



HAL
open science

Searching and retrieving multi-levels annotated data

Brigitte Bigi, Jorane Saubesty

► **To cite this version:**

Brigitte Bigi, Jorane Saubesty. Searching and retrieving multi-levels annotated data. Proceedings of Gesture and Speech in Interaction - 4th edition, 2015, Nantes, France. pp.31-36. <hal-01455315>

HAL Id: hal-01455315

<https://hal.science/hal-01455315v1>

Submitted on 22 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Searching and retrieving multi-levels annotated data

Brigitte Bigi¹, Jorane Saubesty¹

¹Laboratoire Parole et Langage, CNRS, Aix-Marseille Université
5, avenue Pasteur, 13100 Aix-en-Provence, France

brigitte.bigi@lpl-aix.fr, jorane.saubesty@blri.fr

Abstract

The annotation of recordings is related to many Linguistics subfields as Phonetics, Prosody, Gestures or Discourse... Corpora are annotated with detailed information at various linguistic levels thanks to annotation software. As large multimodal corpora become prevalent, new annotation and analysis requirements are emerging. This paper addresses the problem of exploring annotations in order to extract multimodal data in the linguistic field ranging from general linguistic to domain specific information. The answer chosen to fulfill this purpose is a user-oriented approach: the data can be extracted without any specific knowledge or skill. The paper exposes two ways to filter the annotations by a predicative approach: 1/ single filters, i.e. search in one tier depending of the data content, by the extraction of the time values and the duration; 2/ relation filters, i.e. search on annotations of a tier in time-relation with annotations of another one. This system is distributed in SPPAS software, under the terms of a public license.

Index Terms: software, multi-levels annotations, filtering

1. Introduction

When people communicate: gestures, postural shifts, facial expression, backchannel continuers such as “mm-hmm”, spoken turns and many more, all together work in concert to bring about mutual understanding. Annotating recordings of people communicating may therefore involve many Linguistics subfields such as Phonetics, Prosody, Gestures or Discourse... As a consequence, the last ten years or so have witnessed a real increase of linguistic annotated data. Whereas few years ago it was common to formulate linguistic models on the basis of rather limited data, today it is becoming more and more expected for linguists to take into account large quantities of empirical data, often including several hours of recordings. As a consequence, a number of software for the manual annotation of audio and/or video recordings have become available, such as *Anvil* [1], *Elan* [2], *Praat* [3], *Transcriber* [4] or *Exmaralda* [5], to name just some of the most popular tools, all of which are both free and multi-platform. Furthermore, linguists need tools for the automatic annotation, including the alignment of the speech recording with a phonetic transcription of the speech, as SPPAS [6].

As large multimodal corpora become prevalent, new analysis requirements emerge. Multimodal analysis has become a crucial part of research, teaching and practice for a wide range of academic and practical disciplines. The difficulties of multimodal analysis are visible in most of the works that explore this field. Multimodal annotation requires the possibility to encode many different information types, from different domains, with different levels of granularity [7].

“Corpora that include time-based data, such as video and marking gestures, make annotation and analysis of language

and behavior much more complex than analysis based solely on text corpora and an audio signal” [8]. Thus, nowadays one of the biggest barriers with which the linguists must cope, is not the storage of data, nor its annotation, but rather *its exploration*. In addition to annotation, some tools provide statistical analysis capabilities. A minimum capability required is to search for annotated entities and their relationships [8]. Generally, different annotation tools are designed and used to annotate the audio and video contents of a corpus that can later be merged in query systems or databases [9]. With the help of multimodal corpora searches, the investigation of the temporal alignment (synchronized co-occurrence, overlap or consecutivity) of gesture and speech has become possible [9]. “Obviously, the *raison d’être* of annotation in general is to allow linguists to retrieve all and only all instances of a particular phenomenon” [10].

