible with the unicode standard. Because they are in xml format, they can be accessed using other protocols, and even automatically generated.

The general functionality of ELAN is described in detail in the ELAN manual (available from the ELAN website), in various training materials for documentary and corpus linguistics, and in a few scientific publications; for instance, see § 4 in Gerstenberger et al. (2016) for a general description, and Nagy and Meyerhoff (2015) for a detailed example of an ELAN implementation for sociolinguistic research. With these publications in mind, the present discussion will be limited to those aspects of ELAN which are directly relevant to its use as a corpus search engine.

Annotations in ELAN are time-aligned with a media file or files, and are organized into layers called "tiers" which can be defined on an individual basis; typically, each tier corresponds to the specific type of information it contains (e.g., orthographic transcription, meta-language translation, etc.). The information provided in the annotations must be represented as a string of characters, but ELAN provides neither restrictions nor suggestions concerning the type of content annotations contain; as a result, every user or project must come up with a set of relevant tiers. Tiers can be structured hierarchically, such that one tier is subordinate to another tier, e.g., a Russian translation may be under a tier containing a target language transcription. The hierarchical relationship between superordinate and subordinate tiers is governed by "Tier Types"[1] which essentially define how tiers are organized with respect to the timeline and within the hierarchy. Having hierarchically structured tiers allows ELAN searches to be more targeted, and thus more powerful, than when no tier hierarchy is present because it is therefore possible to limit the scope of a search to specific inter-tier relationships; this is illustrated below in section 3.3. Note that a typical ELAN annotation file is structured so that each participant in the annotated linguistic event has his/her own set of tiers using the same hierarchy. This makes it possible for ELAN to deal with overlapping speech, a typical characteristic of spoken language.

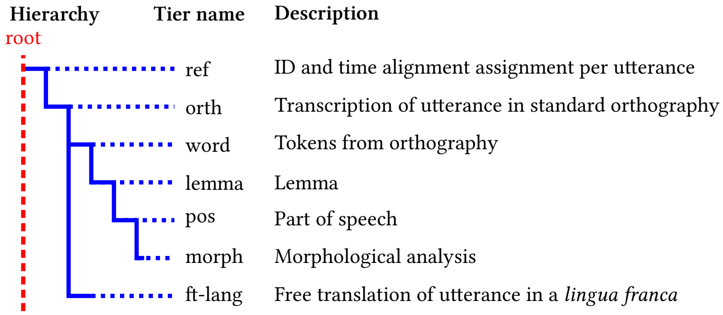
In the following section (section 2), I briefly present a specific implementation of ELAN as a corpus collection tool in order to later illustrate how ELAN searches can be performed. After that, the various search functions built into ELAN are summarized in section 3, including examples for how these can be used to search hierarchical tier structures (as illustrated by the Freiburg template). Finally, in section 4, I describe a significant problem concerning how to limit the scope of search criteria found in the "Multiple Layer Search" function, and discuss why this likely prevents ELAN from

[1] In older versions of ELAN, these were referred to as 'Linguistic Types'.

ultimately being used as corpus search engine for complex queries. A summary and conclusions are found in section 5.

2 An example tier structure

Although tiers do not necessarily have to be organized hierarchically in ELAN, searches in ELAN can be more powerful if a meaningful tier hierarchy is present. In order to understand how ELAN can be used as a search tool, it is useful to provide an example for how ELAN annotation tiers can be organized hierarchically. In this section, I provide an overview of the ELAN tier hierarchy standard as developed and implemented in various projects on Saami languages and Komi variants carried out within the auspices of the Freiburg Research Group in Saami Studies. Note that this structure is only one possible template, and is provided here simply as an illustration; indeed, ELAN allows users to define any kind of hierarchy structure (including a flat structure).

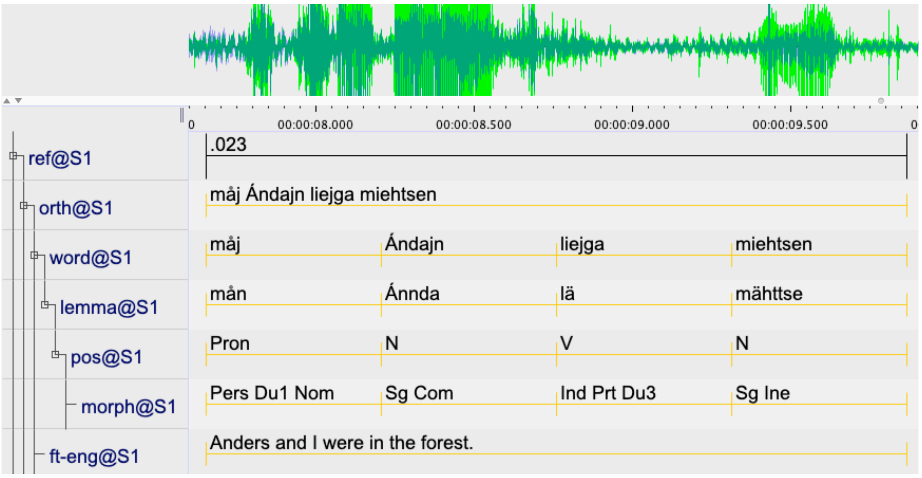


Hierarchy	Tier name	Description
	ref	ID and time alignment assignment per utterance
	orth	Transcription of utterance in standard orthography
	word	Tokens from orthography
	lemma	Lemma
	pos	Part of speech
	morph	Morphological analysis
	ft-lang	Free translation of utterance in a *lingua franca*

