The CAMOMILE Collaborative Annotation Platform for Multi-modal, Multi-lingual and Multi-media Documents

Johann Poignant[1], Mateusz Budnik[2], Hervé Bredin[1], Claude Barras[1], Mickael Stefas[3], Pierrick Bruneau[3], Gilles Adda[1,6], Laurent Besacier[2], Hazim Ekenel[4], Gil Francopoulo[6], Javier Hernando[5], Joseph Mariani[1,6], Ramon Morros[5], Georges Quénot[2], Sophie Rosset[1], Thomas Tamisier[3]

1. LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, F-91405 Orsay, firstname.lastname@limsi.fr
2. LIG, Univ. Grenoble Alpes, Grenoble, France, firstname.lastname@imag.fr
3. LIST, Esch-sur-Alzette, Luxembourg, firstname.lastname@list.lu
4. ITÜ, Istanbul, Turkey, lastname@itu.edu.tr
5. UPC, Barcelona, Spain, firstname.lastname@upc.edu
6. IMMI-CNRS, Orsay, France

Abstract

In this paper, we describe the organization and the implementation of the CAMOMILE collaborative annotation framework for multimodal, multimedia, multilingual (3M) data. Given the versatile nature of the analysis which can be performed on 3M data, the structure of the server was kept intentionally simple in order to preserve its genericity, relying on standard Web technologies. Layers of annotations, defined as data associated to a media fragment from the corpus, are stored in a database and can be managed through standard interfaces with authentication. Interfaces tailored specifically to the needed task can then be developed in an agile way, relying on simple but reliable services for the management of the centralized annotations. We then present our implementation of an active learning scenario for person annotation in video, relying on the CAMOMILE server; during a dry run experiment, the manual annotation of 716 speech segments was thus propagated to 3504 labeled tracks. The code of the CAMOMILE framework is distributed in open source.

Keywords: Annotation tool, collaborative annotation, multimedia, active learning, person annotation.

1. Introduction

Human activity is generating growing volumes of heterogeneous data, available in particular via the Web. Multimodal, multimedia, multilingual (3M) data can be collected and explored to gain new insights in social sciences, linguistics, economics, behavioural studies as well as artificial intelligence and computer sciences. But to be analyzed through statistical-based machine learning methods, these data should be available in very large amounts and annotated. Annotating data is costly as it involves manual work, and in this regard 3M data, for which we need to annotate different modalities with different levels of abstraction is especially costly.

Current annotation frameworks often involve a local manual annotation, e.g. LDC annotation tools (Maeda and Strassel, 2004), Anvil (Kipp, 2001), Viper[1] (Doermann and Mihalcik, 2000) sometimes supported by automatic processing as proposed by ELAN (Auer et al., 2010). In this case, dealing with multiple annotators and different versions of the annotation files quickly becomes unfeasible. Browser-based annotations interfaces, linked to a server for storing the annotations, can provide a solution to this problem, all the more so as browser performance and multimedia support dramatically improved in the recent years; this is a direction taken e.g. by LDC with its WebAnn initiative or by the PubAnnotation project[2] which is a repository of text annotations and associated services in the domain of life science literature.

In the context of the CHIST-ERA CAMOMILE project (Collaborative Annotation of multi-MOdal, multI-Lingual and multi-mEdia documents)[3], we developed a collaborative annotation framework for 3M data. We focused our work on the annotation of people, which are usually the centre of attention in 3M documents, and chose a use-case driven approach with different scenarios: collaborative annotation where several human annotators simultaneously label either the same or a different layer of a document, the monitoring and coordination of the annotation workflow, or active learning applications (Ayache and Quénot, 2008), where an automatic person recognition system is used to bootstrap a manual annotation and is retrained or adapted using this result. Developers of mono-modal person identification components are also interested in the error analysis of automatic annotation systems, and need an interface that allows an easy visualization of errors and navigation between the reference annotation and the hypothesized outputs.

In this paper, we first describe the organization and the implementation of the framework server. Given the versatile nature of the analysis which can be performed on 3M data, the structure of the server was kept intentionally simple in order to preserve its genericity. Interfaces tailored specifically to the needed task can then be developed in an agile