The question of multi-levels filtering for linguistic annotated resources covers different aspects. It firstly requires a representation framework making it possible to compare, and eventually merge, different annotation schemes from different annotation tools. The basic structures of speech/video annotated data are “tiers” or “tracks” of annotations. Thus, speech/video annotation tools rely on this formalism because the *Tier* representation is appropriate to any multimodal annotated data given its genericity and flexibility and that it simply maps the annotations on the timeline. In the context of such tools, a *Tier* is a series of *Annotation* instances, each one defined by a temporal localization (an interval or a point) and a label. Obviously, due to the diversity of linguistic phenomena, annotation tools lead to a variety of models, theories and formalisms. This diversity results in heterogeneous description formats, each tool developing its own framework. Then, even if some are compatible, none of the annotation tools are directly interoperable, each one using a native format, some of them on top of XML, some others developing an *ad hoc* markup language. The heterogeneity of such annotations has been recognized as a key problem limiting the interoperability and re-usability of Natural Language Processing tools and linguistic data collections.

This paper focuses on the problem of *searching and retrieving data from multi-levels annotated corpora*. After a review of the main tools allowing to build queries in a multimodal annotated corpus, this paper presents the specifications of a software development according to eight criteria it must respect. The system proposed in this paper is a component named *DataFilter* in SPPAS software [6], described in Section 3. The method to search and retrieve data is based on a predicative approach allowing the definition of 2 types of filters: 1/ single filters, i.e. search in one tier depending of the data content, by the extraction of the time values or the duration (Section 4); 2/ relation filters, i.e. search on annotations of a tier in time-relation with annotations of another one (Section 5). Finally, Section 6 shows with a concrete study the benefit of the proposed software.

2. Background and motivations

A query is a request for a subset of all annotation elements, given some constraint. A query language (QL) is a programming language allowing to write queries. In the context of extracting multi-levels annotated data, multi-levels annotations can quickly become cluttered, so that the user needs query functionality to efficiently find relevant information. The following explores some popular and freely available tools.

Praat allows to paint intervals in green color, labels matching a given pattern with one of the following criteria: is equal to, is not equal to, contains, does not contain, starts with, does not start with, ends with, does not end with, matches a regular expression.

EXAKT (EXMARaLDA Analysis- and Concordance Tool) is the query and analysis tool for EXMARaLDA corpora, and can also be used for corpora created with other tools as Transcriber or Elan. Labels of annotations can be searched in the corpus using regular expressions. It allows to save query results (HTML, text) or export them to other applications (e.g. Excel).

Elan proposes an advanced search form. It allows cascading constraints on the labels of the annotations and/or on relations between intervals. The relations are: is inside, overlaps, overlaps only begin time, overlaps only end time, is within...around, is within...around begin time of, is within...around end time of. The result is a list of filtered annotations the user can click on to visualize; it can also be saved as text file.

ANVIL internally maps the user's annotations to a temporary SQL database that is kept in sync at all times. Constraints can be formulated in SQL syntax. Labels of annotations can be queried using regular expressions. ANVIL also implements seven of the Allen relations [11] to compare intervals: equals, before, meets, overlaps, starts, finishes and during. In addition, the user can specify a tolerance limit in seconds. To spare the user from using long and complex SQL expressions, it implements a special syntax to ask for annotations from two tiers that are characterized by a certain temporal relationship.

The ANNIS2 system [12] proposes a query language (QL) including exact and regular expression matching on words forms and annotations, together with complex relations between individual elements, such as all forms of overlapping, contained or adjacent annotation spans, hierarchical dominance (children, ancestors, left- or rightmost child etc.) and more. Alternatively to the QL, data can be accessed using a graphical query builder. The result can be saved as text file or ARFF file.

To sum-up, the previously mentioned annotation tools offer the possibility to search or to retrieve annotations given some constraints. However, none of them fulfills the whole list of the following specifications a system should include:

- allowing to import multi-levels annotated data from most of the existing annotation tools;
- providing the filtered result in the form of a new annotation tier;
- dealing with interval tiers as well as point tiers;
- allowing to export the filtered tier(s) in most of the existing annotation tools;
- allowing to filter multiple files at once;
- proposing both a scripting language and a Graphical User Interface (GUI);
- being powerful enough to meet the reasonable needs of end-users;
- can be used without requiring any XML-related or QL-related knowledge or skill;

3. DataFilter in SPPAS

The system proposed in this paper is implemented as a component named DataFilter in SPPAS [6], a software for "Automatic Annotation of Speech" and distributed under the terms of the GNU Public License. It is implemented using the programming language Python. This software fulfills the specifications listed in [13]: it is a linguistic tool, free of charge, ready and easy to use, it runs on any platform and it is easy to install, the maintenance is guaranteed and it is XML-based.