Figure 1: The minimal ELAN annotation tier hierarchy template used in the Freiburg Research Group in Saami Studies' corpora

ELAN annotation tiers used in the Freiburg projects are organized hierarchically using the minimal template shown in Figure 1 for each individual participant in a text.[2] Time-alignment relative to the original media file (usually at least a .wav-file, often with accompanying video) is set in the root node tier named `ref`, which also serves to assign the utterance a unique number within the text; this is the only tier in the hierarchy which is linked directly to the time line (as opposed to being symbolically linked via another tier). The `orth` tier contains an orthographic representation of the utterance at hand; there is one and only one `orth` tier for each `ref` tier, and, due to its tier type, it time-aligns exactly with its superordinate `ref` tier. The `word` tier contains individual annotations for each token in the `orth` tier. Each token in the `word` tier is assigned a lemma in the subordinate `lemma` tier. The part of speech for each lemma is presented in the `pos` tier. When applicable, relevant morphological values for the specific wordform found in the token are presented in the `morph` tier, which completes

[2] A more thorough, dynamic description of the Freiburg tier structure can be found at `https://github.com/langdoc/FRechdoc/wiki/ELAN-tiers`, including an inventory of the tier types used. An older version of the hierarchy is presented in Gerstenberger et al. (2016, 37-38).



ref@S1	.023			
orth@S1	måj Ándajn liejga miehtsen			
word@S1	måj	Ándajn	liejga	miehtsen
lemma@S1	mån	Ánnda	lä	mähttse
pos@S1	Pron	N	V	N
morph@S1	Pers Du1 Nom	Sg Com	Ind Prt Du3	Sg Ine
ft-eng@S1	Anders and I were in the forest.			

Figure 2: A screenshot presenting an implementation of the hierarchical tier structure for an utterance annotated in ELAN

the grammatical annotations. Finally, the `ft−lang` tier provides a free translation of the utterance in a specific *lingua franca* (here, the iso-639 code is used in place of 'lang', e.g., the tier `ft−eng` is for a free English translation).

A screenshot is provided in figure 2 to show what the implementation of this actually looks like in an ELAN annotation file. Here, the hierarchical tier structure is on the left, and the wave file is at the top; the utterance itself, here numbered ".023", and the corresponding annotations are shown in the rest of the image. Each participant has the same set of tiers, but each tier name is extended by a "domain" name identifying the speaker (formatted much like an email address); in the example in figure 2, the first speaker is simply identified as "S1", and thus all of this speaker's tiers are modified with the extension "@S1", as in `ref@S1`, `orth@S1`, `word@S1`, etc. Aside from being a clear way to mark the speaker for a specific annotation, naming tiers this way allows ELAN search queries to also be limited to a specific tier for a specific speaker, but across the corpus.

Other, project- or text-specific tiers may also exist, and these are located at the relevant level of the hierarchy.[3] In the Freiburg corpora, all annotations from the `word` tier through the `morph` tier are created automatically (using a python script) from the output that results from feeding the orthographic representation in the `orth` tier through Finite State Transducer and Constraint Grammar implementations.[4]

[3] Examples of other tiers found in some of the Freiburg corpora include an `orth−orig` tier containing older orthographic transcriptions of a text and subordinate to the `ref` tier, or a `gloss` tier presenting rough translations of each lemma and subordinate to the `pos` tier.

[4] See Blokland et al. (2015); Gerstenberger et al. (2016, 2017b,a) for discussions of various aspects of this approach, including how ambiguous analyses are handled.

3 ELAN as a corpus search engine

Typically, a single ELAN file contains annotations for a single recorded linguistic event, and corresponds to one or more audio or video files.[5] An ELAN corpus thus consists of all ELAN annotation files corresponding to the texts considered to be part of the corpus.

The ability to search within a single ELAN file when it is currently open is impressively powerful, and includes the ability to limit the search to specific tiers, to use regular expressions, to replace all hits with a different string, and to recursively perform searches limited to the results of a previous search. However, since this discussion is interested in ELAN as a *corpus* search engine, this functionality will not be discussed here in any further detail. Instead, search functions that can be applied simultaneously to multiple ELAN files (i.e., an ELAN corpus) will be described and reviewed below.