[1] http://viper-toolkit/sourceforge.net/

[2] http://www.pubannotation.org

[3] https://camomile.limsi.fr

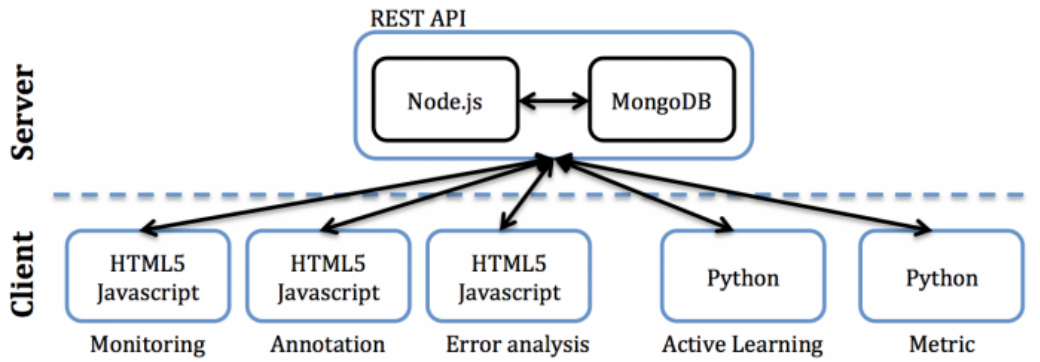

Figure 1: CAMOMILE client-server platform

way, relying on simple but reliable services for the management of the centralized annotations. We then present our implementation of an active learning scenario for person annotation in video, relying on the CAMOMILE server.

2. Collaborative Annotation Server

Aiming for a flexible development and use, the collaborative annotation framework is implemented using state of the art web-based technologies, namely libraries and tools developed mostly in Javascript and Python. The ultimate purpose of the platform is to provide on demand overviews and details regarding the 3M data, and the associated automatic and manual annotations. Some views would be dedicated to the fine-grain inspection of annotation files, while some others would serve some higher-level task, such as summarizing the media of a specific corpus according to the exhaustiveness of their associated annotations, or the performance of the algorithms to infer the latter. Multiple users may be involved, synchronously or asynchronously, and with several roles (manual annotators, recognition algorithm developer, adjudicator). Web-based technologies and Javascript already comprise many building blocks suitable for supporting this kind of collaborative behavior. As illustrated in Figure 1, the proposed collaborative annotation framework follows a client/server architecture. This paradigm facilitates the work of multiple users on consistent data sources, as required by the project specifics. The

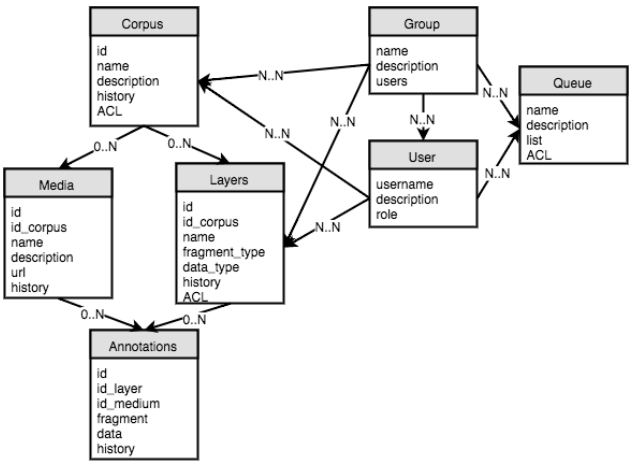

Figure 2: CAMOMILE server data model

involved server-side technologies rely solely on exchanges via the HTTP protocol, facilitating the design of interoperable software components. The server focuses essentially on data and authentication management tasks, leaving the application logic to the client side. The aim of doing so is to design a general and consistent service, allowing the agile development of browser-based clients, each implemented according to a concrete use-case (e.g., annotation, error analysis . . .).

There are two main reasons that made us implement client/server interfaces using REST services. First, in the REST architecture style, clients and servers exchange representations of resources using a standardized interface and protocol, which is in line with a clear separation between the data model and the interface controlling the annotations as specified in the Model-View-Controller paradigm. These principles encourage RESTful applications to be simple, lightweight, and have high performance. Second, RESTful web services typically map the four main HTTP methods to perform predefined operations: GET, PUT, DELETE, and POST. These operations can fulfill the needs of annotating 3M data, such as create, read, update or delete annotations. Resources are the fundamental concept in any RESTful API, and thus they need to be specified (as collections, tables, relationships between them) before designing API services. In our framework, resources are annotations, which are represented in JSON formats, stored in a MongoDB database. Based on the use cases designed in the CAMOMILE project, we specify the following collections (i.e. tables in traditional database systems) for our application: corpus, media, layers, annotations. The corpus collection describes all available corpora. Each corpus contains a set of media and a set of layers. A medium corresponds to a multimedia resource (e.g., a video or audio file). A layer is composed of multiple annotations with the same type (e.g. one layer for manual annotations of speech turns or one layer for annotations of face tracks). An annotation is uniquely defined by a media fragment (e.g., a temporal segment) and attached data (e.g. the name of the current speaker).