Our proposal is to use the simplest level of representation, which is independent from the constraints of the coding procedure and the tools. Requests are based on the common set of information all tools are currently sharing. Basic operations are proposed and their combination allows the data to be requested, even by non-experts. Such a system fulfills the eight specifications mentioned in Section 2.

The framework implemented in this software to represent multi-levels annotated data is particularly suitable in the context of this paper to compare bounds of intervals or points between the various tiers: SPPAS solves the problem of the imprecision of annotations for each domain. Indeed, it allows to represent a bound as a tuple (M, R) , where M is the midpoint value and R is a radius value, i.e. the vagueness of the point, as described in [14]. Consequently, each boundary of the annotations is represented as an uncertain time value: it makes it possible to account explicitly for the imprecision of input data. For example, the radius value can be fixed to 40-80ms in case of Gestures annotations and 5-10ms in case of Phonetics. This representation allows robust comparisons of multi-levels annotations over time. SPPAS also allows annotations to contain more than one label, each one associated with a score: the one with the highest score is considered as the main label, and the others as alternatives. Moreover, labels can be of 3 types: string, number or Boolean.

Actually, it is also quite easy to read some existing annotation file formats and to instantiate them into the SPPAS framework. Among others, it allows to open and save files from *Praat* [3], *Phonedit* [15], *Elan* [2], *HTK* [16] and *Sclite* [17] and some subtitles formats. It also allows to import data from *Anvil* [1] and *Transcriber* [4].

The common denominator of most of the file formats consists in the basic building blocks (e.g. labels with start and end times, or labels and one time point) plus the additional structural entities (tiers). So, the system proposed in this paper is exploiting only these information: it allows to request all annotations regardless the input file format or the annotation type.

The exploration method is based on the creation of 2 different types of predicates. These latter are then respectively used in 2 types of filters:

1. single filters (Section 4), i.e. search in a tier depending on the data content, the time values or the duration of each annotation;
2. relation filters (Section 5), i.e. search on annotations of a tier in time-relation with annotations of another one.

4. Filtering annotations of a single tier

The main principle here is to create a predicate *Sel*, or a combination of predicates, that will be used as parameters to create a filter on a tier, named *SingleFilter(predicate, tier)*.

4.1. Filtering on the annotation labels

Pattern selection is an important part to extract data of a corpus and is obviously an important part of any filtering system, as shown in Section 2. Thus, if the label of an annotation is a string, the following predicates are proposed:

- exact match: $Sel(exact = P)$ is true if the label of an annotation strictly corresponds to the pattern P ;
- contains: $Sel(contains = P)$ is true if the label of an annotation contains the expected pattern P ;
- starts with, $Sel(startswith = P)$ is true if the label of an annotation starts with the expected pattern P ;
- ends with, $Sel(endswith = P)$ is true if the label of an annotation ends with the expected pattern P .

All these matches can be reversed to represent respectively: not exactly match, not contains, not starts with or not ends with. Moreover, this pattern matching can be case sensitive or not. For complex search, a selection based on regular expressions is available for advanced users, as $Sel(regex = R)$, where R is the expected regexp. Moreover, in case of numerical labels, we implemented: $Sel(equal = v)$, $Sel(greater = v)$ and $Sel(lower = v)$, and in case of Boolean: $Sel(bool = v)$. Finally, this pattern matching can be optionally applied either on the label with the highest score, which is the default, or on all labels of an annotation (i.e. the better label and its alternatives).

4.2. Filtering on annotations durations or on a time-range

Another important feature for a filtering system is the possibility to retrieve annotated data of a certain duration, and in a certain range of time in the timeline. Therefore, the same predicate Sel can be used on match duration of an interval, compared to a value v , as follow:

- lower: $Sel(duration.lt = v)$;
- lower or equal: $Sel(duration.le = v)$;
- greater: $Sel(duration.gt = v)$;
- greater or equal: $Sel(duration.ge = v)$;
- equal: $Sel(duration.e = v)$;

Search can also starts and ends at specific time values in a tier by using Sel predicate with $begin.ge$ and $end.le$.