In order to perform a corpus search, one first has to choose the set of ELAN files to be considered.[6] These can be selected either one by one, or users can choose all the ELAN files in a certain path, or a domain can be constructed of ELAN files sharing specific metadata characteristics (the last option requires having metadata for each file in IMDI[7] format). Once the files have been selected to comprise the corpus, searching can commence.

3.1 Basic search modes

The results of searches using either of the menu items "Search Multiple eaf" or "FAST-Search" are listed in concordance format, including information such as file name, tier name, etc., and with the preceding and following annotations shown to provide immediate context. Regular expressions[8] can be used in this interface, case-sensitivity can be set, and the results can be exported into tab delimited format. However, that is the extent of the functionality of this type of search; as such, it is useful to get a quick, impressionistic result set, but it is not sufficient for more complex, specific corpus searches, and thus is rather insignificant for corpus linguistics and computational linguistics, and will not be discussed further here.

Choosing the menu item "Structured Search Multiple eaf" opens a search interface window with three types of searches which increase in complexity from left to right. A "Substring Search" is similar to the "Search Multiple eaf" functionality outlined in the previous paragraph, but without even the regular expression or case-sensitivity options. However, search results in this mode can be presented in multiple ways: as a concordance, as a list of frequencies, or as found in the individual ELAN files, including time-alignment, file name, tier name and tier type; these types of results can be saved in tab separated value format. As with "Search Multiple eaf", this search

[5] Note that it is not obligatory to have a media file; it is thus also possible to use ELAN to annotate exclusively written sources, such as heritage texts.

[6] In the ELAN interface, this set of files is referred to as the "domain".

[7] Cf. `https://tla.mpi.nl/imdi-metadata/`.

[8] These are based on regular expressions in java, cf. `https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html`.

is quite superficial for essentially the same reasons, and it is not clear why these two types of search exist as separate entities.

3.2 Complex corpus search modes

The other two search modes are called “Single Layer Search” and “Multiple Layer Search”. These are significantly more powerful concerning many aspects of the search criteria accepted, from mere convenience features to significantly increased query precision. The existence of these modes is what allows the ELAN search functionality to even be considered a potentially useful corpus search engine. The difference between these two search modes is found in the complexity of queries concerning the features of tiers which can be referenced; this will be further examined below. But to begin with, their common functionalities will be specified.

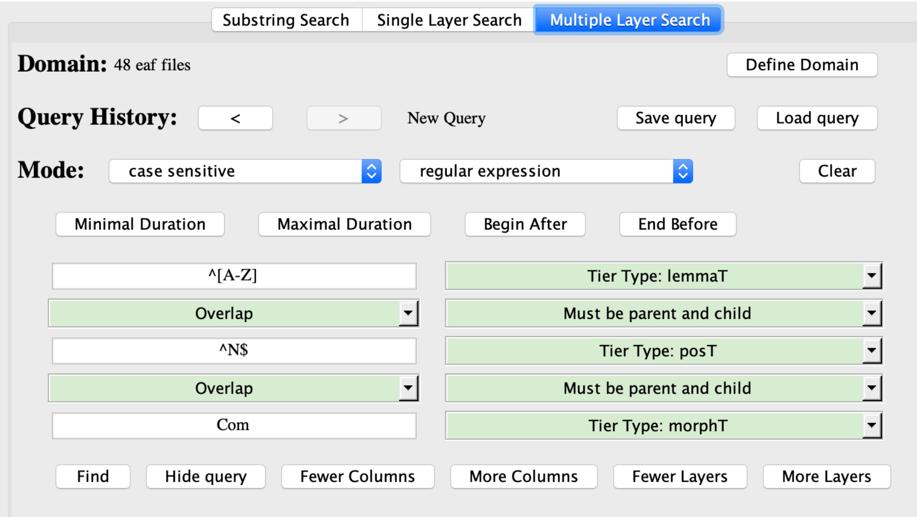
Search queries in these two modes can be saved and loaded again later, which allows for increased ease of reproducibility. There are < and > buttons for conveniently ‘browsing’ between previously entered search queries (essentially like those found in internet browsers). In the basic annotation mode, one can further specify a query for character matches (either substrings or white-space separated units (the latter are known as “exact matches” in ELAN)), or one using regular expressions. Furthermore, the search scope can be set to all extant tiers in the corpus, to a subset defined either by a specific tier name, or all tiers with a common tier type, or finally, to all tiers corresponding to a specific participant. However, these searches are limited to a single tier name, a single tier type or a single participant; no complex subset of various tier names, or multiple participants, etc. is possible. Therefore, any more specific restriction on the structural scope of a search query (i.e., filtering any type of information not directly included in the actual annotations, e.g., speaker gender, age, etc.) must be done in either pre-processing (by defining the corpus for the ELAN search), or in post-processing results outside of ELAN.