Figure 2 illustrates an entity-relationship diagram of these collections (tables), where: *id* is the identifier of a resource; *id_xxx* is the identifier of the resource *xxx*; *name* provides a short description of a resource (corpus, media, layer, user, group or queue). *fragment_type* is a generic type and de-

CRUD action	HTTP command	Python/Javascript interface
Create	POST /resource	client.createResource(...)
Read	GET /resource/*id_resource*	client.getResource(*id_resource*)
Update	PUT /resource/*id_resource*	client.updateResource(*id_resource*, ...)
Delete	DELETE /resource/*id_resource*	client.deleteResource(*id_resource*)

Table 1: Correspondance between HTTP commands and Python or Javascript interface for a given resource (either corpus, medium, layer or annotation).

scribes the type of the fragments, which are annotation units stored in the annotation collection. *fragment_type* can currently be segments or rectangles (for face recognition). *data_type* is the data type of each annotated fragment, and it can be a person name or a spoken word. The range of supported types is designed to be easily extended to potentially new annotation tasks; in any case, their proper visualization and management will be the responsability of the client application. History is a list of modifiers and each medium can be accessed via an url. Similar to file rights management on operating systems, ACL (Access Control List) correspond to the user or group rights (read, write, admin) on the resource; they are defined for the corpus and layers but not directly for media or annotations: indeed, the media inherits the rights of the corpus to which it belongs and the annotations inherit the rights of the layers to which it belongs, limiting the granularity of the ACL management. Group and user are defined by a unique name and a description, a group contains a list of users and the user role can be a simple "user" or "admin" of the server. Finally, a last table is dedicated to queues which correspond to a list of elements. Queues have also a management of users and groups rights.

A documentation with all the routes available on this server is provided online[4]; the source code of the camomile server is distributed under MIT open source license[5] and is also provided in a Docker image allowing an easy installation. Along with the server, Python and JavaScript clients embedding the REST API into native language objects have also been developed and are distributed in sibling repositories; Table 1 displays the general correspondance between the HTTP and client object-oriented syntax.

3. Active Learning Use Case

An interface has been designed for person annotation in videos using the proposed platform. This above framework can be easily extended with additional functionality to make the annotation process more efficient. To that end, an active learning use case is presented, which can be useful when the full annotation of the whole dataset is impossible due to cost or time constraints. Active learning is a set of algorithms that help with selecting potentially informative samples to be labeled by human annotators, while trying to avoid those that are redundant. These samples can help to efficiently train accurate models for prediction or to enhance the quality and accuracy of the data clustering.

[4] `http://camomile-project.github.io/camomile-server`

[5] `https://github.com/camomile-project/camomile-server`

In this scenario, the active learning (AL) system is based on the propagation of labels with the use of hierarchical clustering. Ideally, each cluster should correspond to a single person. Normally, the goal is to give a label (be it manually or automatically) to every segment in the dataset and to create the purest possible clusters.

Automatically generated segments of a multimedia resource are used as queries. These segments are initially clustered using a distance measure dependent on the media (e.g. distance between HOG descriptors for face segments). Afterwards, a cluster is selected for annotation based on a specific criterion (e.g. size of the cluster or how dense and well separated it is) and depending on the annotation scenario. Next, a representative segment from the selected cluster is presented for the user to label. Initially, it is the one closest to the centroid of a cluster, i.e. having the minimal distance to all other segments within it. Once a cluster contains at least one annotated segment, potential outliers are selected, which are most likely to contain information belonging to a different person.

After a segment is annotated by the user the cluster structure is modified accordingly. For example, two clusters that contain segments with the same name are merged. On the other hand, a cluster which contains segments with different labels is divided. A label assigned to a segment within a cluster is propagated to all of its unlabeled members.

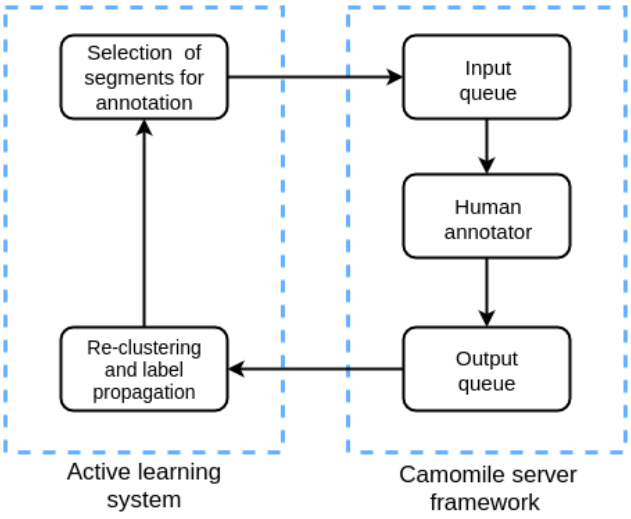

Figure 3: Active learning system

The AL system is connected to the collaborative annotation framework via the use of two different types of queues. The interaction between these two systems is shown in Figure 3. An input queue is filled with annotations, i.e. media fragments (see Figure 2), which are suggested by the AL