4.3. Multiple selections

A multiple pattern selection as well as duration or time selections can be expressed with the operator "|" to represent the logical "or" and the operator "&" to represent the logical "and", for example:

$Sel(startswith = P_1) \& Sel(duration.gt = v)$.

5. Filtering on relations between two tiers

Regarding the searching, linguists are typically interested in locating patterns on specific tiers, with the possibility to relate different annotations a tier to another. The proposed system offers a powerful way to request/extract data, with the help of Allen's interval algebra. The main principle here is to create a predicate Rel that will be used as parameter to create a filter on a tier: $RelationFilter(predicate, tier1, tier2)$.

5.1. Framework: Allen's interval algebra

In 1983 James F. Allen published a paper [11] in which he proposed 13 basic relations between time intervals that are distinct, exhaustive, and qualitative. They are distinct because no pair of definite intervals can be related by more than one of the relationships; exhaustive because any pair of definite intervals are described by one of the relations; qualitative (rather than quantitative) because no numeric time spans are considered. These relations and the operations on them form Allen's interval algebra.

SPPAS extended Allen's work to its framework that can handle relationships between intervals with *precise as well as imprecise bounds*. This results in a generalization of Allen's 13 interval relations that are also applicable when the bounds of the intervals are imprecise. Table 1 indicates the Allen's relations between TimeInterval $X = [X^-, X^+]$ and $Y = [Y^-, Y^+]$, where X^-, X^+, Y^-, Y^+ are TimePoint instances, as defined in [14]. This generalization preserves the 3 properties of Allen's original algebra mentioned above.

X relation Y	Description
before	$(X^+ < Y^-)$
after	$(X^- > Y^+)$
meets	$(X^+ = Y^-)$
met by	$(X^- = Y^+)$
overlaps	$(X^- < Y^-) \wedge (X^+ > Y^-) \wedge (X^+ < Y^+)$
overlapped by	$(X^- > Y^-) \wedge (X^- < Y^+) \wedge (X^+ > Y^+)$
starts	$(X^- = Y^-) \wedge (X^+ < Y^+)$
started by	$(X^- = Y^-) \wedge (X^+ > Y^+)$
during	$(X^- > Y^-) \wedge (X^+ < Y^+)$
contains	$(X^- < Y^-) \wedge (X^+ > Y^+)$
finishes	$(X^- > Y^-) \wedge (X^+ = Y^+)$
finished by	$(X^- < Y^-) \wedge (X^+ = Y^+)$
equals	$(X^- = Y^-) \wedge (X^+ = Y^+)$

Table 1: Allen's relations between two imprecise intervals X, Y

The proposed framework was also developed to include time annotations represented by a single TimePoint (mainly used in the Prosody domain). The relations can be extended to such time representation, as we propose in Table 2 between two TimePoint instances. Tables 3 and 4 show relations between a TimePoint and a TimeInterval. Each table considers all possible relations (each table forms a complete relation system).

X relation Y	Description
before	$(X < Y)$
after	$(X > Y)$
equal	$(X = Y)$

Table 2: Relations between two imprecise points X and Y .

These relations can then be used to search annotations of any kind in time-aligned tiers. It is particularly favorable in the context of multimodal annotations, where annotations are carried out thanks to various annotation tools, each one using its own representation of time. The proposed framework solves this problem in a clear, well-suited and well-defined way.

X relation Y	Description
before	$(X^+ < Y)$
after	$(X^- < Y)$
starts	$(X^- = Y)$
finishes	$(X^+ = Y)$
contains	$(X^- < Y) \wedge (X^+ > Y)$

Table 3: Relations between an imprecise interval X and an imprecise point Y .

X relation Y	Description
before	$(X < Y^-)$
after	$(X > Y^-)$
starts	$(X = Y^-)$
finishes	$(X = Y^-)$
during	$(X > Y^-) \wedge (X < Y^+)$

Table 4: Relations between an imprecise point X and an imprecise interval Y .