Search results for both modes can be displayed as a concordance, as a frequency table, or as individual annotations.[9] Results from each of these ways of organizing hits can be stored as a tab-separated value file. This allows search results to be exported for further processing elsewhere, if desired.

Generally speaking, a “Single Layer Search” is useful because of the characteristics detailed above, but defining the scope of the search is limited (as the name implies). With this in mind, the “Multiple Layer Search” mode is the focus of the rest of this discussion because it presents the only opportunity to perform complex search queries across the corpus while taking advantage of the hierarchical structure of tiers. Figure 3 provides a screen shot of a relatively simple multiple layer search query which restricts the search scope to the hierarchical limitations of a single column. This image serves to illustrate the basic idea behind multiple layer searches in ELAN. Note that users need to be thoroughly familiar with the tier hierarchy of the ELAN files in the search domain to use and take full advantage of the Multiple Layer Search.

Here, a case-sensitive regular expression search looking vertically through the

[9] These are discussed in more detail below, and illustrated there by screenshots in figures 4, 5 and 6, respectively.



Domain: 48 eaf files

Query History: New Query

Mode: case sensitive | regular expression

^[A-Z]	Tier Type: lemmaT
Overlap	Must be parent and child
^N$	Tier Type: posT
Overlap	Must be parent and child
Com	Tier Type: morphT

Figure 3: An example of a "Multiple Layer Search" with a search query for a single column, looking for proper names in comitative case

hierarchy is defined.[10] In the column on the left, the search criteria themselves are entered in the white fields, while the temporal relationship between the layers are set in the green drop-down menu boxes. In the column on the right, the search criteria setting the scope of search for each of the white search-criteria boxes is defined, as is a further hierarchical relationship between the layers to be searched. In this example, the search is intended to find all hits of proper names in comitative case.

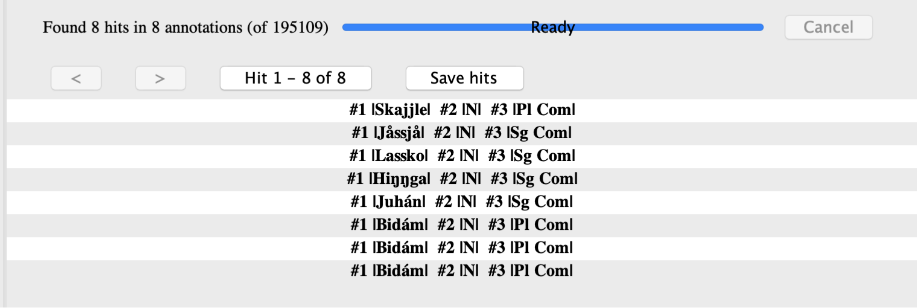
The uppermost layer is set on the right to look only at tiers with the type `lemmaT`, which in the Freiburg hierarchy[11] selects only `lemma` tiers, and on the left to look for lemma annotations that begin with a capital letter using the regular expression `^[A-Z]`.

The middle layer is set on the right to look only at tiers with the type `posT`, which in the Freiburg hierarchy selects only `pos` tiers, and only when a specific annotation is in a "parent and child" hierarchical relationship to the uppermost level; in other words, ELAN is set to only find hits on the `pos` tier which are directly subordinate to a `lemma` tier. Similarly, the middle layer is set on the left side to look for annotations that consist solely of the character `N` (used to signify 'noun') using the regular expression `^N$`.

Finally, the lowest layer is set on the right to look only at tiers with the type `morphT`, which in the Freiburg hierarchy selects only `morph` tiers, and only when a specific annotation is in a "parent and child" hierarchical relationship to the middle

[10] Time restrictions on the duration or location within the recording can be set using the "Minimal Duration", "Maximal Duration", "Begin After" and "End Before" buttons, but are not used in this example. Indeed, for the type of searches looking for lexical or grammatical structures that the author uses, these are not relevant at all.

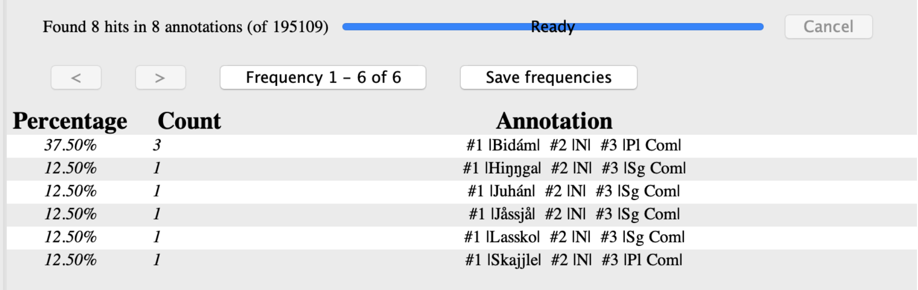
[11] Cf. section 3.3.