algorithm for annotation. The task for the human annotator can be either identification, if the data field is empty (no label available yet to the fragment), or verification when a label is already assigned to the media fragment. The latter case can be used to verify the quality of automatically propagated labels or previous annotations done by humans. The most recent label given to a fragment is stored in the data field, while all the previous ones are added to the history list. This mechanism can be used for data cleaning and conflict resolution (Safadi et al., 2012). For example a segment, which is first annotated automatically through label propagation and then again with the same label by a human annotator can be consider as correct, while a mismatch between the two would require a third opinion. From the input queue the annotations are distributed among the human annotators. Once a media fragment is assigned to a user, it is removed from the queue.

The annotations processed by the users are pushed to the output queue. Because the media fragments are automatically generated, some may be noisy or may not contain any information related to the task (speech segments with only music or silence for example). They can be skipped by the user and will be stored separately and not be used by the AL system. The named annotations are removed from the output queue and used in the next step of the system.

The technical implementation of such a system would require the computations to not be carried on the client side. Instead, an additional server may be used. This has the advantage of having just a single input and output queue for all the remote annotators. Also, each multi-media document (e.g. a single video or a part of it) can be processed in parallel, which would increase the responsiveness of the AL system.

To test this setting in practice a limited dry run was done involving human annotators. The task consisted of annotating speech segments extracted automatically from TV broadcast videos. The segment extraction followed an approach presented in (Barras et al., 2006). Each annotator was given a fragment of a video corresponding to the time frame of a speech segment and was asked to name the person who is speaking at the time.

A second separate server was used to run the active learning system. It was communicating with the CAMOMILE server via the input and output queues as mentioned earlier. Connection to the clients as well as unlabeled segment allocation to the human annotators was handled by the CAMOMILE server. The annotation were done using the graphical user interface (GUI) developed within the CAMOMILE project[6] (Bruneau et al., 2014). An example of this interface can be seen in Figure 4. The GUI allows the users to see the video fragment, which needs to be annotated. They also have access to the parts of the video before and after the current fragment. Previous annotations of this video segment are also visible.

A total of 9 annotators were involved in the dry run. The annotations were all done at the same time and lasted for around 1.5 hours per user. In this run only the speaker annotation scenario was tested (the faces of people that could be seen in the video were not annotated). The corpus consisted of 62 videos from the REPERE dataset (Giraudel et al., 2012), which contains French TV debates, news programs and parliamentary broadcasts.

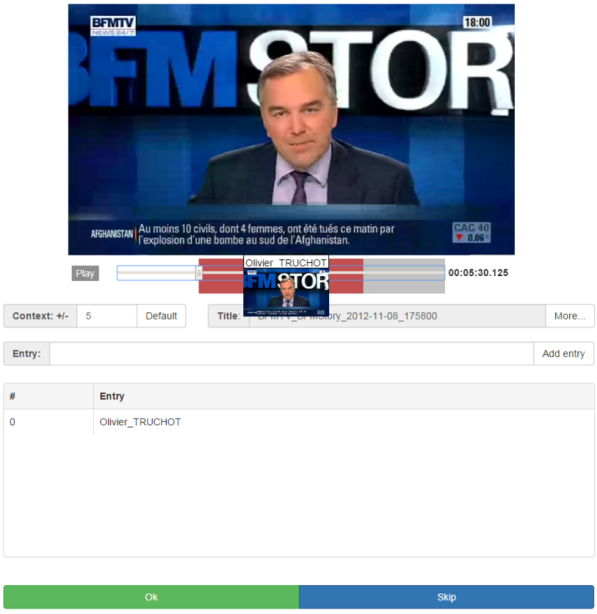

Figure 4: The browser interface used in the dry run.

During the dry run, 716 speech segments (with the total duration of 81 minutes) were manually annotated. On top of that, 654 segments (around 68 minutes) were marked as skipped, which means that they do not contain speech, but music, external noises or silence. Additionally, due to the clustering used in the active learning system, the annotations were propagated to the corresponding clusters. This gave a total number of 3504 labeled segments (including the 716 annotated manually) with the total time of 7.81 hours. Additionally, the use of the multimodal clusters during the dry run enabled to get face annotation of the people seen in the video fragments corresponding to the speech segments (1973 face annotations with the duration of 5.47 hours). The median annotation time of a single segment is equal to 10.8 seconds.