5.2. Filtering with time-relations

For the sake of simplicity, only the 13 relations of the Allen's algebra are available in the GUI. We withal implemented in Python the 25 relations proposed by [18] in the INDU model. This model fixes constraints on INtervals with Allen's relations and on DURATION - duration are equals, one is less/greater than the other.

Moreover, both our experience while using the proposed system and the user comments and feedback have led us to add the following options:

1. a maximum delay for the relations "before" and "after",
2. a minimum delay for the relations "overlaps" and "overlapped by".

All the above mentioned relations were implemented as predicates. With this proposal, a predicate can be for example $predicate = Rel("overlaps") | Rel("overlappedby")$ to find with syllables stretch across two words, and then by creating the filter $RelationFilter(predicate, tiersyllables, tiertokens)$.

6. Illustrations

DataFilter of SPPAS has been already used in several studies as to find correlations between speech and gestures [19], to find which gestures are produced while pausing [20] or to extract lexical feedback items [21] just to cite some of them.

While using the GUI, the user starts filtering tiers by running DataFilter and loading files of a corpus. The user selects the tier of each file that will serve as basis, and click on the appropriate "Filter" button (either Single or Relation). The user has then to define the predicates and to apply such filters. The program will generate new tiers with the matching annotations; each one is automatically added to its corresponding file.

In order to illustrate possible queries using SPPAS, the following request is processed in this section: *What speech and hand gestures the locutor produces right before, during and right after the interlocutor produces multimodal feedbacks versus verbal feedbacks only?*

We performed this request on 6 files of a corpus created by and belonging to the Institut Paoli-Calmettes (Marseille, France). This corpus is an authentic corpus of training sessions for doctors involved in role plays with an actor playing the role of a patient. The corpus is annotated on different levels of granularity. Tiers contain annotations of vocabulary, hand gestures, gaze, among other. In the context of this article, we will consider only 3 tiers:

1. P - Feedback: feedback produced by the patient
2. M - IPU: speech produced by the doctor and segmented into Inter Pausal-Units
3. M - Dimensions: hand gestures produced by the doctor

To perform the illustration request, the first stage consists in filtering the "P - Feedback" tier of each file to create an intermediate result with a tier containing *head and oral feedback* ("P + T") and *oral feedback* only ("P").

While using the GUI, this predicates are fixed as represented in Figure 1. It allows to enter multiple patterns at the same time (separated by commas) to mention the system to retrieve either one pattern or the other, etc.

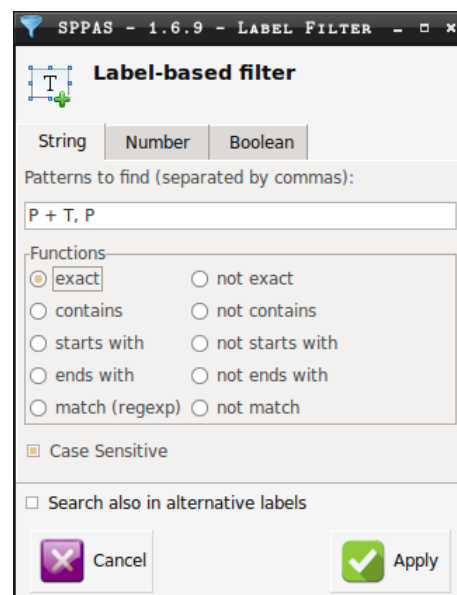


Figure 1: Frame to create a filter on annotation labels. In that case, labels that are exactly matching "P + T" or "P" strings.

So, here the patterns are "P + T, P". Finally, the user has to select the tier name for the result as shown in Figure 2 and must click either to "Apply all" or "Apply any". The user has now one filtered tier by file, each one containing only *oral feedbacks* and *oral and head movements feedbacks*.