Found 8 hits in 8 annotations (of 195109)

#1 |Skajjle| #2 |N| #3 |Pl Com|
#1 |Jåssjål #2 |N| #3 |Sg Com|
#1 |Lassko| #2 |N| #3 |Sg Com|
#1 |Hiŋŋga| #2 |N| #3 |Sg Com|
#1 |Juhán| #2 |N| #3 |Sg Com|
#1 |Bidám| #2 |N| #3 |Pl Com|
#1 |Bidám| #2 |N| #3 |Pl Com|
#1 |Bidám| #2 |N| #3 |Pl Com|

Figure 4: An example of search results presented in "Concordance view"



Found 8 hits in 8 annotations (of 195109)

<table>
<tr><th>Percentage</th><th>Count</th><th>Annotation</th></tr>
<tr><td><i>37.50%</i></td><td><i>3</i></td><td>#1 |Bidám| #2 |N| #3 |Pl Com|</td></tr>
<tr><td><i>12.50%</i></td><td><i>1</i></td><td>#1 |Hiŋŋga| #2 |N| #3 |Sg Com|</td></tr>
<tr><td><i>12.50%</i></td><td><i>1</i></td><td>#1 |Juhán| #2 |N| #3 |Sg Com|</td></tr>
<tr><td><i>12.50%</i></td><td><i>1</i></td><td>#1 |Jåssjål #2 |N| #3 |Sg Com|</td></tr>
<tr><td><i>12.50%</i></td><td><i>1</i></td><td>#1 |Lassko| #2 |N| #3 |Sg Com|</td></tr>
<tr><td><i>12.50%</i></td><td><i>1</i></td><td>#1 |Skajjle| #2 |N| #3 |Pl Com|</td></tr>
</table>

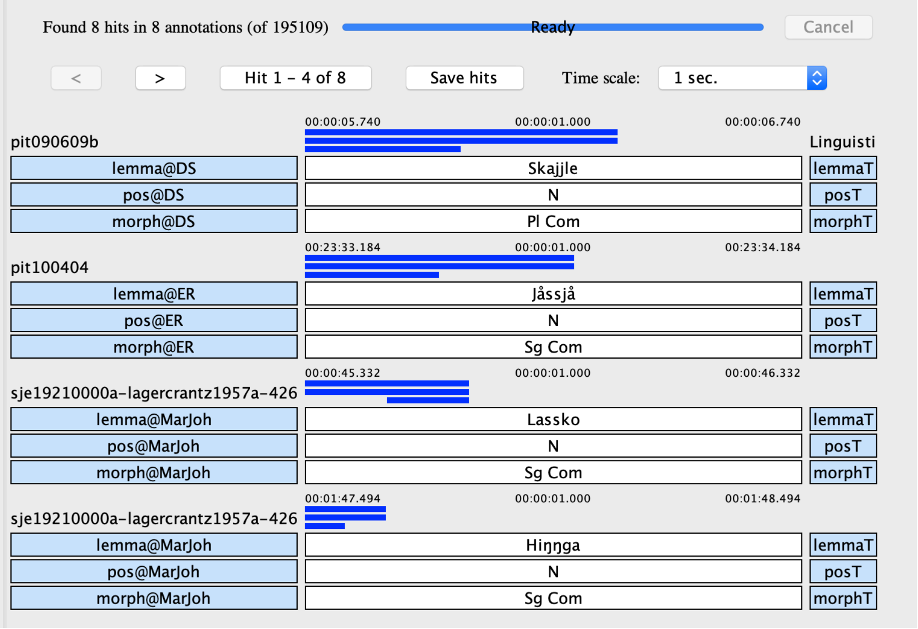
Figure 5: An example of search results presented in "Frequency view"

level; in other words, here ELAN limits hits to those on the `morph` tier which are directly subordinate to a `pos` tier. Similarly, the lowest layer is set on the left side to look for annotations that contain the character string `Com` (used to tag wordforms in comitative case).

Resulting hits can be viewed in three ways: 1) as a concordance (cf. figure 4); 2) listed by frequency, and further arrangeable by frequency (from most to least) or alphabetically by annotation (cf. figure 5); as well as in 3) "Alignment view" showing each hit as found in the respective set of annotations and time-aligned (cf. figure 6). Clicking on a hit automatically opens the corresponding ELAN file to the specific place where the hit is found. This makes it very easy to go to a specific spot in the corpus to further inspect a hit in its actual context.

In addition to being able to search vertically within a tier hierarchy, the "Multiple Layer Search" also allows one to search horizontally by specifying search criteria that look at annotations to the left or right on a specific tier. This is done by adding additional columns in the search interface, as illustrated by the screenshot in figure 7.[12] This idea is essentially the same as with the single column search presented above (cf. figure 3), but here, the horizontal distance between annotations which fulfill the

[12] Note that columns and layers can be added or taken away, depending on the specific search query, using the "Fewer Columns" and "More Columns" or "Fewer Layers" and "More Layers" buttons; a maximum of eight columns and eight layers can be used.