Overall, the real-life application of this framework showed its usability and potential benefits. This is especially true when one considers the amount of annotation obtained through automatic means of label propagation, even for modalities, like the faces seen in the segments, that were not explicitly labeled.

4. Conclusion

The purpose of the CAMOMILE project has been to explore new practices around collaborative annotation and test it on specific use cases with dedicated prototypes. The developed framework can be summarized as a remote repository of annotations which are metadata attached to fragments of the media from a corpus, along with the associated RESTful API. It is thus compatible with other abstraction layers, e.g., annotation graphs (Bird and Liberman, 2001), and the metadata can follow standards

[6] https://github.com/camomile-project/camomile-web-frontend

in the domain as proposed in the META-SHARE initiative (Piperidis, 2012)[7]. This simple framework was robust enough to support the active learning scenario described in this paper, as long as the organization of a MediaEval task with 20 annotators involved (73426 annotations) (Poignant et al., 2016). Its source code is freely available on GitHub. Further developments would improve the platform, like a direct communication between the clients through WebSockets or a flexible historization of the annotations.

5. Acknowledgements

We thank the members of the CAMOMILE international advisory committee for their time and their precious advices and proposals. This work was done in the context of the CHIST-ERA CAMOMILE project funded by the ANR (Agence Nationale de la Recherche, France) under grant ANR-12-CHRI-0006-01, the FNR (Fonds National de La Recherche, Luxembourg), Tübitak (scientific and technological research council of Turkey) and Mineco (Ministerio de Economía y Competitividad, Spain).

6. References

Auer, E., Russel, A., Sloetjes, H., Wittenburg, P., Schreer, O., Masnieri, S., Schneider, D., and Tshöpel, S. (2010). ELAN as Flexible Annotation Framework for Sound and Image Processing Detectors. In *7th International Conference on Language Resources and Evaluation (LREC)*, pages 890–893.

Ayache, S. and Quénot, G. (2008). Video corpus annotation using active learning. In *30th European Conference on Information Retrieval (ECIR'08)*, pages 187–198.

Barras, C., Zhu, X., Meignier, S., and Gauvain, J. (2006). Multistage speaker diarization of broadcast news. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(5):1505–1512.

Bird, S. and Liberman, M. (2001). A formal framework for linguistic annotation. *Speech Communication*, 33(1-2):23–60.

Bruneau, P., Stefas, M., Budnik, M., Poignant, J., Bredin, H., Tamisier, T., and Otjacques, B. (2014). Collaborative annotation of multimedia resources. In *Cooperative Design, Visualization, and Engineering (CDVE 2014)*, pages 163–166. Springer.

Doermann, D. and Mihalcik, D. (2000). Tools and techniques for video performance evaluation. In *15th International Conference on Pattern Recognition (ICPR)*, pages 167–170.

Giraudel, A., Carré, M., Mapelli, V., Kahn, J., Galibert, O., and Quintard, L. (2012). The REPERE Corpus: a multimodal corpus for person recognition. In *8th International Conference on Language Resources and Evaluation (LREC)*, pages 1102–1107.

Kipp, M. (2001). Anvil - a generic annotation tool for multimodal dialogue. In *Proc. of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pages 1367–1370.

Maeda, K. and Strassel, S. (2004). Annotation tools for large-scale corpus development: Using agtk at the linguistic data consortium. In *4th International Conference on Language Resources and Evaluation (LREC)*, pages 2077–2080.

Piperidis, S. (2012). The meta-share language resources sharing infrastructure: Principles, challenges, solutions. In *8th International Conference on Language Resources and Evaluation (LREC)*, pages 36–42.

Poignant, J., Bredin, H., Barras, C., Stefas, M., Bruneau, P., and Tamisier, T. (2016). Benchmarking Multimedia Technologies with the CAMOMILE Platform: the Case of Multimodal Person Discovery at MediaEval 2015. In *10th International Conference on Language Resources and Evaluation (LREC)*.

Safadi, B., Ayache, S., and Quénot, G. (2012). Active cleaning for video corpus annotation. In *International MultiMedia Modeling Conference (MMM)*, pages 518–528.

7. Language Resource References

REPERE Evaluation Package. (2014). REPERE project, distributed via ELRA, 1.0, ISLRN : 360-758-359-485-0.

[7] http://www.meta-net.eu/meta-share