To complete the original request, the previous tiers must be unchecked. The user must now find annotations of speech and hand gestures that occur right before, during and right after the feedbacks previously filtered. To do so, the newly filtered tiers must in turn be checked and the user must click on the "RelationFilter" button. Then, he/she selects "M - IPU" in the "X" windows, and the filtered tier previously created in the "Y" window in the list of proposed tiers, as he/she wants to filter speech. Finally, the Allen's relations must be selected: see a glimpse in Figure 3. Regarding the example, quite every relations are needed. Though, the relations "Before" and "After" must be

customized. The user needs to extract IPU's before and after the feedbacks. Customizing the delay allows the user to choose the exact delay between the feedback utterance and the nearer IPU's the user wants to take into consideration. To complete the filtering process, it must be clicked on the "Apply filter" button and the new resulting tiers are added in the annotation file(s).

In order to answer the question firstly asked, the user must complete the filter loop once again. He/she must click again on the "Relation Filter" button and select "M - Dimensions" in the "X" windows, and the previously filtered tier in the "Y" window, as the user wants, this time, to filter hand gestures in the list of proposed tiers. Then, the relations must be selected afresh. As the user does not want hand gestures produced out of the IPU's window, the user must check: Starts, Started by, Finishes, Finished by, Contains, During, Overlaps, and Overlapped by. Then, it must be clicked one last time on the "Apply filter" button and the new resulting tiers are added in the annotation file(s). The last resulting tier therefore contains the annotations of hand gestures produced by the locutor while speaking, right before, during and right after the interlocutor produced *oral or oral and head movements* feedback.

The user can keep or delete intermediate tiers and click on the "Save" button. The files are saved in their original file format and can therefore be opened in the annotation tool used to create the files in the first place. They can also be opened by "Statistics" component proposed in SPPAS.

7. Conclusions

This paper described a system to filter multi-levels annotations. It is based on the definition of predicates that are used to create filters. These later are applied on tiers of many kind of input files (TextGrid, eaf, trs, csv...). The search process results in a new tier, that can re-filtered and so on. A large list of predicates, applied on a single tier, is available and allows to filter annotations of various label types (string, number, Boolean). The full list of binary relations of the INDU model are also available to filter the annotations of a tier in relation with the annotations of another one. Moreover, this request system can be used either on precise or unprecise annotations. It is available as a Python Application Programming Interface, and with a GUI. Since the proposed system has been developed in a constant exchange with users from various Linguistics fields, we expect it to be especially intuitive.

8. Acknowledgements

This work has been partly carried out thanks to the support of the French state program ORTOLANG (Ref. Nr. ANR-11-EQPX-0032) funded by the "Investissements d'Avenir" French Government program, managed by the French National Research Agency (ANR). This work was supported by Brain and Language Research Institute.