Found 8 hits in 8 annotations (of 195109)

Time scale: 1 sec.

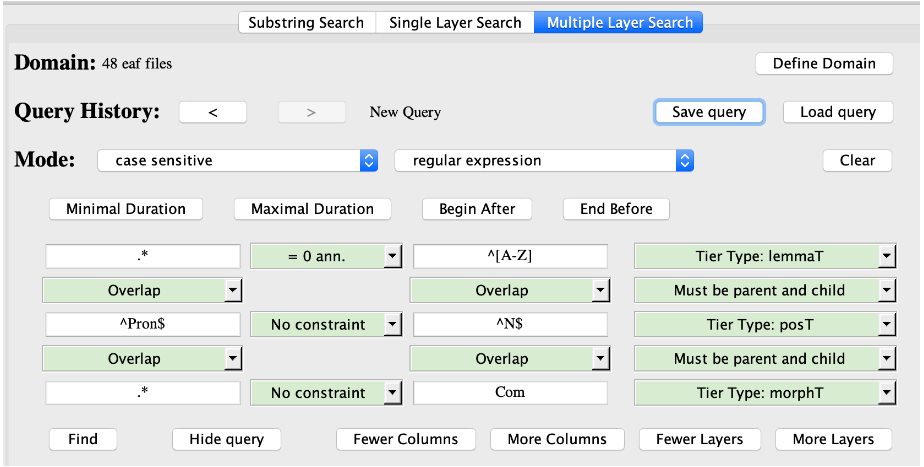
pit090609b	00:00:05.740 00:00:01.000 00:00:06.740	Linguisti
lemma@DS	Skajjle	lemmaT
pos@DS	N	posT
morph@DS	Pl Com	morphT

pit100404	00:23:33.184 00:00:01.000 00:23:34.184	
lemma@ER	Jåssjå	lemmaT
pos@ER	N	posT
morph@ER	Sg Com	morphT

sje19210000a–lagercrantz1957a–426	00:00:45.332 00:00:01.000 00:00:46.332	
lemma@MarJoh	Lassko	lemmaT
pos@MarJoh	N	posT
morph@MarJoh	Sg Com	morphT

sje19210000a–lagercrantz1957a–426	00:01:47.494 00:00:01.000 00:01:48.494	
lemma@MarJoh	Hiŋnga	lemmaT
pos@MarJoh	N	posT
morph@MarJoh	Sg Com	morphT

Figure 6: An example of search results presented in "Alignment view"



Domain: 48 eaf files

Query History: New Query

Mode: case sensitive regular expression

.*	= 0 ann.	^[A-Z]	Tier Type: lemmaT
Overlap		Overlap	Must be parent and child
^Pron$	No constraint	^N$	Tier Type: posT
Overlap		Overlap	Must be parent and child
.*	No constraint	Com	Tier Type: morphT

Figure 7: An example of a "Multiple Layer Search" with a search query for two columns, looking for proper names in comitative case immediately preceded by a pronoun

search criteria can be set, and is measured either in the number of intervening annotations or in milliseconds. The minimum setting is zero, i.e., no annotations or no milliseconds between neighboring annotations with hits. Note, however, that this entails that any given hit *must* consist of at least two separate annotations throughout the respective hierarchies which return the hits; thus any given search result *is not able* to refer to the same individual annotation in more than one part of the respective sub-hit's vertical hierarchy. This is a significant weakness of ELAN searches using hierarchical tier structures that is discussed at the end of section 3.3 below. Aside from this additional horizontal operation, the search interface is the same as presented above for the "Single Layer Search".

3.3 Searching Freiburg-style ELAN corpora

It is hopefully obvious from the description in section 2 above that the hierarchical tier structure developed for corpora in the language documentation projects carried out by the Freiburg Research Group in Saami Studies is intended to take advantage of two functionalities of ELAN searches. First, we distinguish structurally between different types of information by restricting tiers to contain only specific types of information. Thus, the orthographic representation is saved in the `orth` tier, an English translation is in the `ft-eng` tier, etc. For linguistic annotations, individual tokens[13] are in the `word` tier, the corresponding lemma is in the `lemma` tier, the part of speech in the `pos` tier, and relevant values of morphological categories are in the `morph` tier. Second, these types of information are symbolically linked to each other by structuring the tiers into a hierarchy. In this way, any given annotation in the `ref` tier is the root node and time-aligned to the master media file; annotations on this tier consist only of a unique and symbolic identifier (a number), while all other relevant annotations are subordinate to this main, time-aligned annotation. Tokens from the `orth` tier are found as individual annotations in the `word` tier, the corresponding lemma is subordinate to each token, the part of speech is subordinate to the lemma, and morphological information is subordinate to the part of speech (cf. figures 1 and 2).

In addition to being a transparent, well structured presentation format, the idea behind structuring annotations in this way is to increase efficiency in searching. As is probably obvious, grouping types of information separately allows one to more easily limit search results to a specific type, or filter out unwanted hits; a simple example for this would be restricting the result set to only include nouns by looking only for hits with 'N' in the `pos` tier. By combining this type of specific searches restricting results to specific hits on more than one tier in the search interface, searches in ELAN can, theoretically, be quite specific, without having exceptionally complicated regular expression statements that would otherwise be required in a flat tier structure. An example of this is provided above in figure 3), where proper nouns in comitative case are the target of the search criteria; here, detailed search criteria on three levels of the tier hierarchy are specified.

[13] As it is consistent with the Giellatekno preprocessing scripts, we treat punctuation as tokens. However, note that particularly spoken language corpora are not consistently annotated using punctuation to mark the end of utterances, so punctuation characters are not a reliable tool to find utterance boundaries.

4 A fundamental problem of scope restriction

Because the information which the Freiburg-style annotations contain are of a lexicogrammatical nature, as well as due to the hierarchical tier structure, the Freiburg corpora are intended to be particularly useful for searching for morphosyntactic, syntactic or discourse syntactic patterns. However, the ELAN multiple layer search interface has a significant flaw that prevents it from being the powerful corpus search engine it appears to be on the surface, both for Freiburg-style tier hierarchies and likely for any hierarchically structured ELAN file. This flaw stems from the fact that it is impossible to restrict search criteria in two or more columns lower in the hierarchy to fall within one and the same higher-level parent (or grandparent, great-grandparent, etc.) tier.[14]

This is best illustrated with an example. Say for instance you want to find all utterances which have a dual pronoun followed by a singular noun in comitative case (for example in searching for instances of comitative coordination (cf. Morottaja et al., 2017)). For the left column, the morphological search criteria (in the `morph` tier) would be `Du` to find hits marked for dual, and for the right column `Sg Com` to find hits for "comitative singular". For the `pos` tier, the left column would be set for `Pron` for "pronoun", and the right column for `^N$` for nouns.[15] But there is no way to restrict hits to be *within* a single superordinate tier (such as the `orth` tier), and thus even hits which cross annotations boundaries on the `orth` tier will be included in the result set. It is possible to set the search to be limited to directly neighboring annotations (i.e., two annotations which do not have other annotations in between; in the ELAN search interface, this corresponds to "`= 0 ann.`"), but even this does not exclude hits with an intervening annotation boundary in a superordinate tier. Thus the Pite Saami example in (1) would correctly produce the hit *måjå Ándajn.* However, if the examples in (2) and (3) are neighboring annotations, they would *also* produce the hit *måjå Ándajn*, even though these are two *separate* utterances.[16]

(1)	*måjå*	*Ándajn*	*lijmen*	*miehtsen*
	PERS.1DU.NOM	Anders-COM.SG	be-1DU.PRT	forest-INESS.SG

'Anders and I were in the woods'

(2)	*dä*	*buhtin*	*måjå*
	then	come-1DU.PRS	PERS.1DU.NOM

'Then we (two) came'

[14] Note that ELAN has a search mode in the "Multiple Layer Search" interface which one could potentially expect to be able to deal with this: the "variable match" mode. However, variables cannot be self-referential, so the higher-level matches must be separate, unique annotations which are identical in form, so this would not work. On top of that, regular expressions are not allowed in this mode, so it would not be particularly powerful or useful in any case.

[15] This regular expression is necessary because the set of abbreviations for parts of speech includes "Num" for numerals, and a search simply for `N` would include numerals as well.

[16] Note that particularly spoken language corpora are not consistently annotated using punctuation to mark the end of utterances, so punctuation is not a reliable tool to be used for ruling out such hits as in the second half of this example.

(3)	*Ándajn*	*lä*	*állkep*
	Anders-COM.SG	be\3SG.PRS	easy-COMP.NOM.SG

‘It’s easier with Anders’

Note that one could think that a search which looks for directly neighboring hits (such as in the examples above, which look for a dual pronoun directly followed by a singular noun in comitative case) could get around this flaw by setting the constraint concerning number of annotations allowed to intervene between hits to “0”. However, this still does not avoid getting hits such as the one arising from examples 2 and 3, as illustrated above, since the scope still cannot be set to take higher-level annotation boundaries into account. Furthermore, in the current state of the Freiburg workflow and infrastructure, this sort of restriction is useless as well because each ambiguity which is not removed by constraint grammar rules is automatically added as a unique annotation to an ELAN file in random order. Thus, it is feasible that another possible analysis (arising from ambiguous morphological surface forms) may occur between the correct form itself and a following annotation, but such actual hits would not be output to the search results if the intervening number of annotations is set to “0”. For instance, since Pite Saami comitative singular and inessive plural forms are always homophonous, if the constraint grammar rules are not able to disambiguate, both possible analyses will be written as annotations in the ELAN file, as in the double gloss for *Ándajn* in example 4 below. If no annotations are allowed to occur between the hits, then this entirely relevant hit will not be found.