9. References

- [1] M. Kipp, "Anvil, DFKI, German Research Center for Artificial Intelligence," <http://www.anvil-software.de/>, 2011.
- [2] Nijmegen: Max Planck Institute for Psycholinguistics, "ELAN - linguistic annotator. language archiving technology portal [computer software]," <http://www.lat-mpi.eu/tools/elan/>, 2011.
- [3] P. Boersma and D. Weenink, "Praat: Doing phonetics by computer. [Computer Software] Amsterdam: Department of Language and Literature, University of Amsterdam.," <http://www.praat.org/>, 2011.
- [4] Transcriber, "A tool for segmenting, labeling and transcribing speech. [computer software] paris: DGA," <http://transag.sourceforge.net/>, 2011.
- [5] T. Schmidt and K. Wörner, "Exmaralda," in *Handbook on Corpus Phonology*, U. G. Jacques Durand and G. Kristoffersen, Eds. Oxford University Press, 2014, pp. 402–419.
- [6] B. Bigi, "SPPAS: a tool for the phonetic segmentations of Speech," in *The eighth international conference on Language Resources and Evaluation, ISBN 978-2-9517408-7-7*, Istanbul, Turkey, 2012, pp. 1748–1755.
- [7] C. Jewitt, "Handbook of multimodal analysis," London: Routledge, 2009.
- [8] T. Bigbee, D. Loehr, and L. Harper, "Emerging requirements for multi-modal annotation and analysis tools." in *INTERSPEECH*, 2001, pp. 1533–1536.
- [9] Á. Abuczki and E. B. Ghazaleh, "An overview of multi-modal corpora, annotation tools and schemes," *Argumentum*, vol. 9, pp. 86–98, 2013.
- [10] S. T. Gries and A. L. Berez, "Linguistic annotation in/for corpus linguistics," *Handbook of Linguistic Annotation. Berlin, New York: Springer. Abgerufen von*, 2015.
- [11] J. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, pp. 832–843, 1983.
- [12] A. Zeldes, A. Lüdeling, J. Ritz, and C. Chiarcos, "ANNIS: a search tool for multi-layer annotated corpora," 2009.
- [13] S. Dipper, M. Götze, and M. Stede, "Simple annotation tools for complex annotation tasks: an evaluation," in *Proceedings of the LREC Workshop on XML-based richly annotated corpora*, 2004, pp. 54–62.
- [14] B. Bigi, T. Watanabe, and L. Prévot, "Representing multimodal linguistics annotated data," in *9th International conference on Language Resources and Evaluation, ISBN: 978-2-9517408-8-4*, Reykjavik (Iceland), 2014, pp. 3386–3392.
- [15] Phonedit, <http://www.lpl-aix.fr/~lpldev/phonedit/>, 2014.
- [16] S. Young and S. Young, "The HTK Hidden Markov Model Toolkit: Design and Philosophy," *Entropic Cambridge Research Laboratory, Ltd*, vol. 2, pp. 2–44, 1994.
- [17] "Speech Recognition Scoring Toolkit, version 2.4.0," <http://www.itl.nist.gov/iad/mig/tools/>, 2009.
- [18] A. K. Pujari, G. V. Kumari, and A. Sattar, "Indu: An interval & duration network," in *Advanced Topics in Artificial Intelligence*. Springer, 1999, pp. 291–303.
- [19] M. Tellier, G. Stam, and B. Bigi, "Same speech, different gestures?" in *5th International Society for Gesture Studies*, Lund (Sweden), 2012.
- [20] —, "Gesturing while pausing in conversation: Self-oriented or partner-oriented?" in *The combined meeting of the 10th International Gesture Workshop and the 3rd Gesture and Speech in Interaction conference*, Tillburg (The Netherlands), 2013.
- [21] L. Prévot, B. Bigi, and R. Bertrand, "A quantitative view of feedback lexical markers in conversational French," in *Proceedings of 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Metz, France, 2013.

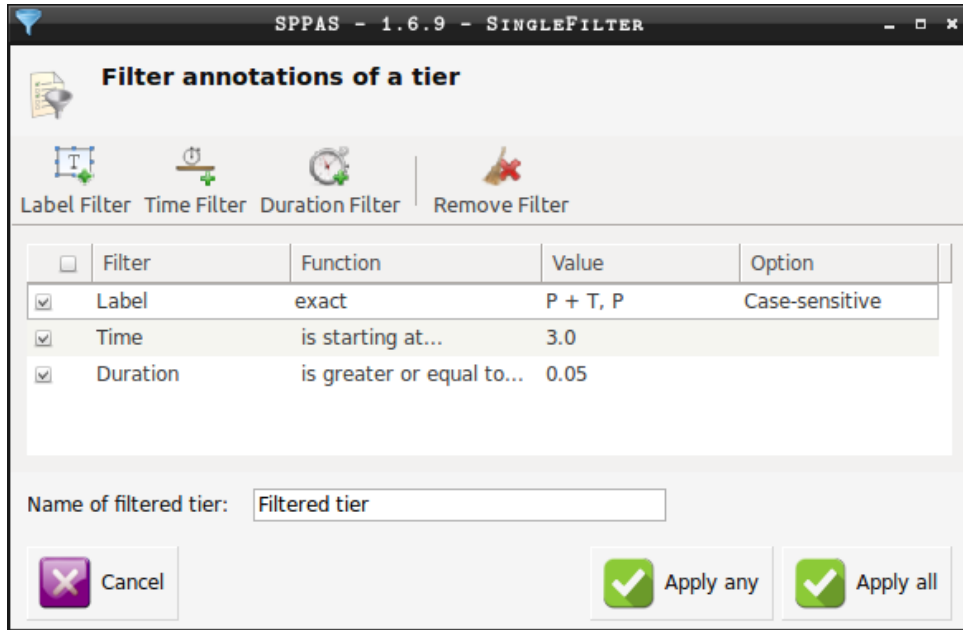


Figure 2: Example of the Single filter frame. For the purpose of an exhaustive illustration, 3 predicates are described here 1/ to select patterns, 2/ to create a filter on annotation duration and 3/ on a time-range.

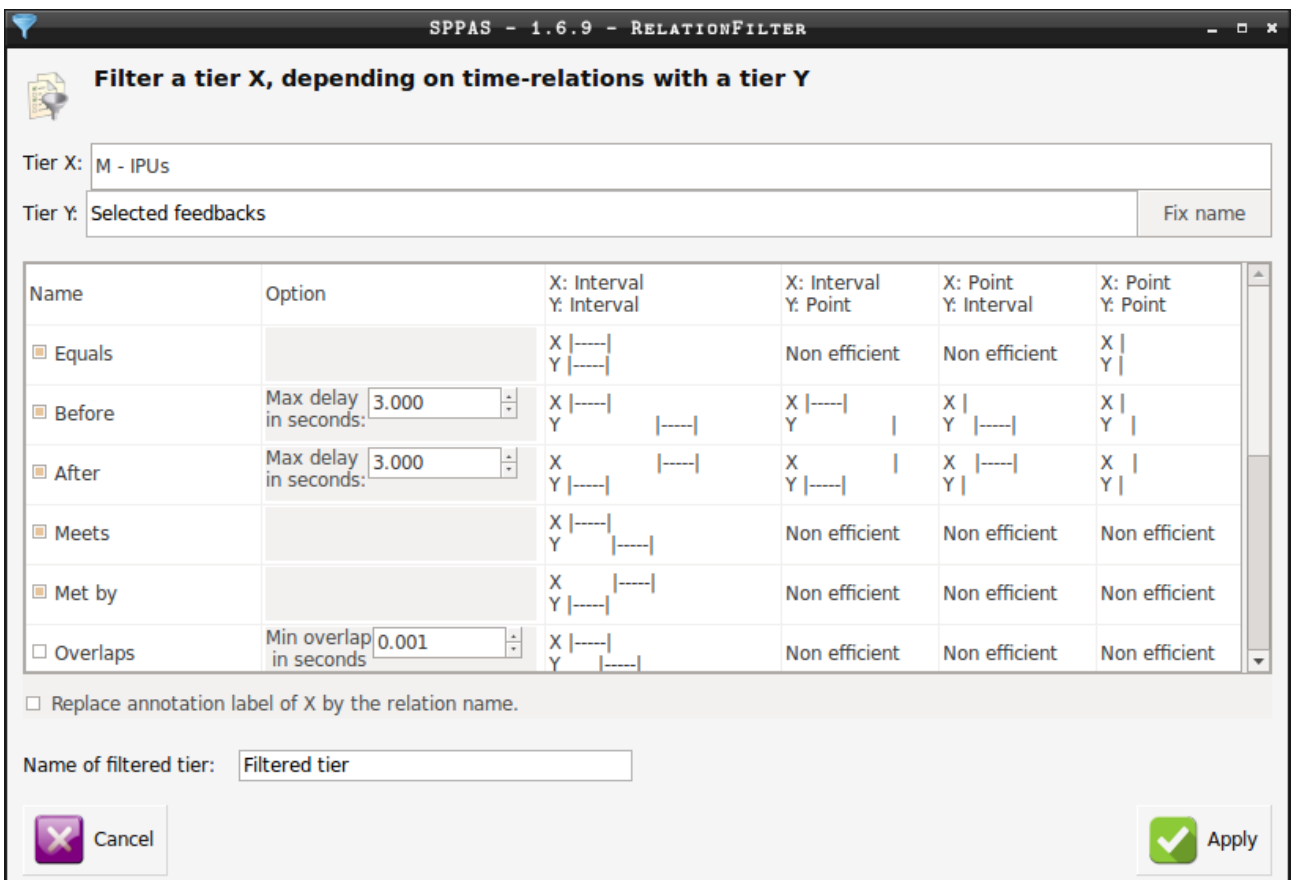


Figure 3: Frame to create a filter on time-relations