(4)	*måjå*	*Ándajn*	*lijmen*	*miehtsen*
	PERS.1DU.NOM	Anders-**INESS.PL/COM.SG**	be-1DU.PRT	forest-INESS.SG

‘Anders and I were in the woods’

On the other hand, if no constraint is set, then *any* and *every* possible co-occurance of the two criteria throughout the entirety of any given ELAN file will be found. In other words, given an ELAN file with a hundred utterance annotations, an instance of *måjå* in the first annotation and an instance of *Ándajn* in the hundredth annotation will also be returned as a hit.

It could potentially be claimed that this is not a flaw in the ELAN interface, but instead an unsuitable hierarchical tier structure developed by the Freiburg group. Perhaps a different tier structure would allow for better searching, but the fundamental problem that a higher-level annotation cannot be set as the scope of a search query still exists. This calls into question whether a hierarchical tier structure consisting of annotations with lexico-grammatical information is even a useful construction, aside from its clear benefits of being a functional storage format and an elegant presentation format. Indeed, one current work-around for doing complex searches in ELAN which are limited in scope to looking within – and not across – higher-level annotation boundaries (specifically those of the `orth` tier) involves a flat structure consisting of the utterance-level annotations each containing an utterance’s entire FST/CG[17] out-

[17] Finite State Transducer and Constraint Grammar

put. For such search queries, it is sufficient to use complex regular expressions in the "Single Layer Search" mode of the ELAN search interface.[18]

5 Conclusion

In summary, ELAN presents a complex way of handling linguistic annotations, including the ability to differentiate between types of information by using an annotation-tier hierarchy. With this in mind, the Freiburg Research Group in Saami Studies has developed such a hierarchy for annotating lexico-grammatical features such as lemma, part of speech and morphological information for the group's various, mainly spoken-language corpora for endangered Uralic languages. It is clear that, as an annotation and presentation tool, ELAN is very useful; this paper has attempted to explore the functionality of ELAN as a corpus search engine using the complex hierarchical tier structure developed by the Freiburg group to illustrate this.

ELAN offers various levels of complexity in its search capabilities. The most complex of these, the "Multiple Layer Search", includes the ability to stipulate search criteria both vertically within the hierarchy on a tier-by-tier level, and horizontally across annotations. Despite this complex-looking search interface, it has a significant weakness which makes it insufficient for complex corpus queries looking for morphosyntactic or syntactic patterns. Specifically, it is not possible to limit the scope of a search to take utterance-level boundaries into account. Thus, even false hits which contain one or more utterance-level boundaries will always be returned. With this weakness in mind, ELAN is not an ideal corpus search tool. Fortunately, ELAN search results can be exported to other open formats such as tab separated files, which can then be further refined using other utilities.

References

Rogier Blokland, Ciprian Gerstenberger, Marina Fedina, Niko Partanen, Michael Rießler, and Joshua Wilbur. 2015. Language documentation meets language technology. In Tommi A. Pirinen, Francis M. Tyers, and Trond Trosterud, editors, *First International Workshop on Computational Linguistics for Uralic Languages, 16th January, 2015, Tromsø, Norway*, The University Library of Tromsø, number 2015:2 in Septentrio Conference Series, pages 8–18.

Ciprian Gerstenberger, Niko Partanen, and Michael Rießler. 2017a. Instant annotations in ELAN corpora of spoken and written Komi, an endangered language of the Barents Sea region. In Antti Arppe, Jeff Good, Mans Hulden, Jordan Lachler, Alexis Palmer, and Lane Schwartz, editors, *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, Association for Computational Linguistics, ACL Anthology, pages 57–66.

[18] To be fair, it should be noted that if the data set is small enough, search results can be gone through by hand, eliminating those which should not in fact be included. But from a computational linguistics point of view, this is obviously not a tenable solution, especially when dealing with massive data sets ('big data').

Ciprian Gerstenberger, Niko Partanen, Michael Rießler, and Joshua Wilbur. 2016. Utilizing language technology in the documentation of endangered Uralic languages 4:29–47.

Ciprian Gerstenberger, Niko Partanen, Michael Rießler, and Joshua Wilbur. 2017b. Instant annotations. Association for Computational Linguistics, ACL Anthology, pages 25–36.

Petter Morottaja, Raj Singh, and Ida Toivonen. 2017. Comitative coordination in inari saami. Presentation at the 3rd Saami Linguistics Symposium, Albrecht-Ludwigs-Universität Freiburg, 19-20 October 2017.

Naomi Nagy and Miriam Meyerhoff. 2015. Extending ELAN into variationist sociolinguistics. *Linguistics Vanguard* 1(1):271–281.